การจัดตั้งทีมพัฒนาซอฟต์แวร์แบบสกรัมโดยใช้ทักษะด้านเทคนิคเป็นฐาน และผลกระทบต่อทักษะการเขียนโปรแกรมของกลุ่มผู้พัฒนาซอฟต์แวร์ระดับเริ่มต้น

Technical Skills-Based Team Formulation in Scrum Software Development and Its Impact on Programming Skills in Entry-Level Software Developers

อภิสิทธิ์ แสงใส (Apisit Saengsai)* และ ณัฐพร ภักดี (Nuttaporn Phakdee)*

Received: June 10, 2021
Revised: July 30, 2021
Accepted: August 31, 2021

* ผู้นิพนธ์ประสานงาน: ณัฐพร ภักดี (Nuttaporn Phakdee) อีเมล: nuttaporn@go.buu.ac.th

บทคัดย่อ

หลักการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern Software Development Principles) มีลักษณะเด่น คือ เน้นการพัฒนาซอฟต์แวร์ด้วยความเร็ว (Fast-Paced) โดยใช้ทีมพัฒนาร่วมมือกันทำงาน (Collaboration) ในลักษณะ Feature-Driven โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) ได้รับความนิยมและถูกนำมาใช้อย่างแพร่หลาย อย่างไรก็ตาม แม้ว่าองค์กรจะใช้วิธีการพัฒนาซอฟต์แวร์ที่ได้รับการยอมรับ แต่ยังคงพบกับความล้มเหลวของการพัฒนาซอฟต์แวร์อยู่บ่อยครั้ง จากงานวิจัยที่เกี่ยวข้องพบว่า สมาชิกภายในทีม (Team Members) ยังคงเป็นปัจจัยหลักในหลาย ๆ ปัจจัยความสำเร็จที่ส่งผลต่อการพัฒนาซอฟต์แวร์ งานวิจัยนี้มีวัตถุประสงค์เพื่อทดลองและนำเสนอผลลัพธ์จากการจัดทีมสกรัม โดยพิจารณาจากทักษะด้านเทคนิคเป็นฐาน และแสดงผลกระทบต่อการพัฒนาทักษะการเขียนโปรแกรมของกลุ่มผู้พัฒนาระดับเริ่มต้นหรือมีประสบการณ์น้อย ซึ่งพบว่า การจัดทีมสกรัมโดยใช้ทักษะด้านเทคนิคเป็นฐานของกลุ่มทดลองไม่มีความผลต่อการพัฒนาทักษะการเขียนโปรแกรมอย่างมีนัยสำคัญที่ระดับ 0.05 ทั้งนี้พบข้อสังเกตว่าทีมที่สมาชิกส่วนใหญ่มีทักษะด้านเทคนิคระดับสูงอยู่หลายคน (ร้อยละ 60 – 90 ของสมาชิกทั้งหมด) มีผลลัพธ์การพัฒนาทักษะการเขียนโปรแกรมที่แย่ที่สุด

คำสำคัญ: ทักษะด้านเทคนิค หลักการพัฒนาซอฟต์แวร์สมัยใหม่ การพัฒนาทักษะการเขียนโปรแกรม ผู้พัฒนาซอฟต์แวร์ระดับเริ่มต้น

Abstract

Modern software development principles are fast-paced, collaborative tasks that focus on real working software to meet deadlines called “feature-driven.” Scrum is the mostpopular modern software development tool. Nevertheless, team members are key success factors for software development. This research studied team formation based on the technical skill of each team member and its impact on programming skills in entry-level software developers. In this experiment the relationship between practical examination scores and team formation configuration could not be confirmed with a statistical significance of 0.05. However, team formation with a high percentage of technical skill members (60 - 90 percent of the total members in a team.) had the worst results.

Keywords: Technical Skills, Modern Software Development Team, Programming Skills Development, and Entry Level Software Developer.

1. บทนำ

สกรัม (Scrum) คือหลักการพัฒนาซอฟต์แวร์สมัยใหม่ที่ส่งเสริมให้สมาชิกภายในทีมพัฒนาซอฟต์แวร์สามารถปฏิบัติงานต่าง ๆ ด้วยตนเองได้ ตั้งแต่การวางแผนการออกแบบการเขียนโปรแกรม การทดสอบ รวมถึงการส่งมอบซอฟต์แวร์ จากการวิจัยพบว่า [1] สมาชิกภายในทีมพัฒนาซอฟต์แวร์เป็นปัจจัยหนึ่งที่ส่งผลต่อความสำเร็จหรือความล้มเหลว

* สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
* Software Engineering Program, Faculty of Informatics, Burapha University.

ในการพัฒนาซอฟต์แวร์ ซึ่งสมาชิกควรมีทักษะพื้นฐานที่สำคัญในการทำงาน เช่น ทักษะการออกแบบเชิงวัตถุ (Object Oriented design) ทักษะการทดสอบ (Testing skill) ทักษะการทำงานเป็นทีม (Teamwork skill) และทักษะการสื่อสาร (Communication) เป็นต้น อย่างไรก็ตาม ยังคงไม่มีการระบุอย่างเป็นทางการ สำหรับทักษะที่จำเป็นต้องมีในการพัฒนาซอฟต์แวร์ โดยเฉพาะอย่างยิ่งในการพัฒนาซอฟต์แวร์สมัยใหม่ Scrum, Extreme programming, Lean และ Kanban เป็นต้น

ในการพัฒนาซอฟต์แวร์ปัจจุบัน หลายองค์กรประสบปัญหาว่าการสรรหา คัดเลือก หรือจ้างงานบุคลากรที่มีทักษะ/ความเชี่ยวชาญ ครบถ้วนตามความต้องการขององค์กรเป็นเรื่องที่เป็นไปได้ยาก โดยเฉพาะอย่างยิ่ง ทักษะการเขียนโปรแกรม (Programming skill) ดังนั้นการพัฒนาทักษะการเขียนโปรแกรมจึงเป็นเรื่องที่สำคัญ โดยเฉพาะอย่างยิ่งในกลุ่มผู้พัฒนาระดับเริ่มต้นหรือมีประสบการณ์น้อย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอผลลัพธ์จากการจัดทีมพัฒนาซอฟต์แวร์แบบสกรัม โดยผู้วิจัยเรียกว่า ทีมสกรัม ซึ่งพิจารณาจากทักษะด้านเทคนิคเป็นฐาน และผลกระทบต่อการพัฒนาทักษะการเขียนโปรแกรมในกลุ่มผู้พัฒนาระดับเริ่มต้นหรือมีประสบการณ์น้อย โดยทดลองกับกลุ่มตัวอย่างนักศึกษาหลักสูตรวิศวกรรมซอฟต์แวร์ จำนวน 86 คน แบ่งเป็น 10 กลุ่ม ทั้งนี้ นักศึกษาแต่ละคนจะถูกวัดระดับทักษะด้านเทคนิคจากผลการเรียนในรายวิชาของหลักสูตรฯ ตามแนวปฏิบัติของการจำแนกด้านความรู้ (Knowledge areas) ที่นำเสนอโดย Association for Computing Machinery (ACM) ของหลักสูตรวิศวกรรมซอฟต์แวร์ [2] ได้แก่ 1) รายวิชาพื้นฐานที่จำเป็นสำหรับคอมพิวเตอร์ (Computing Essentials) 2) รายวิชาด้านคณิตศาสตร์คอมพิวเตอร์และวิศวกรรมซอฟต์แวร์ (Mathematical and engineering fundamentals) และ 3) รายวิชาด้านการวิเคราะห์ออกแบบ (Software design) รวมทั้งสิ้นจำนวน 8 รายวิชา โดยกำหนดโครงการให้นักศึกษาพัฒนา Web-based Application ออกแบบและพัฒนาโดยใช้แนวคิด Object-Oriented และใช้วิธีสกรัม (Scrum) ในการพัฒนาซอฟต์แวร์

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

"ทักษะด้านเทคนิค" (Technical Skills) ในงานวิจัยนี้หมายถึง ทักษะที่สำคัญและเกี่ยวข้องกับการพัฒนาซอฟต์แวร์เป็นการส่งผลโดยตรงกับชิ้นงานซอฟต์แวร์

2.1 หลักการพัฒนาซอฟต์แวร์สมัยใหม่

คือการพัฒนาซอฟต์แวร์ที่ให้ความสำคัญกับกรอบเวลา (Box of Time) แบ่งช่วงการพัฒนาออกเป็นช่วงสั้น ๆ (Sprint/Cycle) นำส่งซอฟต์แวร์อย่างต่อเนื่อง (Continuous Deployment) เน้นที่ความพึงพอใจของลูกค้า (User Satisfaction) สามารถทำงานในหลากหลายหน้าที่ (Cross-Functional) และสามารถบริหารจัดการตนเองได้ (Self-Organization) โดยตัวอย่างวิธีการพัฒนาซอฟต์แวร์สมัยใหม่ที่เป็นที่นิยม เช่น

- สกรัม (Scrum) เป็นวิธีที่ได้รับความนิยมในปัจจุบัน โดยแบ่งการพัฒนาซอฟต์แวร์ออกเป็นวงรอบ (Incremental) โดยใช้ความต้องการของลูกค้าเป็นเป้าหมายในการพัฒนาในแต่ละวงรอบ อยู่ในรูปแบบของชิ้นงาน (Product Backlog) โดยมี Product Owner เป็นผู้กำหนดความต้องการ
- Extreme programming (XP) เน้นการปฏิบัติงานร่วมกันของสมาชิกภายในทีม (Collaboration) โดยการสื่อสารกันอย่างต่อเนื่อง (Communication) สมาชิกทุกคนสามารถช่วยกันปฏิบัติงานและสามารถถ่ายทอดความรู้สู่สมาชิกคนอื่นได้ และมักใช้เทคนิคการเขียนโปรแกรมแบบคู่ (Pair Programming) ให้ถูกต้องตามความต้องการและตรงตามมาตรฐานการเขียนโปรแกรม (Coding Standards)

2.2 ทักษะสำหรับนักพัฒนาซอฟต์แวร์

Tan และ Teo [3] ได้นำรูปแบบการพัฒนาซอฟต์แวร์แบบเอไจล์ไปใช้ในการเรียนการสอน โดยนำเสนอผลลัพธ์ว่านักศึกษามีทัศนคติที่ดีขึ้นเกี่ยวกับเรื่องของคุณภาพของซอฟต์แวร์ และยังนำเสนอว่ายังส่งผลถึงทักษะในการจัดการกับการเปลี่ยนแปลง (Dealing with Changes) และทักษะการจัดการความขัดแย้ง (Conflict Management) Rebecca [4] เสนอว่าทักษะการสื่อสาร (Communication) เป็นทักษะที่สำคัญสำหรับการทำงานในรูปแบบสกรัม เนื่องจากการสื่อสารที่ไม่ดีจะทำให้การทำงานของทีมนั้นประสบกับปัญหาในหลาย ๆ ด้าน เนื่องจากทีมสกรัมต้องช่วยกันทำงาน (Collaboration) และเน้นที่ความรวดเร็วในการปฏิบัติงาน นอกจากนี้ผู้วิจัยได้รวบรวมทักษะสำหรับนักพัฒนาซอฟต์แวร์แบบสกรัมเพิ่มเติม (เฉพาะสมาชิกในทีมเท่านั้น ไม่รวม Scrum Master) ดังแสดงในตารางที่ 1

ตารางที่ 1 *ทักษะสำหรับนักพัฒนาซอฟต์แวร์แบบสกรัม*

ทักษะ	[5]	[6]	[7]	[3]	[8]	[9]	[10]	[11]	[12]
1. Acceptance test		✓					✓		
2. Architecture design	✓						✓		
3. Coding standard	✓								
4. Communications		✓	✓		✓	✓			
5. Conversation									✓
6. Conflict management				✓					
7. Detail design								✓	
8. Debugging								✓	
9. Documentation							✓		
10. Inspections							✓		
11. Integration testing				✓			✓		
12. Keeping Code Fit		✓							
13. Modeling							✓		
14. Negotiation									✓
15. OO Design		✓					✓		✓
16. Pair programming		✓							
17. Problem solving					✓				
18. Programming	✓	✓					✓	✓	
19. Self- management					✓				
20. Technical writing									✓

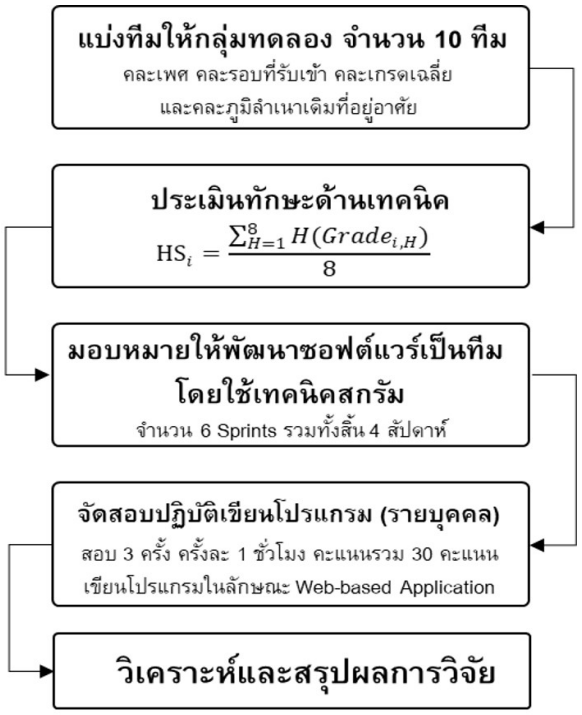
3. วิธีดำเนินการวิจัย

การดำเนินงานวิจัยครั้งนี้ เป็นการวิจัยเชิงประจักษ์ (Empirical study) ซึ่งต้องมีการกำหนดกลุ่มทดลอง รูปแบบการทดลองและการวัด และการประเมินผลการทดลอง โดยมีกรอบแนวคิด ดังแสดงในภาพที่ 1

3.1 กลุ่มทดลอง

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ชั้นปีที่ 2 คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา จำนวน 86 คน ซึ่งทั้งหมดเคยศึกษาและมีประสบการณ์การเขียนโปรแกรมด้วยภาษา C, C++, HTML, CSS, JavaScript, MySQL, PHP และต้องลงทะเบียนเรียนในรายวิชา 1) รายวิชาพื้นฐานที่จำเป็นสำหรับคอมพิวเตอร์ (Computing Essentials) 2) รายวิชาด้านคณิตศาสตร์คอมพิวเตอร์และวิศวกรรมซอฟต์แวร์ (Mathematical and engineering fundamentals) และ 3) รายวิชาด้านการวิเคราะห์ออกแบบ (Software design) รวมทั้งสิ้นจำนวน 8 รายวิชา ดังนี้

1. Logical Thinking and Problem Solving for Innovation
2. Principles of Programming for Software Engineering
3. Principles of Programming for Software Engineering Laboratory
4. Object-Oriented Programming and Modeling
5. Object-Oriented Programming and Modeling Laboratory
6. Data Structure and Algorithms for Software Engineering
7. Object-Oriented System Analysis and Design
8. Personal Software Development Process Laboratory



ภาพที่ 1 *กรอบแนวความคิดในการวิจัย*

โดยจัดแบ่งกลุ่มทดลองเป็นทีม ซึ่งในงานวิจัยนี้จะใช้คำว่า “มกุล” (Cluster) จำนวน 10 มกุล เรียงลำดับจาก มกุล 0 - มกุล 9 แต่ละมกุลมีสมาชิกภายในทีม 7 – 9 คน โดยคละเพศ คละรอบที่รับเข้า คละเกรดเฉลี่ย และคละภูมิลำเนาเดิมที่อยู่อาศัย

3.2 รูปแบบการทดลองและการวัดผล

การทดลองจัดทำขึ้นในค่ายพัฒนาซอฟต์แวร์ ครั้งที่ 9 (Open Source Software Developer Camp#9) ในภาคการศึกษาที่ 2/2563 จำนวน 6 Sprint รวมทั้งสิ้น 4 สัปดาห์ มีพี่เลี้ยงประจำทีม ลักษณะงานที่มอบหมายจะเกี่ยวข้องกับการพัฒนาซอฟต์แวร์ โดยมี Product Owner ซึ่งเป็นตัวแทนสถานประกอบการจริงเป็นผู้ให้ความต้องการ มีการประยุกต์ใช้การพัฒนาแบบสกรัม จัดส่งงานตามวงรอบ และมีข้อกำหนดว่าทุกคนภายในทีมต้องได้รับงานที่ตนรับผิดชอบ ให้อิสระในการแบ่งงานแก่นักศึกษา และกำชับให้แบ่งในลักษณะปริมาณใกล้เคียงกัน มีการเก็บรวบรวมข้อมูลการพัฒนาซอฟต์แวร์และการทดสอบการเขียนโปรแกรมผ่านระบบออนไลน์

การวัดผลทักษะการเขียนโปรแกรมภายหลังการทดลอง จัดทำโดยการจัดสอบปฏิบัติ เพื่อวัดทักษะการเขียนโปรแกรมรายบุคคล ผู้สอบสามารถค้นหาข้อมูลเพิ่มเติมบนอินเทอร์เน็ตได้ มีผู้คุมสอบจำนวนมากเพื่อคอยตรวจสอบและควบคุมไม่ให้มีการทุจริตในการสอบ ในการให้คะแนนแต่ละข้อพิจารณาจากการทำงานของโปรแกรมที่ตรงตามข้อกำหนด เวลาที่ใช้ในการพัฒนา และพิจารณา Source Code ร่วมด้วย

3.3 การประเมินผลการทดลอง

งานวิจัยนี้อนุมานวิธีการประเมินระดับทักษะของกลุ่มทดลองแต่ละคน จากผลการเรียนของรายวิชาที่เกี่ยวข้องกับทักษะด้านเทคนิค (Technical Skills) 8 รายวิชา ตามแนวปฏิบัติของการจำแนกด้านความรู้ (Knowledge areas) ที่นำเสนอโดย Association for Computing Machinery (ACM) ของหลักสูตรวิศวกรรมซอฟต์แวร์ ดังแสดงในหัวข้อ 3.1

อย่างไรก็ตาม ถึงแม้ว่าแต่ละรายวิชาจะมีเกณฑ์การวัดผลที่แตกต่างกัน และอาจจะประเมินความสามารถ หรือวัดความรู้เฉพาะทางในรายวิชานั้น ๆ ร่วมด้วย แต่เนื่องจากกลุ่มทดลองแต่ละคนถูกวัดผลด้วยวิธีเดียวกันในแต่ละรายวิชา การอนุมานระดับทักษะด้านเทคนิค จึงมีความเหมาะสมระดับหนึ่ง ซึ่งคะแนนที่ใช้เป็นฐานในการวิจัยนี้เป็นไปตามสมการที่ 1

$$\mathrm{HS}_i = \frac{\sum_{H=1}^{8} H(Grade_{i,H})}{8} \qquad (1)$$

กำหนดให้

H = {(A,7),(B+,6),(B,5),(C+,4),(C,3),(D+,2),(D,1)}

Grade = เกรดรายวิชาที่เน้นทักษะด้านเทคนิค (Technical Skills) ที่ได้รับ

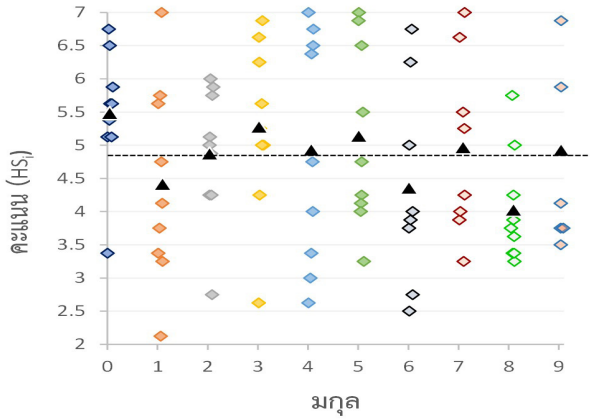
HS_i = คะแนนรวมรายนักศึกษา

สมการที่ 1 เป็นการคำนวณคะแนนรวมผลการเรียนของรายวิชาที่เกี่ยวข้องกับทักษะด้านเทคนิคซึ่งใช้เป็นฐานในการเปรียบเทียบสำหรับการทดลอง โดยกำหนดให้เกรด A ได้คะแนนสูงสุดมีค่าเป็น 7 เกรดแต่ละขั้นมีค่าคะแนนลดลงขั้นละ 1 คะแนน และแต่ละวิชามีน้ำหนักในการคำนวณคะแนนรวมเท่ากัน ดังแสดงในตาราง 2

การกระจายของผลคะแนนแต่ละมกุล มีค่าเฉลี่ยอยู่ที่ 4.84 คะแนน มีค่าฐานนิยม (Mode) เป็น 4.25 คะแนน และมีค่าเบี่ยงเบนมาตรฐาน (S.D.) เป็น 1.33 โดยมกุล 1 มกุล 6 และมกุล 8 มีคะแนนเฉลี่ยทักษะด้านเทคนิคต่ำกว่าคะแนนเฉลี่ยรวมทั้งหมดส่วนมกุลที่เหลือมีคะแนนเฉลี่ยสูงกว่าค่าเฉลี่ยรวมทั้งหมด อย่างไรก็ตาม ถึงแม้ว่า มกุล 2 มกุล 4 มกุล 7 และมกุล 9 จะมีคะแนนเฉลี่ยสูงกว่าค่าเฉลี่ยรวมทั้งหมด แต่อยู่ในระดับที่ใกล้เคียงกับค่าเฉลี่ยมาก ซึ่งห่างกับค่าเฉลี่ยไม่เกิน 0.13 คะแนนโดยคะแนนของสมาชิกแต่ละคนจำแนกตามมกุล ดังแสดงในภาพที่ 2

ตารางที่ 2 *จำนวนสมาชิกและคะแนนเฉลี่ยทักษะฯ แต่ละมกุล*

มกุล	จำนวนสมาชิก : คน	เพศ : คน (ชาย - หญิง)	ภูมิลำเนา : คน (กทม. และปริมณฑล : ต่างจังหวัด)	ค่าเฉลี่ยทักษะฯ
0	9	5:4	2:7	5.49
1	9	5:4	1:8	4.42
2	9	5:4	2:7	4.88
3	9	5:4	1:8	5.28
4	9	5:4	1:8	4.93
5	9	4:5	1:8	5.14
6	8	4:4	3:5	4.35
7	8	4:4	2:6	4.97
8	9	5:4	2:7	4.03
9	7	4:3	1:6	4.93
รวม	**86 คน**	**46:40**	**15:71**	**4.84**

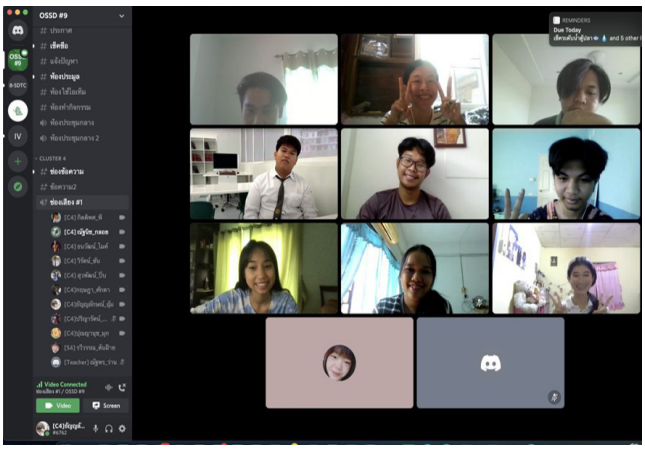


ภาพที่ 2 *การกระจายของคะแนนสมาชิกแต่ละมกุล*

4. ผลการวิจัยและการอภิปรายผล

การทดลองกับกลุ่มตัวอย่างในครั้งนี้จัดทำขึ้นภายในค่ายพัฒนาซอฟต์แวร์ ครั้งที่ 9 (Open Source Software

Developer Camp#9) ของหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ มหาวิทยาลัยบูรพา ปีการศึกษา 2563 โดยใช้ระยะเวลารวมทั้งสิ้น 4 สัปดาห์ กลุ่มทดลองจะได้รับโจทย์ในการพัฒนาซอฟต์แวร์เป็น Web-based Application และต้องออกแบบและพัฒนาโดยใช้แนวคิด Object-Oriented โดยมี Product Owners ซึ่งเป็นตัวแทนจากสถานประกอบการจริงที่มีความร่วมมือกับทางหลักสูตรฯ เป็นผู้ให้ความต้องการ โดยใช้การพัฒนาแบบสกรัม ซึ่งมีข้อกำหนด และกำชับต่อทุกมกุลว่าสมาชิกทุกคนจะต้องได้รับงานปริมาณเท่า ๆ กัน ซึ่งให้อิสระในการแบ่งงานแก่กลุ่มทดลองอย่างไรก็ตามเนื่องจากสถานการณ์โรคระบาดโควิด 19 จึงจำเป็นต้องดำเนินการจัดงานทั้งหมดในลักษณะออนไลน์ (Online) โดยใช้โปรแกรม DISCORD และโปรแกรม ZOOM Meeting เป็นโปรแกรมหลักในการติดต่อสื่อสารดังแสดงในภาพที่ 3

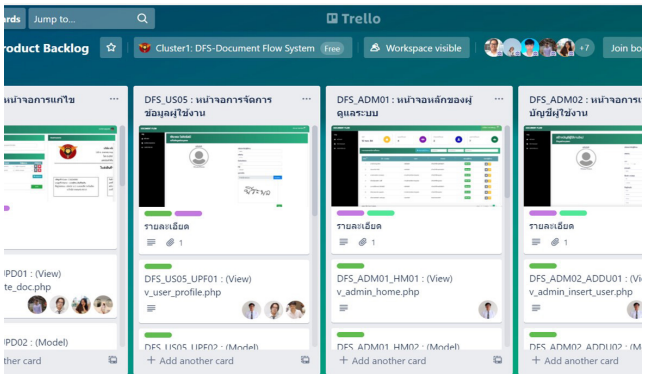


ภาพที่ 3 *การติดต่อสื่อสารระหว่างการปฏิบัติงาน*

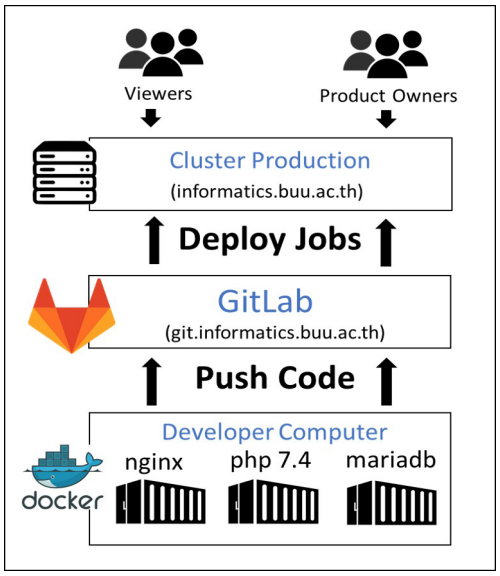
มกุลต้องจัดทำ Product Backlog และนำมาวางแผนงานในแต่ละ Sprint ตลอดจนส่งงานกับ Product Owners ในขั้นตอน Sprint Review ทั้งนี้กำหนดให้ใช้โปรแกรม Trello ในการบริหารจัดการการทำงานภายในมกุลดังภาพที่ 4

ในการพัฒนาซอฟต์แวร์กลุ่มทดลองใช้ภาษา HTML, CSS, JavaScript, และ PHP ในการพัฒนา และใช้ฐานข้อมูล MariaDB ซึ่งกลุ่มทดลองทั้งหมดเคยศึกษาและมีประสบการณ์การเขียนโปรแกรมด้วยภาษาดังกล่าวแล้ว

กลุ่มทดลองถูกกำหนดให้มีสิ่งแวดล้อมในการพัฒนาเหมือน ๆ กัน คือ ติดตั้งโปรแกรม Docker เพื่อจำลองสิ่งแวดล้อมภายในคอมพิวเตอร์ส่วนตัวของกลุ่มทดลองให้มีเครื่องมือที่จำเป็นต้องใช้ในการพัฒนาซอฟต์แวร์ และใช้โปรแกรมควบคุมการเปลี่ยนแปลง (Version control) โดยงานที่เสร็จสิ้นและได้รับการยอมรับจะถูกนำมาแสดงในแอปพลิเคชันหลักของแต่ละมกุล ดังแสดงในภาพที่ 5



ภาพที่ 4 *การติดต่อสื่อสารระหว่างการปฏิบัติงาน*



ภาพที่ 5 *สถาปัตยกรรมการพัฒนาซอฟต์แวร์*

4.1 ผลการวัดทักษะด้านเทคนิคหลังการทดลอง

ผู้วิจัยดำเนินการจัดสอบด้านเทคนิคแบบปฏิบัติ เพื่อวัดทักษะด้านเทคนิครายบุคคลหลังจบโปรแกรมค่ายพัฒนาซอฟต์แวร์ กลุ่มทดลองสามารถค้นหาข้อมูลเพิ่มเติมบนอินเทอร์เน็ตได้ มีผู้คุมสอบจำนวนมากเพื่อคอยตรวจสอบและควบคุมไม่ให้มีการทุจริตในการสอบ โดยจัดให้มีการสอบ 3 ครั้ง มีเวลาในการทำข้อสอบครั้งละ 1 ชั่วโมง โดยข้อสอบมีคะแนนรวม 30 คะแนน ซึ่งโจทย์ในการสอบคือการเขียนโปรแกรมในลักษณะ Web-based Application โดยผู้วิจัยกำหนดข้อสอบ

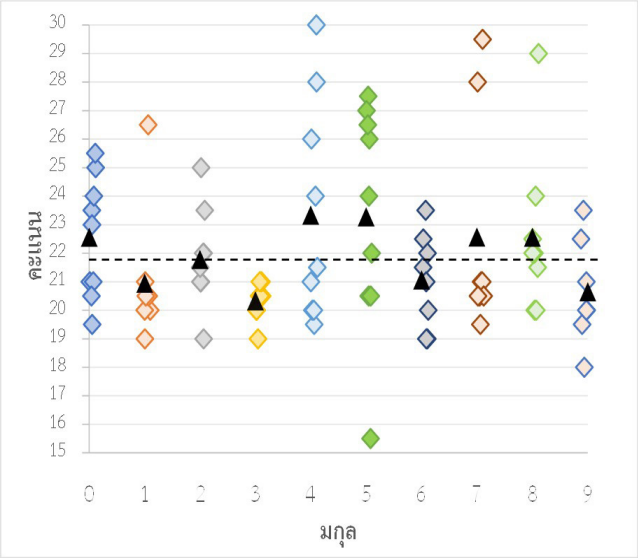
มี 3 ลักษณะ คือ 1) ข้อสอบมีความใกล้เคียงกับโจทย์ที่ทั้ง 10 มกุลได้พัฒนาเป็นอย่างมาก 2) ข้อสอบมีความใกล้เคียงกับโจทย์ แต่ต้องการประยุกต์หรือดัดแปลงบางส่วน และ 3) ข้อสอบเป็นโจทย์ที่ทั้ง 10 มกุล ไม่เคยพัฒนามาก่อน ซึ่งผลการสอบเป็นดังตารางที่ 3

ตารางที่ 3 *คะแนนเฉลี่ยและหลังปฏิบัติงานแต่ละมกุล*

ก่อนปฏิบัติงาน ค่าฐานนิยม (Mode) เป็น 4.25 คะแนน			หลังปฏิบัติงาน ค่าฐานนิยม (Mode) เป็น 20.5 คะแนน	
มกุล	คะแนนเฉลี่ย	S.D.	คะแนนเฉลี่ย	S.D.
0	5.49	0.92	22.56	2.01
1	4.42	1.42	20.95	2.03
2	4.88	0.97	21.78	1.60
3	5.28	1.23	20.33	0.58
4	4.93	1.65	23.33	3.64
5	5.14	1.30	23.28	3.76
6	4.35	1.44	21.06	1.53
7	4.97	1.27	22.56	3.62
8	4.03	0.80	22.56	2.55
9	4.93	1.36	20.64	1.73
รวม	**4.84**	**1.33**	**21.93**	**2.47**

จากตารางที่ 3 ผลการสอบปฏิบัติของกลุ่มทดลองมีคะแนนเฉลี่ยรวมเท่ากับ 21.93 คะแนน มีฐานนิยม (Mode) เป็น 20.5 คะแนน และมีค่าเบี่ยงเบนมาตรฐาน (S.D.) เท่ากับ 2.47 เมื่อพิจารณาที่มกุล 3 และมกุล 9 พบว่ามีคะแนนเฉลี่ยประจำมกุลน้อยกว่าค่าเฉลี่ยรวมทั้งหมดซึ่งตรงกันข้ามกับก่อนปฏิบัติงานที่มีคะแนนเฉลี่ยประจำมกุลมากกว่าคะแนนเฉลี่ยรวมทั้งหมด ในทางกลับกัน มกุล 8 มีคะแนนเฉลี่ยประจำมกุลสูงกว่าคะแนนเฉลี่ยรวมทั้งหมด ในขณะที่ก่อนปฏิบัติงานมีคะแนนเฉลี่ยรวมประจำมกุลน้อยกว่าค่าเฉลี่ยรวมทั้งหมด โดยคะแนนสมาชิกแต่ละมกุลแสดงดังภาพที่ 6

จากภาพที่ 6 เป็นที่น่าสนใจอย่างยิ่ง เมื่อพิจารณาผลสอบการเขียนโปรแกรมของมกุล 0 มกุล 3 และมกุล 5 ซึ่งเป็นมกุลที่ก่อนปฏิบัติงานภาพรวมสมาชิกส่วนใหญ่มีคะแนนสูงกว่าค่าเฉลี่ย ซึ่งอยู่ใน 3 มกุลแรกที่ทำคะแนนได้สูงที่สุด ซึ่งอนุมานได้ว่าเป็นมกุลที่สมาชิกส่วนใหญ่มีทักษะด้านเทคนิคระดับสูงอยู่หลายคน แต่ในทางกลับกัน ผลสอบหลังปฏิบัติงานพบว่า มกุล 3 ภาพรวมสมาชิกส่วนใหญ่มีคะแนนน้อยกว่าค่าเฉลี่ยอย่างชัดเจน ส่วนมกุล 0 และมกุล 5 ถึงแม้ว่าภาพรวมสมาชิกส่วนใหญ่จะยังคงมีคะแนนสูงกว่าค่าเฉลี่ย แต่เมื่อพิจารณาเป็นรายบุคคลพบว่า มีจำนวนสมาชิกที่มีคะแนนน้อยกว่าค่าเฉลี่ยเพิ่มมากขึ้น



ภาพที่ 6 *การกระจายคะแนนหลังปฏิบัติงานแต่ละมกุล*

มกุล 2 มกุล 4 มกุล 7 และมกุล 9 ซึ่งอนุมานว่าเป็นมกุลที่สมาชิกมีทักษะด้านเทคนิคระดับสูง และระดับต่ำอยู่เท่า ๆ กัน เนื่องจากผลคะแนนก่อนปฏิบัติงานของมกุลอยู่ใกล้เคียงกับค่าเฉลี่ยของกลุ่มทดลองทั้งหมด ซึ่งผลการสอบหลังการปฏิบัติงานพบว่า มีผลสอบใกล้เคียงกับค่าเฉลี่ยเช่นเดิม ยกเว้นมกุล 9 ที่มีคะแนนประจำมกุลต่ำกว่าค่าเฉลี่ย

มกุล 1 มกุล 6 และมกุล 8 ก่อนปฏิบัติงานสมาชิกส่วนใหญ่มีคะแนนทักษะด้านเทคนิคต่ำกว่าค่าเฉลี่ย ซึ่งอยู่ใน 3 มกุลที่ทำคะแนนได้น้อยที่สุด อนุมานได้ว่าเป็นมกุลที่สมาชิกส่วนใหญ่มีทักษะด้านเทคนิคระดับต่ำหลายคน พบว่า สมาชิกมกุล 8 สามารถพัฒนาทักษะและทำคะแนนเฉลี่ยประจำมกุลได้สูงกว่าค่าเฉลี่ยทั้งหมดของกลุ่มทดลอง ส่วนมกุล 1 และมกุล 6 ยังคงมีค่าเฉลี่ยน้อยกว่าค่าเฉลี่ยทั้งหมดของกลุ่มทดลอง

4.2 ผลการวิเคราะห์ทางสถิติ

ผู้วิจัยดำเนินการวิเคราะห์ข้อมูลโดยใช้หลักทางสถิติ One-way ANOVA และการวิเคราะห์ค่าเฉลี่ยรายคู่แบบ Bonferroni method พบว่า ไม่มีความแตกต่างอย่างมีนัยสำคัญที่ 0.05 ของผลคะแนนกับมกุลดังแสดงในตารางที่ 4

ตารางที่ 4 *การวิเคราะห์ความแปรปรวนของคะแนน*

คะแนน	df	Mean Square	F	Sig.
Between Groups	9	10.425	1.442	.186
Within Groups	76	7.231		
Total	85			

5. สรุป

งานวิจัยนี้นำเสนอการจัดทีมสกรัมโดยพิจารณาจากทักษะด้านเทคนิคเป็นฐาน และผลกระทบต่อการพัฒนาทักษะการเขียนโปรแกรมในกลุ่มผู้พัฒนาระดับเริ่มต้น โดยมีกลุ่มทดลอง จำนวน 86 คน แบ่งเป็น 10 กลุ่ม แต่ละคนจะถูกวัดระดับทักษะด้านเทคนิคจากผลการเรียนในรายวิชาของสาขาวิชาวิศวกรรมซอฟต์แวร์ กลุ่มทดลองต้องพัฒนา Web-based Application และใช้วิธีสกรัม (Scrum) ในการพัฒนาซอฟต์แวร์ และทำการวัดผลการพัฒนาทักษะการเขียนโปรแกรมจากการสอบปฏิบัติ ผลที่ได้รับจากการวิจัยคือไม่สามารถสรุปผลทางสถิติอย่างมีนัยสำคัญที่ 0.05 ได้ว่ารูปแบบการจัดทีมสกรัมโดยพิจารณาจากทักษะด้านเทคนิคเป็นฐาน มีผลต่อคะแนนสอบปฏิบัติ อย่างไรก็ตาม จากการสัมภาษณ์ (Interview) กลุ่มทดลองส่วนใหญ่ให้ความเห็นว่าความพยายามในการทำงานที่มีความยากขึ้น ส่งผลดีต่อการพัฒนาทักษะในการพัฒนาซอฟต์แวร์ การปฏิบัติงานซ้ำ ๆ เดิมทำให้เกิดความชำนาญ ใช้เวลาน้อยลงในครั้งถัด ๆ ไป และสามารถส่งผลดีด้านคุณภาพของซอฟต์แวร์ ซึ่งสอดคล้องกับผลการทดลองของ Tan และ Teo [3]

ทั้งนี้ ผู้วิจัยพบข้อสังเกตเพิ่มเติมว่า มกุลที่สมาชิกส่วนใหญ่มีทักษะด้านเทคนิคระดับสูงอยู่หลายคน (ร้อยละ 60 – 90 ของสมาชิกทั้งหมด) ได้คะแนนสอบหลังปฏิบัติงานน้อยกว่าค่าเฉลี่ยรวม ซึ่งตรงข้ามกับก่อนปฏิบัติงานที่สามารถทำได้สูงกว่าค่าเฉลี่ยรวม

อย่างไรก็ตาม เนื่องจากกลุ่มทดลองเป็นนักศึกษาในระดับอุดมศึกษา และศึกษาอยู่ในหลักสูตรเดียวกัน อาจมีภาพรวมของระดับความรู้ด้านการพัฒนาซอฟต์แวร์ที่ใกล้เคียงกัน ซึ่งอาจเป็นเหตุให้ไม่มีผลต่อการพัฒนาทักษะการเขียนโปรแกรม ทั้งนี้ผู้วิจัยเสนอแนวทางการทำวิจัยเพิ่มเติมในอนาคต จำนวน 3 ประเด็น ได้แก่ 1) ควรทำการทดลองเพิ่มเติมกับกลุ่มทดลองที่สำเร็จการศึกษาและได้รับเข้าปฏิบัติงานในสถานประกอบการแล้ว 2) ควรศึกษาและปรับปรุงวิธีการวัดระดับทักษะที่มีความแม่นยำและเที่ยงตรงมากขึ้น และ 3) นอกจากกำชับว่าสมาชิกทุกคนภายในทีมจะต้องได้รับงานปริมาณเท่า ๆ กัน อาจต้องกำหนดปัจจัยอื่น ๆ ร่วมด้วย เช่น ความยาก/ง่ายของงาน ความหลากหลายของงาน และคุณภาพของงาน เป็นต้น

6. เอกสารอ้างอิง

[1] A. Cockburn, and J. Highsmith. “Agile Software Development: The People Factor”, *Computer,* Vol. 34, Issue 11, pp. 131-133, 2001.

[2] M. Ardis et al. "SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering.", *IEEE Comput*, Vol. 48, No. 11, pp. 106-109, November, 2015.

[3] C. Tan, and H. Teo, “Training Future Software Developers to Acquire Agile Development Skills.”, *Communication of The ACM*, Vol. 50, No. 12, pp. 97-98. 2007.

[4] R. J. Wirfs-Brock. “Designing with an Agile Attitude.” *IEEE Software Magazine,* Vol. 26, Issue 2, pp. 68-69, 2009.

[5] M. Vitolo, J. Brinkman, M. Vernaza, and E. Steinbrink. “Serving with Engineering Skills within 15 Miles of Campus: The Scholars of Excellence in Engineering and Computer Science Program.”, *Frontiers in Education Conference (FIE),* pp. 12-15 October, USA, 2016.

[6] K. Tong. “Essential Skills for Agile Development.”, *Essential Skills for Agile Development,* June, 2004.

[7] A. Dagnino. “An Evolutionary Lifecycle Model with Agile Practices for Software Development at ABB.”, *the Eighth IEEE international Conference on Engineering of Complex Computer Systems*, September, 2002.

[8] A. Masek, and S. Yamin., “Nurturing Personal Skills Development: Model of monitoring and assessment of student centered learning.”, *International Conference on Engineering Education,* Malaysia, Kuala Lumpur, 7-8 December, 2009.

[9] G. Duffy, and B. Bowe., "A strategy for the development of lifelong learning and personal skills throughout an undergraduate engineering programme.", *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, 6-9 April 2010.

[10] G. Beranek, and B. Bowe. "Functional Group Roles in Software Engineering Teams", *HSSE '05 Proceedings of the 2005 workshop on Human and social factors of software engineering,* 16 May 2005.

[11] A. Tiwana. "An empirical study of the effect of knowledge integration on software development performance.", *Technology Information and Software,* Vol. 46, pp. 899-906, 2004.

[12] B. Kovitz. "Hidden Skills that Support Phased and Agile Requirements Engineering.", *Requirements Engineering,* pp. 135-141, 2003.