



รายงานวิจัยฉบับสมบูรณ์
โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรม
การพัฒนาซอฟต์แวร์
(The Study and Collect of Necessary Skills for Practitioners in
The Software Development Industry)

นายอภิสิทธิ์ แสงใส

นางสาวณัฐพร ภัคดี

หัวหน้าโครงการ

ผู้ร่วมวิจัย

โครงการวิจัยประเภทเงินรายได้

คณะวิทยาการสารสนเทศ

ประจำปีงบประมาณ พ.ศ. 2564

มหาวิทยาลัยบูรพา

รายงานวิจัยฉบับสมบูรณ์
โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์
(The Study and Collect of Necessary Skills for Practitioners in
The Software Development Industry)

นายอภิสิทธิ์ แสงใส

หัวหน้าโครงการ

นางสาวณัฐพร ภัคดี

ผู้ร่วมวิจัย

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

มิถุนายน 2565

กิติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนการวิจัยจากเงินรายได้ คณะวิทยาการสารสนเทศ ประจำปีงบประมาณ
2564 เลขที่สัญญา 8/2564

สารบัญ

หน้า

บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ขอบเขตของการวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 แผนการดำเนินงานวิจัย	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ทักษะ (Skill) และทัศนคติ (Attitude).....	4
2.2 กระบวนการพัฒนาซอฟต์แวร์ทั่วไป	5
2.3 กระบวนการพัฒนาซอฟต์แวร์แบบสกรัม (Scrum).....	9
บทที่ 3 วิธีดำเนินการวิจัย	16
3.1 การกำหนดปัญหาและรวบรวมข้อมูล	17
3.2 การสังเคราะห์ผล.....	26
3.3 การวิพากษ์โดยผู้เชี่ยวชาญ	33
บทที่ 4 สรุปผลการดำเนินงาน.....	34
4.1 ผลการวิพากษ์โดยผู้เชี่ยวชาญ	34
4.2 สรุปทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์.....	35
4.3 ทักษะด้านเทคนิค (Technical skills).....	36
4.4 ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills).....	40
4.5 ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)	43
บทที่ 5 สรุปผลการดำเนินงาน.....	45
5.1 สรุปผลการดำเนินงาน	45
5.2 ข้อจำกัดของงานวิจัย	45

สารบัญ (ต่อ)

	หน้า
5.3 ข้อเสนอแนะในอนาคต.....	46
บรรณานุกรม.....	47
ภาคผนวก	51

สารบัญภาพ

หน้า

ภาพที่ 2.1 วงรอบ/วัฏจักรการพัฒนาซอฟต์แวร์.....	5
ภาพที่ 2.2 กระบวนการวิศวกรรมความต้องการ [4].....	5
ภาพที่ 2.3 The Agile Scrum Framework [5].....	9
ภาพที่ 3.1 ขั้นตอนการวิจัย.....	16
ภาพที่ 3.2 การค้นหาข้อมูลที่เกี่ยวข้องกับงานวิจัย.....	18
ภาพที่ 4.1 สรุปทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์.....	35

สารบัญตาราง

หน้า

ตารางที่ 2.1 กระบวนการวิศวกรรมความต้องการ	6
ตารางที่ 2.2 ระดับการทดสอบซอฟต์แวร์ (Level Software Testing)	7
ตารางที่ 2.3 ประเภทของการบำรุงรักษาซอฟต์แวร์	8
ตารางที่ 2.4 ทักษะทางด้านเทคนิค และทักษะที่เห็นได้ทางด้านพฤติกรรม.....	13
ตารางที่ 3.1 ทักษะและทัศนคติที่ได้จากการรวบรวมข้อมูล.....	19
ตารางที่ 3.2 การรวบรวมทักษะและทัศนคติที่ได้ในแต่ละแหล่งข้อมูล.....	26
ตารางที่ 3.3 ประเภทของทักษะและทัศนคติที่ผู้วิจัยเสนอเพื่อตอบคำถามงานวิจัย	31
ตารางที่ 3.4 ประเภทผู้เชี่ยวชาญวิพากษ์ผลการสังเคราะห์ของงานวิจัย	33
ตารางที่ 4.1 ข้อมูลจากผู้เชี่ยวชาญ	34
ตารางที่ 4.3 ทักษะด้านเทคนิค (Technical skills).....	36
ตารางที่ 4.4 ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills).....	40
ตารางที่ 4.5 ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)	43

บทที่ 1 บทนำ

1.1 ความสำคัญและความเป็นมาของโครงการ

ลักษณะเด่นของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) คือ การให้ความสำคัญกับทีมพัฒนาร่วมมือกันทำงาน (Collaboration) ด้วยความเร็ว (Fast-paced) และคำนึงถึงความพึงพอใจของลูกค้า โดยเฉพาะอย่างยิ่งการพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) ซึ่งได้รับความนิยม และถูกนำมาใช้อย่างแพร่หลาย โดยสมาชิกของทีมไม่ได้ถูกกำหนดหน้าที่กันอย่างชัดเจนตายตัว สมาชิกของทีมสามารถบริหารจัดการตนเองได้ ส่งผลให้สมาชิกทุกคนจำเป็นต้องมีทักษะที่เพียงพอต่อการปฏิบัติงาน

ทั้งนี้ แมว่าองค์กรจะใช้วิธีการพัฒนาซอฟต์แวร์ที่ได้รับการยอมรับ แต่ยังคงพบกับความล้มเหลวของการพัฒนาซอฟต์แวร์อยู่บ่อยครั้ง จากงานวิจัยที่เกี่ยวข้องพบว่า สมาชิกภายในทีม (Team members) ยังคงเป็นปัจจัยหลักในหลายๆ ปัจจัยความสำเร็จ ที่ส่งผลต่อการพัฒนาซอฟต์แวร์ โดยเฉพาะอย่างยิ่งปัญหาการขาดทักษะที่จำเป็นสำหรับการพัฒนาซอฟต์แวร์ เนื่องจากยังไม่มีข้อกำหนดกรอบทักษะที่จำเป็นสำหรับการพัฒนาซอฟต์แวร์ ด้วยกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ไว้อย่างชัดเจน มีเพียงร่างแนวทางเบื้องต้นที่นำไปสู่การบริหารจัดการตนเองได้เท่านั้น

งานวิจัยนี้ศึกษา รวบรวม และนำเสนอทักษะที่จำเป็น สำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ เพื่อให้ผู้ปฏิบัติงานสามารถเลือกทักษะดังกล่าวไปใช้ในการพัฒนาตนเอง โดยผู้วิจัยทำการศึกษา และรวบรวมทักษะต่างๆ จากงานวิจัยที่เกี่ยวข้อง พร้อมทั้งจำแนกทักษะออกเป็น 3 ประเภทหลัก ได้แก่ ทักษะด้านเทคนิค (Technical skills) ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) และทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์
2. เพื่อนำเสนอทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์
3. เพื่อให้ผู้ปฏิบัติงานสามารถเลือกทักษะที่จำเป็นไปใช้ในการพัฒนาตนเอง

1.3 ขอบเขตของการวิจัย

1. ศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) จากวารสารและงานวิจัยที่เกี่ยวข้อง
2. นำเสนอทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ โดยแบ่งออกเป็น 3 ทักษะหลัก ได้แก่ ทักษะด้านเทคนิค (Technical skills) ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) และทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. รับรู้/รับทราบทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์
2. ผู้สอนสามารถนำทักษะในงานวิจัยนี้ ไปใช้ในการจัดการเรียนการสอน เพื่อส่งเสริมให้ผู้เรียนมีทักษะที่ตรงต่อความต้องการกับภาคอุตสาหกรรมการพัฒนาซอฟต์แวร์
3. ผู้เรียนสามารถนำทักษะในงานวิจัยนี้ ไปพัฒนาทักษะของตนเองที่ยังไม่เชี่ยวชาญ เพื่อเพิ่มความชำนาญในการปฏิบัติงาน

1.5 แผนการดำเนินงานวิจัย

หัวข้อดำเนินงาน/ระยะเวลาดำเนินการ	2564					2565						
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.
1. วางแผนและกำหนดแนวทางการดำเนินงานวิจัย	/											
2. ศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum)		/	/	/	/							
3. วิเคราะห์ ปรับปรุงแก้ไข จัดกลุ่ม และจำแนกทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์						/	/	/				
4. กำหนดนิยาม/ความหมาย (Definition of terms) ของแต่ละทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ เพื่อความชัดเจนในการอธิบายความหมายของแต่ละทักษะ									/			
5. ศึกษารายงานผลการวิเคราะห์ โดยผู้เชี่ยวชาญหรือผู้ทรงคุณวุฒิ และนำเสนอผลการวิเคราะห์ให้แก่คณาจารย์หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา อย่างน้อย 1 ครั้ง										/		
6. เขียนรายงานรูปแบบฉบับสมบูรณ์นำเสนอทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ทั้งทักษะด้านเทคนิค (Technical skills) ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) และทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)											/	/

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

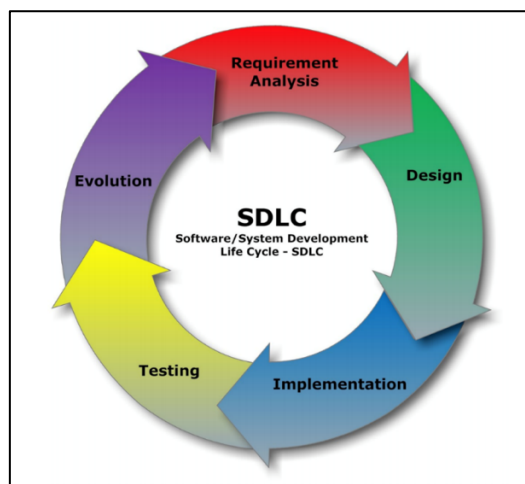
2.1 ทักษะ (Skill) และทัศนคติ (Attitude)

ตามพระราชบัญญัติส่งเสริมการพัฒนาฝีมือแรงงาน [1] กำหนดให้ทักษะคือ “การสะสมประสบการณ์จนเกิดเป็นความชำนาญ มีความสามารถเพียงพอที่จะทำงานได้อย่างมีคุณภาพตามมาตรฐานที่กำหนด” และอธิบายทัศนคติ (Attitude) ไว้ว่า “เป็นจิตสำนึกในการปฏิบัติงานที่ดี ตระหนักถึงความปลอดภัยในการปฏิบัติงาน คำนึงถึงต้นทุนในการผลิต และการบริการที่เกิดจากการสูญเสียของวัสดุ และอุปกรณ์ระหว่างการปฏิบัติงาน เป็นต้น” สอดคล้องกับ Spencer [2] ที่อธิบายทักษะไว้ว่า “ความสามารถในการปฏิบัติงานทั้งที่เกี่ยวข้องกับด้านกายภาพ การใช้ความคิดจิตใจของบุคคล” และอธิบายทัศนคติไว้ว่า “เป็นแรงจูงใจที่ทำให้เกิดพฤติกรรม” และเป็นไปในทิศทางเดียวกับพจนานุกรม ฉบับราชบัณฑิตยสถาน [3] อธิบายว่าทักษะ หมายถึง “ความชำนาญหรือความสามารถในการกระทำหรือการปฏิบัติอย่างใดอย่างหนึ่งซึ่งอาจเป็นทักษะด้านร่างกายสติปัญญา หรือสังคม ที่เกิดขึ้นจากการฝึกฝนหรือการกระทำบ่อย ๆ” ตัวอย่างเช่น ครูมีทักษะการใช้คำถาม การนำเข้าสู่บทเรียนการใช้สื่อการสอน นักเรียนมีทักษะ การฟัง พูด อ่าน เขียน การคิดคำนวณหรือทักษะทางสังคม ทักษะที่จะทำให้บุคคลประสบความสำเร็จในการดำเนินชีวิตและการทำงาน โดยรวมแล้วประกอบด้วย สมรรถนะทักษะ (Hard skill) และ จรณทักษะ (Soft skill) บุคคลที่มี จรณทักษะ จะเป็นผู้ปฏิบัติงานได้ดี ส่วนบุคคลที่มี สมรรถนะทักษะ จะเป็นผู้ปฏิบัติงานได้เก่ง

ซึ่งจากแนวคิดและทฤษฎีเกี่ยวกับทักษะ (Skill) และทัศนคติ (Attitude) ดังกล่าวข้างต้น พิจารณาและให้ความหมายของคำว่าทักษะในงานวิจัยนี้คือ การกระทำหรือพฤติกรรมที่เห็นได้ถึงความชำนาญ ซึ่งความชำนาญดังกล่าวเกิดจากการมีความรู้ความเข้าใจในเรื่องนั้นๆ อย่างถูกต้อง และนำไปประยุกต์ใช้ ฝึกฝนจนเกิดความคล่องแคล่ว โดยแบ่งเป็นทั้งทักษะด้านเทคนิค (Hard Skill / Technical skill) หมายถึงทักษะที่เกี่ยวข้องและส่งผลต่อการปฏิบัติงานโดยตรง และทักษะการเป็นผู้ปฏิบัติงานที่ดี (Soft Skill / Non-technical Skill) หมายถึง ทักษะที่ช่วยสนับสนุนและส่งเสริมการทำงานให้มีประสิทธิภาพมากขึ้น และทัศนคติ (Attitude) หมายถึง แนวคิด และมุมมองที่ดีต่อการปฏิบัติงาน เพื่อส่งเสริมการทำงานให้มีความราบรื่นมากขึ้น

2.2 กระบวนการพัฒนาซอฟต์แวร์ทั่วไป

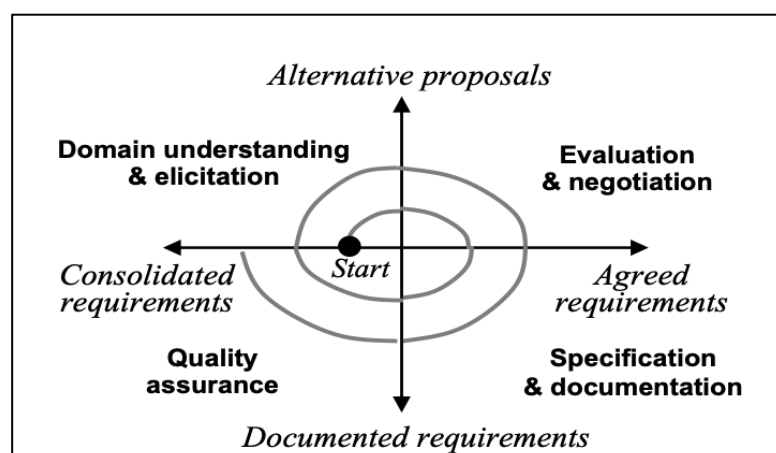
การพัฒนาซอฟต์แวร์เป็นกระบวนการที่เกี่ยวข้องกับการรวบรวมความต้องการ ออกแบบ เขียนโปรแกรม ทดสอบซอฟต์แวร์ และการบำรุงรักษา โดยภาพที่ 2.1 แสดงกระบวนการหรือกิจกรรมต่างๆ ภายในวงรอบของการพัฒนาซอฟต์แวร์ (Software Development Life Cycle : SDLC) โดยที่การพัฒนาซอฟต์แวร์/วิศวกรรมซอฟต์แวร์ จะใช้หลักการเหล่านี้ในการพัฒนาซอฟต์แวร์ ซึ่งประกอบไปด้วย



ภาพที่ 2.1 วงรอบ/วัฏจักรการพัฒนาซอฟต์แวร์

2.2.1 การเก็บรวบรวมและวิเคราะห์ความต้องการ (Requirement Analysis)

กระบวนการในการเก็บรวบรวมข้อมูลปัญหาและความต้องการที่มีต่อการพัฒนาซอฟต์แวร์ จากทั้งเอกสาร (Document) แบบสอบถาม (Questionnaire) การสัมภาษณ์ (Interview) หรือการสังเกต (Observation) และนำความต้องการเหล่านั้นมาวิเคราะห์ โดยจัดทำเป็นขอบเขตและวัตถุประสงค์ของระบบ นำไปสู่การประมาณการและวางแผนในการพัฒนาซอฟต์แวร์ โดยมีกระบวนการในภาพที่ 2.2 และอธิบายกระบวนการในตารางที่ 2.1



ภาพที่ 2.2 กระบวนการวิศวกรรมความต้องการ [4]

ตารางที่ 2.1 กระบวนการวิศวกรรมความต้องการ

กระบวนการวิศวกรรมความต้องการ	รายละเอียด
Domain understanding & elicitation	<ul style="list-style-type: none"> - ศึกษาขอบเขตการทำงานของระบบเดิม (System-as-is) - หาหรือระบุสาเหตุของปัญหาต่างๆ ที่เกิดจากระบบเดิมให้ได้ - ระบุจุดอ่อนและจุดแข็ง (Weaknesses & strengths) ของ System-as-is - กำหนดผู้มีส่วนได้ส่วนเสียทั้งหมด
Evaluation & negotiation	<ul style="list-style-type: none"> - วัตถุประสงค์ในการทำกิจกรรมนี้ คือ ให้มีการตัดสินใจในประเด็นต่างๆ ที่เกิดขึ้นระหว่างกระบวนการ Elicitation - การตัดสินใจนี้จำเป็นต้องใช้การวิเคราะห์ การเปรียบเทียบ การประเมิน การเจรจาต่อรอง (Negotiation) การจัดลำดับความสำคัญ (Prioritization) การลงมติ เพื่อหาข้อยุติจาก Stakeholders
Specification & documentation	<ul style="list-style-type: none"> - กิจกรรมนี้ประกอบไปด้วยรายละเอียด โครงสร้างและเอกสาร ที่เป็นผลลัพธ์ของขั้นตอน Evaluation - จัดทำเอกสารความต้องการ (Requirement document : RD) ซึ่งประกอบไปด้วย <ul style="list-style-type: none"> • วัตถุประสงค์ (Objectives) • คุณสมบัติของโดเมนที่เกี่ยวข้อง (Relevant domain properties) • ระบบ (System) • ความต้องการของซอฟต์แวร์ (Software requirements) • นิยามแนวคิด (Concept definitions) • ความรับผิดชอบ (Responsibilities) • ความต้องการของระบบ (System requirements) • สมมติฐานที่เกี่ยวข้อง (Environmental assumptions)
Quality assurance	<ul style="list-style-type: none"> - การตรวจสอบความถูกต้อง (Validate) ของ Specification กับ Stakeholders ต่างๆ

2.2.2 การออกแบบซอฟต์แวร์ (Design)

กระบวนการในการออกแบบทั้งส่วนการทำงาน (System design) และส่วนหน้าจอรระบบ (UX/UI) ให้สอดคล้องกับความต้องการของลูกค้า ทั้งนี้นิยมใช้แผนภาพ (Diagram) ต่างๆ เข้ามาช่วยในการออกแบบ เช่น Use case diagram, Activity diagram, Class diagram, Sequence diagram และ Entity-Relation diagram เป็นต้น

2.2.3 การลงมือพัฒนาซอฟต์แวร์ตามที่ได้ออกแบบไว้ (Implementation)

กระบวนการในการเขียนโปรแกรมให้สอดคล้องกับความต้องการและการออกแบบซอฟต์แวร์ โดยใช้เครื่องมือและภาษาต่างๆ

2.2.4 การทดสอบซอฟต์แวร์ (Testing)

กระบวนการในการวิเคราะห์ ตรวจสอบ และติดตามผล การพัฒนาซอฟต์แวร์ เพื่อให้แน่ใจว่าซอฟต์แวร์ที่จะส่งมอบงานมีความถูกต้อง สมบูรณ์ และมีประสิทธิภาพตามที่ผู้ใช้งานคาดหวังไว้ โดยแบ่งระดับการทดสอบซอฟต์แวร์ (Level software testing) เป็น 4 ระดับได้แก่ การทดสอบโมดูล การทดสอบภาพรวม การทดสอบฟังก์ชัน การทดสอบประสิทธิภาพ การทดสอบการยอมรับ และการทดสอบการติดตั้งระบบ

ตารางที่ 2.2 ระดับการทดสอบซอฟต์แวร์ (Level Software Testing)

ระดับการทดสอบ	รายละเอียด
การทดสอบโมดูลหรือทดสอบหน่วย (Module testing หรือ Unit testing)	เป็นการทดสอบเกี่ยวกับการทำงานของแต่ละฟังก์ชันโดยละเอียด เพื่อดูว่าแต่ละฟังก์ชันสามารถทำงานได้ถูกต้องหรือไม่ โดยจะต้องทดสอบแต่ละฟังก์ชันอิสระต่อกัน
การทดสอบภาพรวม (Integration testing)	เป็นการทดสอบการทำงานฟังก์ชันต่างๆ ที่สัมพันธ์กันภายในระบบย่อยของโปรแกรม ว่ามีการทำงานได้อย่างมีประสิทธิภาพหรือไม่ โดยการทดสอบนี้จะต้องอ้างอิงถึงขั้นตอนการออกแบบระบบหรือโปรแกรม เพื่อให้รู้ว่าภายในระบบหรือโปรแกรมนั้นมีระบบย่อย ใดๆบ้าง
การทดสอบระบบ (System testing)	เป็นการทดสอบระบบหรือโปรแกรมโดยดูภาพรวมของการทำงาน ว่ามีการตอบสนองความต้องการทั้งในส่วนของฟังก์ชันการทำงาน และประสิทธิภาพการทำงาน ว่าสอดคล้องกับลักษณะของความต้องการของซอฟต์แวร์ (Requirement specification) หรือไม่
การทดสอบการยอมรับ (Acceptance testing)	เป็นการเปรียบเทียบประสิทธิภาพของระบบกับความคาดหวังของลูกค้า โดยตรวจสอบจากเอกสารกำหนดความต้องการ

2.2.5 การวิวัฒนาการซอฟต์แวร์ (Evolution)

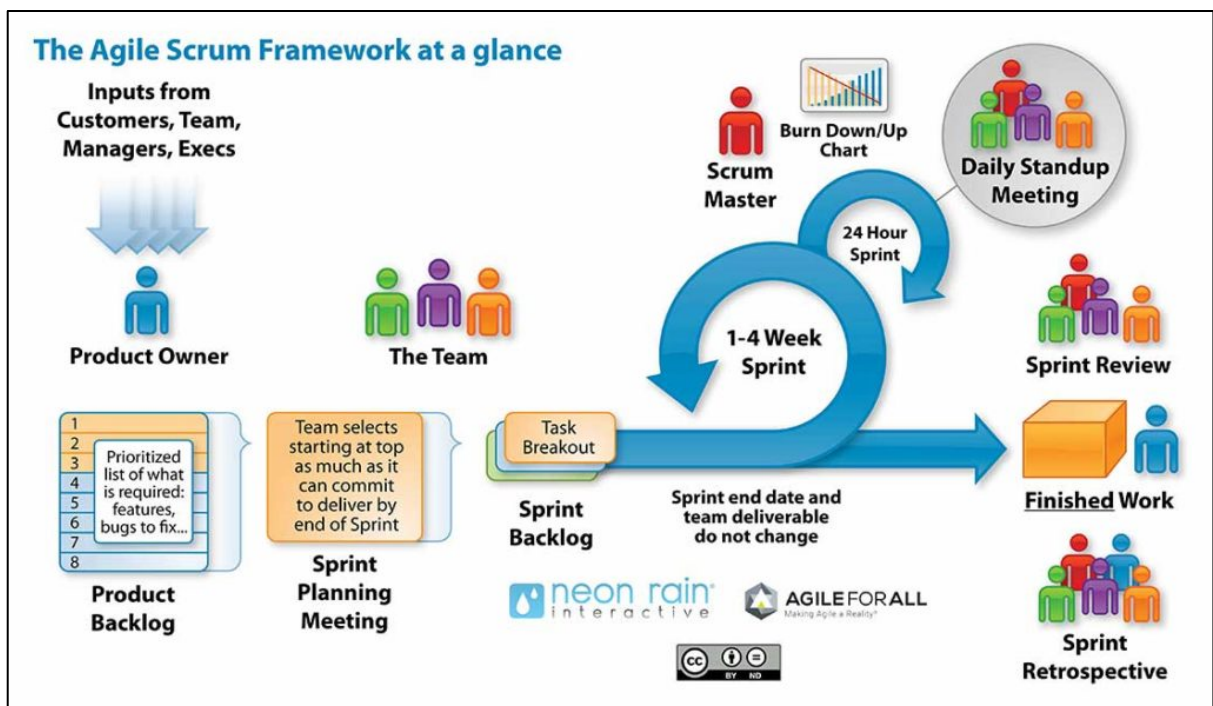
กระบวนการในการติดตามและตรวจสอบการเปลี่ยนแปลงซอฟต์แวร์ภายหลังจากการใช้งาน เพื่อแก้ไขปัญหาหรือข้อผิดพลาดและปรับปรุงซอฟต์แวร์ให้มีประสิทธิภาพ โดยมีประเภทของการบำรุงรักษาซอฟต์แวร์ ดังต่อไปนี้

ตารางที่ 2.3 ประเภทของการบำรุงรักษาซอฟต์แวร์

ประเภท	รายละเอียด
Corrective maintenance	<ul style="list-style-type: none">- ควบคุมการทำงานแต่ละฟังก์ชันของซอฟต์แวร์ ให้ยังคงถูกต้องต้องเป็นประจำทุกวัน- การแก้ไขข้อผิดพลาดทันทีที่ค้นพบ
Adaptive maintenance	<ul style="list-style-type: none">- เมื่อมีการซ่อมบำรุงส่วนหนึ่งและมีผลกระทบต่อบางส่วน จึงต้องใช้วิธีนี้- เพื่อตัดแปลงส่วนที่ได้รับผลกระทบต่อให้สอดคล้องกับส่วนที่ปรับปรุงไปก่อนหน้า
Perfective maintenance	<ul style="list-style-type: none">- ปรับปรุงประสิทธิภาพการทำงานบางอย่าง แม้ว่าไม่มีข้อผิดพลาดเกิดขึ้นก็ตาม เช่น เอกสารให้อ่านง่ายขึ้น ปรับปรุงกรณีทดสอบซอฟต์แวร์ให้ครอบคลุมมากขึ้น ปรับปรุง Code ให้มีคุณภาพมากขึ้น
Preventive maintenance	<ul style="list-style-type: none">- การปรับปรุงที่ไม่ให้การทำงานของซอฟต์แวร์ล้มเหลว เช่น การตรวจหาข้อผิดพลาดร้ายแรง การเขียนโปรแกรมรองรับข้อผิดพลาดลักษณะต่าง ๆ เพิ่มเติม หรือ การเขียน Code เพื่อดักจับข้อผิดพลาดขณะรันโปรแกรมเพิ่มเติม

2.3 กระบวนการพัฒนาซอฟต์แวร์แบบสกรัม (Scrum)

สกรัม (Scrum) เป็นกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) ให้ความสำคัญกับทีมพัฒนาร่วมมือกันทำงาน (Collaboration) ด้วยความเร็ว (Fast-Paced) และคำนึงถึงความพึงพอใจของลูกค้า ซึ่งได้รับความนิยม และถูกนำมาใช้อย่างแพร่หลาย เป็นหนึ่งในวิธีการพัฒนาซอฟต์แวร์แบบเอจิล (Agile) ซึ่งสกรัมเป็นกรอบการทำงานที่ใช้สำหรับการบริหารจัดการโครงการพัฒนาซอฟต์แวร์ อีกทั้งยังช่วยสนับสนุนการพัฒนาโครงการที่มีความซับซ้อนได้ดี โดยแบ่งการพัฒนาออกเป็นวงรอบ (Iteration) มีการเพิ่มฟังก์ชันหรือความต้องการของลูกค้าเข้าไปพัฒนาในแต่ละวงรอบการทำงาน โดยในการทำสกรัมมีกิจกรรมที่เกี่ยวข้องอยู่หลายกิจกรรม ซึ่งแต่ละกิจกรรมจะอยู่ภายใต้กรอบเวลาที่กำหนดไว้ร่วมกันภายในทีม (Time boxing) ความต้องการของลูกค้าจะถูกแตกออกเป็นงานย่อย ๆ เรียกว่า Product backlog และมีการจัดเรียงไว้ตามลำดับความสำคัญ สมาชิกของทีมสกรัมร่วมกันประเมินความยากง่ายของงานออกมาเป็นคะแนน Story point เพื่อนำมาใช้ในการประกอบการตัดสินใจเลือก Product backlog ที่จะพัฒนาให้แล้วเสร็จใน 1 วงรอบการทำงานหรือเรียกว่า Sprint ดังแสดงในภาพที่ 2.3



ภาพที่ 2.3 The Agile Scrum Framework [5]

2.3.1 บทบาทหน้าที่ในทีมสกรัม

Kenneth [6] กล่าวถึงทีมพัฒนาซอฟต์แวร์แบบสกรัม เป็นทีมที่สามารถจัดการตนเองได้ (Self-Organizing Team) กล่าวคือสมาชิกของทีมรู้บทบาทหน้าที่ของตนเอง และทำงานแบบ "ข้ามฟังก์ชันงาน"

(Cross-functional team) โดยที่ไม่ได้มีการกำหนดหน้าที่หรือตำแหน่งงานภายในทีมกันอย่างชัดเจนตายตัว มีเพียงแค่การกำหนดบทบาทหลัก ๆ ไว้ 3 บทบาทเท่านั้น ทำให้สมาชิกภายในทีมต้องทำงานที่หลากหลาย ทั้งที่ตนเองถนัดและไม่ถนัด ส่งผลให้สมาชิกของทีมต้องมีทักษะที่เพียงพอ เพื่อให้สามารถทำงานตามวัตถุประสงค์ที่วางไว้ร่วมกันได้ บทบาทในสกรัม ดังกล่าวสามารถอธิบายได้ ดังนี้

1. เจ้าของผลิตภัณฑ์ (Product Owner) เปรียบเสมือนตัวแทนของผู้มีส่วนได้ส่วนเสียของผลิตภัณฑ์ มีหน้าที่ในการกำหนดคุณลักษณะของซอฟต์แวร์ที่ต้องการ ให้ความชัดเจนเกี่ยวกับ Product backlog และ ลำดับความสำคัญ (Priority) เพื่อให้สอดคล้องกับเป้าหมายของการพัฒนา และเป็นบทบาทเดียวที่มีสิทธิ์ในการปรับปรุง เปลี่ยนแปลงลำดับความสำคัญของ Product backlog ดังกล่าวอีกด้วย อีกทั้งยังต้องทดสอบผลิตภัณฑ์เพื่อให้ทีมพัฒนามั่นใจว่างานที่พัฒนามันมีคุณค่า และทำงานได้จริงตรงตามความต้องการ

2. ทีมพัฒนา (Development team) ลักษณะของทีมพัฒนาในสกรัมไม่ได้มีการกำหนดตำแหน่งหน้าที่กันอย่างชัดเจนเหมือนการพัฒนาซอฟต์แวร์แบบดั้งเดิม เช่น นักเขียนโปรแกรม (Programmer) นักวิเคราะห์ระบบ (System analyst) ผู้ทดสอบระบบ (Tester) นักประกันคุณภาพ (Quality assurance) เป็นต้น ซึ่งทุกคนทำงานแบบข้ามหน้าที่การทำงาน ต้องทำงานที่หลากหลายทั้งที่ตนเองถนัดและไม่ถนัด เพื่อให้สามารถส่งมอบชิ้นงานตรงตามที่ได้จำกัดความของคำว่าแล้วเสร็จในช่วงท้ายของวงรอบได้ โดยขนาดของทีมพัฒนาควรมีสมาชิกตั้งแต่ 5-9 อย่างไรก็ตามในทางปฏิบัติเป็นไปได้ยากที่แต่ละคนจะสามารถทำงานทุกอย่างได้ดี หรือเก่งเท่ากันทุกด้าน ดังนั้นการเลือกสมาชิกของทีมควรเป็นการผสมผสานกันระหว่างผู้ที่มีประสบการณ์มาก (Senior) กับผู้ที่มีประสบการณ์น้อย (Junior) เพื่อกรณีที่สมาชิกของทีมคนหนึ่งคนใดไม่อยู่ สมาชิกคนอื่น ๆ จะยังสามารถทำงานดังกล่าวแทนได้

3. สกรัมมาสเตอร์ (Scrum Master) เป็นผู้ที่ให้คำแนะนำกับทีมพัฒนา สร้างความเข้าใจเกี่ยวกับกรอบการทำงานของสกรัม เพื่อให้สมาชิกของทีมปฏิบัติตามทฤษฎี กฎ ข้อบังคับของสกรัม อีกทั้งยังเป็นคนกลางในการติดต่อประสานงาน แก้ปัญหาที่เกิดขึ้นภายในทีม รวมถึงการอำนวยความสะดวกต่าง ๆ ให้แก่สมาชิกของทีม ยังมีความเข้าใจผิดเกี่ยวกับสกรัมมาสเตอร์อยู่บ้างว่า เปรียบเสมือนผู้จัดการ โครงการ (Project Manager) ในการพัฒนาซอฟต์แวร์แบบดั้งเดิม แต่แท้ที่จริงแล้วทั้งคู่มีความแตกต่างกัน โดยสิ้นเชิง ผู้จัดการโครงการจะทำหน้าที่เหมือนเป็นผู้ควบคุมทีมพัฒนาในการพัฒนาซอฟต์แวร์แบบดั้งเดิม คอยบริหารจัดการโครงการให้เป็นไปตามแผนที่กำหนดไว้ ในขณะที่สกรัมมาสเตอร์เป็นเพียงผู้ให้คำแนะนำ ปกป้อง คอยอำนวยความสะดวกให้กับทีมเท่านั้น หากทำยี่สิบแล้วสมาชิกของทีมตกลงกันว่าจะไม่ทำตามที่สกรัมมาสเตอร์แนะนำก็ยังสามารถทำได้

2.3.2 กระบวนการในสกรัม

Kenneth [6] อธิบายว่าในสกรัมประกอบไปด้วยกิจกรรมต่าง ๆ ซึ่งแต่ละกิจกรรมจะอยู่ภายใต้กรอบเวลาที่กำหนดไว้ร่วมกันภายในทีม กิจกรรมดังกล่าวมีดังต่อไปนี้

1. การรวบรวมความต้องการของแต่ละ Product backlog (Product backlog refinement) เป็นการรวบรวมรายละเอียดเกี่ยวกับความต้องการในแต่ละ Product backlog ดูว่ามีความสมบูรณ์เพียงพอที่จะสามารถนำไปใช้ในแต่ละรอบการพัฒนาได้หรือไม่ สมาชิกของทีมมีการประเมินความยากง่ายของแต่ละ Product backlog ออกมาเป็นคะแนน (Story point) รวมถึงมีการแต่งงานออกเป็นงานย่อย ๆ ซึ่งกิจกรรมนี้จะเป็นการช่วยเจ้าของผลิตภัณฑ์ในการเตรียมความพร้อมให้กับ Product backlog ในรอบการพัฒนาถัดไป

2. การประชุมวางแผนรอบการทำงาน (Sprint planning meeting) ปกติการประชุมวางแผนรอบการทำงานจะถูกกำหนดเวลาประมาณ 8 ชั่วโมง ต่อ 1 รอบ (1 เดือน) หากกรณีที่รอบการทำงานมีระยะเวลาสั้นกว่านั้น การประชุมวางแผนดังกล่าวก็จะลดเวลาลง กิจกรรมนี้มีวัตถุประสงค์เพื่อกำหนดว่า ต้องส่งมอบอะไรบ้างในรอบการทำงานนั้น ๆ และต้องทำอะไรให้งานที่ต้องส่งมอบดังกล่าวสำเร็จลุล่วงได้

3. รอบการทำงาน (Sprint) รอบการทำงานหรือรอบการพัฒนาเป็นหัวใจของสกรัม โดยแต่ละรอบการทำงานใช้เวลาประมาณ 1 เดือน หรือน้อยกว่านั้น เพื่อให้ได้ชิ้นงานที่เสร็จสมบูรณ์และสามารถนำไปประกอบเป็นผลิตภัณฑ์ที่ใช้งานได้ รอบการทำงานใหม่จะเกิดขึ้นทันทีหลังจากที่รอบการทำงานก่อนหน้าสิ้นสุดลง ผู้ที่สามารถทำการยกเลิกรอบการทำงานดังกล่าวได้คือเจ้าของผลิตภัณฑ์เท่านั้น การยกเลิกสามารถทำได้ เช่น กรณีมีนโยบายขององค์กร แผนการตลาด หรือเทคโนโลยีที่เปลี่ยนแปลงไป เป็นต้น แต่ด้วยความที่แต่ละรอบการทำงานมีระยะเวลาค่อนข้างสั้น ทำให้การยกเลิกดังกล่าวมี โอกาสเกิดขึ้นได้น้อยมาก

4. สกรัมประจำวัน (Daily scrum) เป็นการประชุมประจำวัน โดยทีมจะมีการกำหนดเวลาและสถานที่ในการประชุมอย่างชัดเจน ซึ่งจะใช้เวลาและสถานที่เดิมในการประชุมทุกครั้ง เพื่อไม่ให้เกิดความสับสนหรือเสียเวลา ระยะเวลาที่ใช้ในการประชุมประมาณ 5 นาทีของทุกวัน โดยวัตถุประสงค์ของสกรัมประจำวัน คือ เพื่อเป็นการวางแผนการทำงาน ใน 24 ชั่วโมงถัดไป ซึ่งในกิจกรรมนี้สมาชิกของทีมแต่ละคนต้องตอบคำถาม 3 คำถาม คือ 1) ที่ผ่านมา 24 ชั่วโมงที่แล้วทำอะไร 2) ในอีก 24 ชั่วโมงถัดไปจะทำอะไร และ 3) มีปัญหาอะไรหรือไม่ที่ต้องการให้สมาชิกคนอื่น ๆ ช่วยเหลือ

5. การตรวจสอบรอบการทำงาน (Sprint review) เป็นกิจกรรมที่เกิดขึ้นในช่วงท้ายของรอบการทำงาน ใช้เวลาประมาณ 4 ชั่วโมงในการตรวจสอบชิ้นงานที่ได้จากการพัฒนาในรอบนั้น ๆ โดยทีมพัฒนาและผู้มีส่วนได้ส่วนเสียจะร่วมพิจารณาชิ้นงาน เจ้าของผลิตภัณฑ์จะเป็นผู้ระบุว่าเมื่อไรที่เสร็จสมบูรณ์บ้าง ในขณะที่ทีมพัฒนาจะแสดงความคิดเห็น ชี้แจงปัญหาและวิธีการในการแก้ปัญหาดังกล่าว โดยทีมพัฒนาจะเป็นผู้ทำการ

นำเสนอและตอบคำถามเกี่ยวกับชิ้นงานที่ถูกพัฒนา สุดท้ายจะช่วยกันพิจารณารายการ Product backlog ที่เหลืออยู่ และทำการคาดคะเนวันที่จะทำการพัฒนาให้แล้วเสร็จ โดยนำข้อมูลจากรอบการทำงานดังกล่าวไปเป็นตัวอ้างอิงให้กับรอบการทำงานถัดไป

6. การพิจารณาสิ่งที่ผ่านมาในรอบการทำงาน (Sprint retrospective) สมาชิกของทีมพัฒนาจะทำการตรวจสอบตนเองในรอบการทำงานที่ผ่านมา ดูว่ามีจุดบกพร่อง ข้อดีหรือข้อด้อยอะไรบ้าง เพื่อนำไปใช้ปรับปรุงในรอบการทำงานถัดไป การประชุมดังกล่าวใช้ระยะเวลาประมาณ 3 ชั่วโมง ต่อการพัฒนา 1 รอบ ซึ่งเวลาสามารถปรับลดได้ตามความเหมาะสม วัตถุประสงค์ของการประชุมนี้มุ่งเน้นไปที่การตรวจสอบและปรับปรุงทั้งตัวบุคคล กระบวนการทำงาน รวมถึงทีมด้วย

2.3.3 งานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ศึกษา รวบรวม และนำเสนอทักษะที่จำเป็น สำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ เพื่อให้ผู้ปฏิบัติงานสามารถเลือกทักษะดังกล่าวไปใช้ในการพัฒนาตนเอง โดยผู้วิจัยทำการศึกษา และรวบรวมทักษะต่างๆ จากงานวิจัยที่เกี่ยวข้องกับกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ โดยเฉพาะอย่างยิ่งสกรัม อย่างไรก็ตามสกรัมอยู่บนพื้นฐานของการพัฒนาซอฟต์แวร์แบบเอจิล ดังนั้นทักษะที่ถูกใช้ในการพัฒนาซอฟต์แวร์แบบเอจิล จึงถูกนำเสนอในงานวิจัยนี้ด้วย

ในสกรัมไม่เคยมีการกำหนดทักษะที่จำเป็นขึ้นมาอย่างเป็นทางการ Nils [7] ได้ชี้ว่า "สกรัมเน้นการเป็นทีมที่สามารถจัดการตนเองได้ แต่ก็ไม่ได้กำหนดแนวทางหรือทักษะที่จำเป็นไว้อย่างชัดเจน" แต่จากการศึกษาพบว่า มีบางงานวิจัยที่ได้กล่าวถึงทักษะที่ใช้ในสกรัมให้เห็นอยู่บ้าง ซึ่งส่วนใหญ่จะ กล่าวถึงทักษะการสื่อสาร (Communication) สกรัมเน้นการทำงานในสถานที่เดียวกัน เพื่อให้ได้การสื่อสารที่รวดเร็ว หากเกิดข้อสงสัยหรือปัญหาจะได้แก้ไขได้ทันที่ การสื่อสารที่ไม่มีประสิทธิภาพและประสิทธิผล อาจทำให้เกิดการทำงานที่ซ้ำซ้อน และอาจเกิดความผิดพลาดขึ้นมาได้

Anita [8] จัดทำแบบสำรวจทักษะที่จำเป็นในทีมพัฒนาซอฟต์แวร์สกรัมระดับสากล (Scrum Global Software Development Teams) พบว่าทักษะที่มีความสำคัญมากที่สุด 3 ลำดับแรกคือ ทักษะการเขียนโปรแกรม (Programming skill) ทักษะด้านมนุษยสัมพันธ์ (Interpersonal skill) และทักษะการสื่อสาร (Communication skill)

Mazni ศึกษาและนำเสนอทักษะที่จำเป็นในทีมพัฒนาซอฟต์แวร์สกรัม จากการสัมภาษณ์ความคิดเห็นของผู้เชี่ยวชาญ โดยแบ่งเป็น 2 ประเภทได้แก่ ทักษะทางเทคนิค (Hard skill) ประกอบไปด้วย ทักษะการเขียนโปรแกรม (Programming skill) ทักษะด้านฐานข้อมูล (Database) ทักษะด้านความรู้ด้านสกรัม (Scrum knowledge) และทักษะทางการปฏิบัติงาน (Soft skill) ประกอบด้วย ทักษะด้านการคิดวิเคราะห์

(Analytical skill) ทักษะด้านการสื่อสาร (Communication skill) ทักษะการอำนวยความสะดวก (Facilitation skill) ทักษะด้านมนุษยสัมพันธ์ (Interpersonal skill) ทักษะการเป็นผู้นำ (Leadership skill) ทักษะบริหารจัดการ (Management skill) ทักษะในการทำงานร่วมกับคนอื่น (People skills) ทักษะการวางแผน (Planning skill) ทักษะการทำงานเป็นทีม (Teamwork skill) และทักษะการคิด (Thinking skill)

Prabhakaran [9] ได้เขียนบทความในนิตยสาร "Agile Record, the Magazine for Agile Developers and Agile Testers" เกี่ยวกับทักษะของสกรัมทีม โดยมีการแบ่งทักษะที่เกี่ยวข้อง ออกเป็น 2 ประเภท คือ ทักษะทางด้านเทคนิค (Technical Skill) และทักษะที่เห็นได้ทางด้านพฤติกรรม (Behavioral skill) ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 ทักษะทางด้านเทคนิค และทักษะที่เห็นได้ทางด้านพฤติกรรม

Role	Technical skills	Behavioral skills
Developer	<ul style="list-style-type: none"> - CRUD operations, interfacing with different layers of the development frame work. - Unit testing - Code coverage concepts and tools - Code review concepts and tools - Continuous integration tools - Refactoring concepts - Code-smell concepts - Scrum process 	<ul style="list-style-type: none"> - Communication - Collaboration - Time Management/ Planning - Thinking - Conflict Management - Dealing with Change/ Flexibility - Decision making - Teamwork/ Teambuilding - Handling stress - Problem Solving - Leadership - Diplomacy
QA	<ul style="list-style-type: none"> - Definition of done -> acceptance criteria - Test management - Automation / scripting - Environment setup - Database concepts 	<ul style="list-style-type: none"> - Communication - Collaboration - Time Management/ Planning - Thinking - Conflict Management - Dealing with Change/ Flexibility - Decision making - Teamwork/ Teambuilding

Role	Technical skills	Behavioral skills
		<ul style="list-style-type: none"> - Handling stress - Problem Solving - Leadership - Diplomacy
Scrum Master	<ul style="list-style-type: none"> - Scrum process - Templates and usage - Project Management tools - Continuous integration tools - Development environment setup 	<ul style="list-style-type: none"> - Developer skills + facilitation

Alistair และ Jim [10] ได้กล่าวเกี่ยวกับการพัฒนาซอฟต์แวร์โดยใช้เทคนิคเอจิล์ไว้ในบทความวิชาการชื่อ “Agile Software Development: The People Factor” ว่า จะต้องให้ความสำคัญกับปัจจัยรายบุคคล (People Factors) ของสมาชิกภายในทีม ได้แก่ ความสามัคคี (Amicability) ความสามารถ (Talent) ทักษะ (Skill) และการสื่อสาร (Communication) ซึ่งสนับสนุนให้เกิดการเป็นผู้นำในการทำงานร่วมกัน (Leadership-collaboration) มากกว่าการควบคุมด้วยคำสั่ง (Command-control) โดยมีทัศนคติส่วนบุคคล (Attitude) เป็นเรื่องที่สำคัญต่อการพัฒนาทักษะความสามารถ และการทำงานเป็นทีม

Nils [7] นำเสนอผลวิจัยเกี่ยวกับทีมพัฒนาที่ใช้เอจิล์พบว่า การให้อิสระในการทำงาน (Autonomy) และการมีทักษะการแบ่งส่วนงานที่เหมาะสม (Corresponding division of work) มีผลต่อความสำเร็จในการบริหารจัดการตนเอง (Self-organization)

Tan และ Teo [11] ได้นำรูปแบบการพัฒนาซอฟต์แวร์แบบเอจิล์ไปใช้ในการเรียนการสอน โดยนำเสนอผลลัพธ์ว่า นักศึกษามีทัศนคติที่ดีขึ้นเกี่ยวกับเรื่องของคุณภาพของซอฟต์แวร์ และยังนำเสนอว่ายังส่งผลถึงทักษะในการจัดการกับการเปลี่ยนแปลง (Dealing with Changes) และทักษะการจัดการความขัดแย้ง (Conflict Management)

Gerardo [12] สสำรวจทักษะด้าน Soft Skill ในทีมพัฒนาซอฟต์แวร์แบบเอจิล์ในประเทศตุรกีทั้งในมุมมองของหัวหน้าทีม (Team Leaders) และสมาชิกภายในทีม (Team Members) โดยจากทั้ง 2 มุมมองเห็นตรงกันว่า ทักษะการคิดวิเคราะห์แก้ไขปัญหา (Analytic, Problem-solving) ทักษะความรับผิดชอบ (Commitment, responsibility) ทักษะการเรียนรู้ (Eagerness to learn) และทักษะการทำงานเป็นทีม (Teamwork) สอดคล้องกับ David [13] ที่นำเสนอทักษะด้านทางการปฏิบัติงาน (Soft Skill) ในทีมเอจิล์

ระดับสากล (Agile Global Software Development) 3 ลำดับแรก ได้แก่ ทักษะด้านการทำงานเป็นทีม (Team Work) ทักษะการสื่อสาร (Communication) และทักษะการเป็นผู้นำ (Leadership)

บทที่ 3 วิธีดำเนินการวิจัย

งานวิจัยนี้ใช้เทคนิคการทบทวนอย่างเป็นระบบ (Systematic review) ซึ่งเป็นวิธีการที่ใช้เก็บรวบรวมและเรียบเรียงความรู้ในการสืบค้นข้อมูล และสังเคราะห์เป็นข้อมูลใหม่ โดยภายในงานวิจัยนี้แบ่งขั้นตอนออกเป็น 4 ขั้นตอนหลักๆ ได้แก่ 1) การกำหนดปัญหาและรวบรวมข้อมูล 2) การสังเคราะห์ผล 3) การแปลผล และ 4) การวิพากษ์ผลลัพธ์ ดังแสดงในภาพที่ 3.1 ขั้นตอนการวิจัย



ภาพที่ 3.1 ขั้นตอนการวิจัย

3.1 การกำหนดปัญหาและรวบรวมข้อมูล

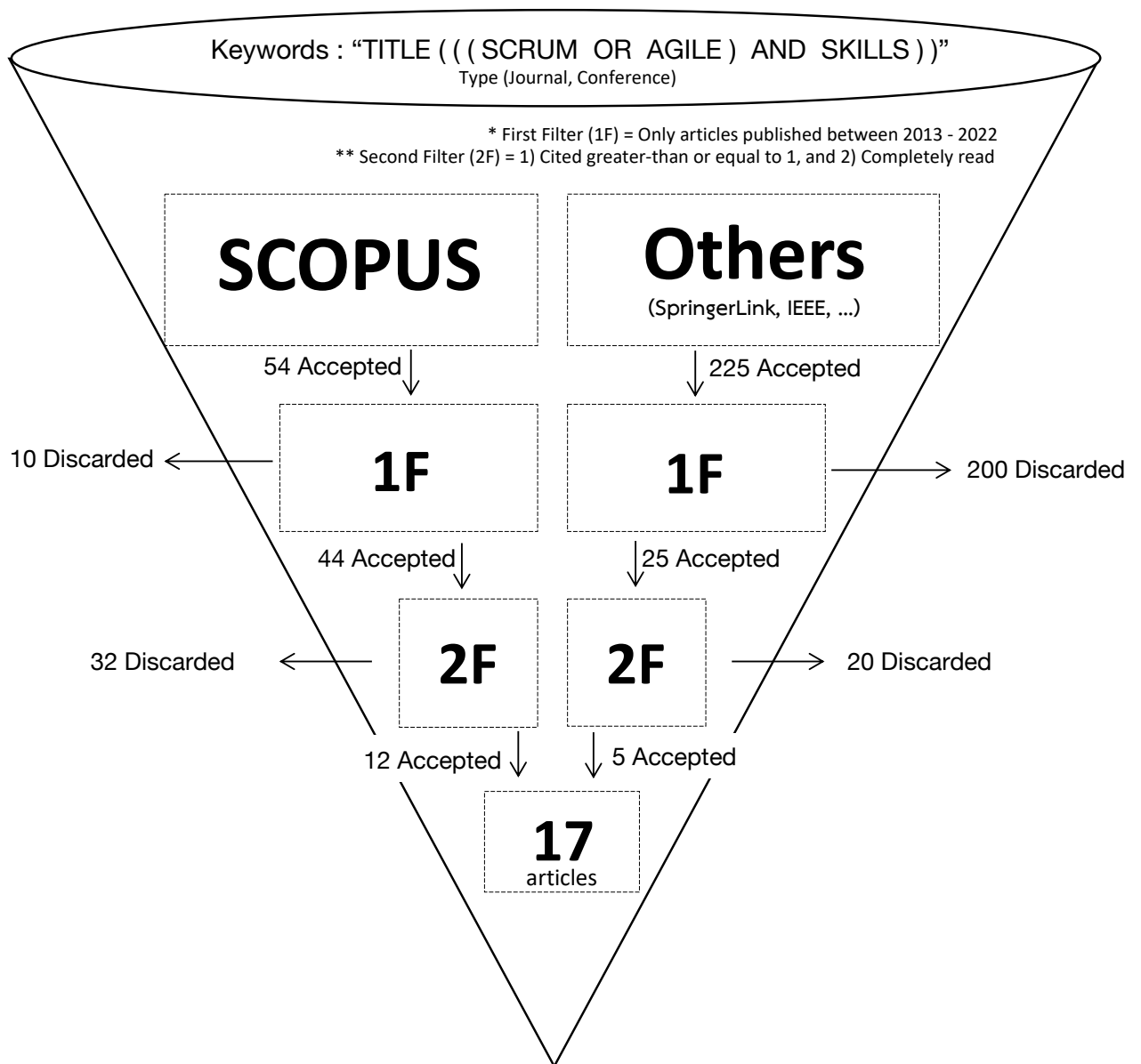
งานวิจัยนี้ศึกษา รวบรวม และนำเสนอทักษะที่จำเป็น สำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบเอจิล (Agile) เพื่อให้ผู้ปฏิบัติงานสามารถเลือกทักษะดังกล่าวไปใช้ในการพัฒนาตนเอง โดยผู้วิจัยทำการศึกษา และรวบรวมทักษะต่างๆ จากงานวิจัยที่เกี่ยวข้อง โดยกำหนดปัญหาของงานวิจัยจำนวน 3 คำถาม ดังต่อไปนี้

1. ทักษะด้านเทคนิค (Technical skills) ไต ที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบเอจิล (Agile)
2. ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) ไต ที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบเอจิล (Agile)
3. ทักษะการที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes) ไต ที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบเอจิล (Agile)

นอกจากนี้ผู้วิจัยได้รวบรวมข้อมูลที่เกี่ยวข้อง เพื่อตอบคำถามของงานวิจัยจากบทความวิจัยในฐานข้อมูลอิเล็กทรอนิกส์จำนวน 2 แหล่งหลักๆ ได้แก่ 1) ฐานข้อมูล SCOPUS และ 2) ฐานข้อมูลอื่นๆ เช่น SpringerLink, IEEE และอื่น ๆ โดยการกำหนดคำสำคัญที่ใช้ในการสืบค้นคือ ค้นหาชื่อบทความวิจัยที่มีคำว่า “scrum” หรือ “agile” และ “skills” ผสมอยู่ ดังคำค้นต่อไปนี้

TITLE (((scrum OR agile) AND skills))

ทั้งนี้ผู้วิจัยได้คัดกรอง (Filter) บทความวิจัยที่ไม่เกี่ยวข้อง ไม่ตรงกับคำถาม หรือไม่ตรงกับประเด็นออกไป เหลือไว้เฉพาะบทความที่มีความเกี่ยวข้องโดยตรง จำนวน 2 รอบ ประกอบด้วย (F1) – ต้องเป็นบทความวิจัยที่ถูกตีพิมพ์/เผยแพร่ในช่วงปี ค.ศ. 2013 – 2022 (10 ปีย้อนหลัง) เพื่อให้ได้ข้อมูลที่มีความทันสมัย สอดคล้องกับยุคสมัยปัจจุบัน (F2) – บทความวิจัยต้องเคยถูกอ้างอิงอย่างน้อย 1 ครั้ง และผู้วิจัยจะดำเนินการอ่านบทความทั้งฉบับเพื่อพิจารณาความเกี่ยวข้อง โดยรายละเอียดการคัดกรองแสดงในภาพที่ 3.2 และผลลัพธ์ของการรวบรวมข้อมูลแสดงดังตารางที่ 3.1



ภาพที่ 3.2 การค้นหาข้อมูลที่เกี่ยวข้องกับงานวิจัย

ตารางที่ 3.1 ทักษะและทัศนคติที่ได้จากการรวบรวมข้อมูล

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
1.	Distributed communication	/																
2.	Distributed teamwork	/																
3.	Building and maintaining trust	/																
4.	Tool usage for distributed collaboration	/																
5.	Inter-cultural collaboration	/							/									
6.	Teamwork		/	/						/		/			/			/
7.	Team communication		/															
8.	Overall agile development process		/															
9.	Problems solving		/									/						
10.	Self-learning		/															
11.	System thinking			/														
12.	Coding and programming languages			/	/													
13.	Architecture			/														
14.	Testing			/	/													
15.	Quality			/	/													
16.	Release management			/														
17.	Management			/						/					/			/

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
18.	Negotiation			/				/				/				/		/
19.	Communication			/	/		/			/		/	/	/	/	/		/
20.	Courtesy			/														/
21.	Flexibility			/														
22.	Integrity			/														/
23.	Interpersonal			/										/	/			
24.	Positive attitude			/														/
25.	Professionalism			/														/
26.	Responsibility			/										/				/
27.	Work ethic			/														
28.	Continuous integration				/				/	/	/					/		
29.	Clean Code				/													
30.	Collaboration				/					/		/	/					
31.	Transparency and openness				/		/											
32.	Craftsmanship				/													
33.	OO-concepts				/											/		
34.	Secure design			/		/												
35.	Secure coding					/												
36.	Static code analysis					/												

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
37.	Security testing					/												
38.	Threat modeling & risk management					/												
39.	Writing						/											/
40.	Oral presentation						/								/	/	/	
41.	Punctuality						/											
42.	Leadership						/		/		/	/	/	/	/			/
43.	Scrum knowledge						/		/									
44.	Entrepreneurship							/		/								
45.	Agile software development							/										
46.	Breaking down a project task into smaller tasks							/							/			
47.	Estimating effort needed to complete software development work							/										
48.	Interacting with customers / product owners							/										
49.	Motivation							/				/						
50.	Domain expertise							/					/					
51.	Backlog of the product								/									
52.	Behavior Driven Development (BDD)								/									

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
53.	Test Driven Development (TDD)								/		/					/		
54.	Short and frequently deliveries								/									
55.	Burndown chart								/		/							
56.	Kanban								/									
57.	Release planning								/	/	/	/	/		/			/
58.	Pair programming								/		/					/		
59.	Collective code ownership								/							/		
60.	Code refactoring								/	/	/					/		
61.	Scrum meetings								/	/	/							
62.	Unit test								/							/		
63.	Programming language									/					/			
64.	Database									/					/			
65.	Spoken and written									/								
66.	Analytical									/					/			/
67.	Facilitation									/					/			/
68.	Thinking									/					/		/	/
69.	Create, Retrieve, Update, Delete (CRUD)									/								
70.	Code coverage concepts and tools									/						/		
71.	Code review concepts and tools									/						/		

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
72.	Code-smell concepts									/								
73.	Time management									/						/	/	
74.	Conflict management									/						/		/
75.	Dealing with change / flexibility									/						/		/
76.	Decision making									/		/				/		/
77.	Teamwork / teambuilding									/						/		
78.	Handling stress									/						/		
79.	Sprint planning meeting										/							
80.	Weekly interactions and meetings										/							
81.	Coding standard										/					/		
82.	Customer orientation											/						
83.	Requirements management											/						
84.	Business-minded											/						
85.	Organizing											/						
86.	Knowledge of tools for work planning												/					
87.	Specify the software requirements												/				/	
88.	Knowledge of software development environments												/					
89.	Apply techniques for estimating effort												/			/		

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
90.	Adaptability												/					
91.	Self-management												/					
92.	Creativity												/					
93.	Second language (English)												/	/				
94.	Learning from error												/					
95.	Team identity												/					
96.	Understand other perspectives needs and values of people												/					
97.	Manage diversity in the team												/					
98.	Organizational												/					
99.	Scrum Process															/		
100.	Belief in self-directing team															/		
101.	Focus on quality															/		
102.	Focus on working software															/		
103.	Focus on time boxing															/		
104.	Respect and trust															/		
105.	Advance prioritization															/		
106.	Belief in simplicity															/		
107.	Courage															/		

#	ชื่อทักษะและทัศนคติ (Skills/Attitudes name)	Reference Index																
		[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
108.	Diplomacy															/		/
109.	Debugging															/		
110.	Active learning															/		
111.	Code smelling															/		
112.	Computational thinking															/		
113.	Creative thinking															/		
114.	Environment preparation															/		
115.	Design pattern															/		
116.	Apply knowledge and method																/	
117.	Use ICT																/	
118.	Searching																/	
119.	Be self-directed learner																/	
120.	Cooperation																	/
121.	Innovation																	/
122.	Listening																	/
123.	Share knowledge																	/

3.2 การสังเคราะห์ผล

เนื่องจากบางทักษะและทัศนคติที่ได้จากการรวบรวมข้อมูล มีความคล้ายคลึงกันอย่างชัดเจน ผู้วิจัยจึงยุบรวมบางทักษะเข้าด้วยกัน และแบ่งประเภทออกเป็น 3 ประเภท เพื่อตอบคำถามงานวิจัยได้แก่ ทักษะด้านเทคนิค (Technical skills) ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) และทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)

3.2.1 การยุบรวมทักษะและทัศนคติที่ได้ในแต่ละแหล่งข้อมูล

ผู้วิจัยสังเคราะห์ทักษะ/ทัศนคติที่ได้จากการรวบรวมข้อมูล จากทั้งหมด 123 ทักษะ ดังแสดงในตารางที่ 3.1 ข้างต้น เหลือเพียง 48 ทักษะ นอกจากนี้ผู้วิจัยได้เสนอชื่อทักษะใหม่ เพื่อให้มีความชัดเจนมากขึ้น ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 การยุบรวมทักษะและทัศนคติที่ได้ในแต่ละแหล่งข้อมูล

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
1.	Communication	<ul style="list-style-type: none"> ● Distributed communication ● Team communication ● Communication ● Writing ● Oral presentation ● Interacting with customers / product owners ● Spoken and written ● Second language (English)
2.	Teamwork / teambuilding	<ul style="list-style-type: none"> ● Distributed teamwork ● Tool usage for distributed collaboration ● Inter-cultural collaboration ● Teamwork ● Collaboration ● Teamwork / teambuilding ● Team identity ● Manage diversity in the team

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
		<ul style="list-style-type: none"> ● Belief in self-directing team ● Cooperation
3.	Respect and trust	<ul style="list-style-type: none"> ● Building and maintaining trust ● Respect and trust
4.	Overall development process	<ul style="list-style-type: none"> ● Overall agile development process ● Scrum knowledge ● Agile software development ● Backlog of the product ● Burndown chart ● Kanban ● Scrum meeting ● Sprint planning meeting ● Weekly interactions and meetings ● Scrum Process ● Breaking down a project task into smaller tasks
5.	Problems solving/ System thinking	<ul style="list-style-type: none"> ● Problems solving ● System thinking ● Analytical ● Thinking ● Decision making ● Creativity ● Computational thinking ● Creative thinking ● Innovation
6.	Handling stress	<ul style="list-style-type: none"> ● Handling stress
7.	Self-learning	<ul style="list-style-type: none"> ● Self-learning

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
		<ul style="list-style-type: none"> ● Learning from error ● Active learning ● Apply knowledge and method ● Searching ● Be self-directed learner
8.	Programming	<ul style="list-style-type: none"> ● Coding and programming languages ● Secure coding ● Static code analysis ● Programming language ● Create, Retrieve, Update, Delete (CRUD)
9.	Environment preparation	<ul style="list-style-type: none"> ● Architecture ● Secure design ● Knowledge of software development environments ● Environment preparation
10.	Testing	<ul style="list-style-type: none"> ● Testing ● Security testing ● Behavior Driven Development (BDD) ● Test Driven Development (TDD) ● Unit test
11.	Release management	<ul style="list-style-type: none"> ● Release management ● Short and frequently deliveries ● Release planning
12.	Focus on quality	<ul style="list-style-type: none"> ● Quality ● Focus on quality
13.	Management	<ul style="list-style-type: none"> ● Management ● Organizing

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
		<ul style="list-style-type: none"> ● Self-management
14.	Negotiation	<ul style="list-style-type: none"> ● Negotiation ● Conflict management
15.	Courtesy	<ul style="list-style-type: none"> ● Courtesy
16.	Dealing with change	<ul style="list-style-type: none"> ● Flexibility ● Dealing with change / flexibility
17.	Responsibility	<ul style="list-style-type: none"> ● Integrity ● Responsibility ● Work ethic ● Transparency and openness ● Collective code ownership ● Organizational
18.	Diplomacy	<ul style="list-style-type: none"> ● Interpersonal ● Diplomacy ● Understand other perspectives needs and values of people ● Listening
19.	Positive attitude	<ul style="list-style-type: none"> ● Positive attitude
20.	Professionalism	<ul style="list-style-type: none"> ● Professionalism ● Craftsmanship ● Domain expertise
21.	Continuous integration	<ul style="list-style-type: none"> ● Continuous integration
22.	Clean Code	<ul style="list-style-type: none"> ● Clean Code
23.	OO-concepts	<ul style="list-style-type: none"> ● OO-concepts
24.	Risk management	<ul style="list-style-type: none"> ● Threat modeling & risk management
25.	Focus on time boxing	<ul style="list-style-type: none"> ● Punctuality ● Time management

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
		<ul style="list-style-type: none"> ● Focus on time boxing
26.	Leadership	<ul style="list-style-type: none"> ● Leadership
27.	Entrepreneurship	<ul style="list-style-type: none"> ● Entrepreneurship ● Business-minded
28.	Effort Estimation	<ul style="list-style-type: none"> ● Estimating effort needed to complete software development work ● Apply techniques for estimating effort
29.	Motivation	<ul style="list-style-type: none"> ● Motivation
30.	Pair programming	<ul style="list-style-type: none"> ● Pair programming
31.	Code refactoring	<ul style="list-style-type: none"> ● Code refactoring
32.	Database	<ul style="list-style-type: none"> ● Database
33.	Facilitation	<ul style="list-style-type: none"> ● Facilitation
34.	Code coverage	<ul style="list-style-type: none"> ● Code coverage concepts and tools
35.	Code review	<ul style="list-style-type: none"> ● Code review concepts and tools
36.	Code smelling	<ul style="list-style-type: none"> ● Code-smell concepts ● Code smelling
37.	Coding standard	<ul style="list-style-type: none"> ● Coding standard
38.	Focus on working software	<ul style="list-style-type: none"> ● Customer orientation ● Focus on working software
39.	Requirement management	<ul style="list-style-type: none"> ● Requirement management ● Specify the software requirements
40.	Planning	<ul style="list-style-type: none"> ● Knowledge of tools for work planning
41.	Adaptability	<ul style="list-style-type: none"> ● Adaptability
42.	Advance prioritization	<ul style="list-style-type: none"> ● Advance prioritization
43.	Belief in simplicity	<ul style="list-style-type: none"> ● Belief in simplicity
44.	Courage	<ul style="list-style-type: none"> ● Courage
45.	Debugging	<ul style="list-style-type: none"> ● Debugging

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
46.	Design pattern	● Design pattern
47.	Use ICT	● Use ICT
48.	Share knowledge	● Share knowledge

3.2.2 ประเภทของทักษะและทัศนคติที่ผู้วิจัยเสนอเพื่อตอบคำถามงานวิจัย

ผู้วิจัยแบ่งประเภททักษะและทัศนคติออกเป็น 3 ประเภท เพื่อตอบคำถามงานวิจัย รายละเอียดแสดงดังตารางที่ 3.3

1. ทักษะด้านเทคนิค (Technical skills) ประกอบด้วยทักษะย่อย จำนวน 23 ทักษะ
2. ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) ประกอบด้วยทักษะย่อย จำนวน 18 ทักษะ
3. ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes) ประกอบด้วยทัศนคติย่อย จำนวน 7 ทัศนคติ

ตารางที่ 3.3 ประเภทของทักษะและทัศนคติที่ผู้วิจัยเสนอเพื่อตอบคำถามงานวิจัย

#	ประเภททักษะ/ทัศนคติ (Skill type)	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)
1.	ทักษะด้านเทคนิค (Technical Skills)	<ol style="list-style-type: none"> 1. Overall development process 2. Programming 3. Environment preparation 4. Testing 5. Release management 6. Continuous integration 7. Clean Code 8. OO-concepts 9. Risk management 10. Effort Estimation 11. Pair programming 12. Code refactoring 13. Database 14. Code coverage 15. Code review

#	ประเภททักษะ/ทัศนคติ (Skill type)	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)
		16. Code smelling 17. Coding standard 18. Requirement management 19. Planning 20. Advance prioritization 21. Debugging 22. Design pattern 23. Use ICT
2.	ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical Skills)	1. Communication 2. Teamwork / teambuilding 3. Problems solving/ System thinking 4. Handling stress 5. Self-learning 6. Management 7. Negotiation 8. Courtesy 9. Dealing with change 10. Responsibility 11. Diplomacy 12. Professionalism 13. Leadership 14. Entrepreneurship 15. Motivation 16. Facilitation 17. Adaptability 18. Share knowledge
3.	ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)	1. Respect and trust 2. Focus on quality 3. Positive attitude 4. Focus on time boxing

#	ประเภททักษะ/ทัศนคติ (Skill type)	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)
		5. Focus on working software 6. Belief in simplicity 7. Courage

3.3 การวิพากษ์โดยผู้เชี่ยวชาญ

ภายหลังจากการสังเคราะห์ผลของงานวิจัยเรียบร้อยแล้ว (การยุบรวมทักษะ และแบ่งประเภททักษะ) ทีมผู้วิจัยได้นำเสนอผลการสังเคราะห์แก่ผู้เชี่ยวชาญด้านสกรัมที่ได้รับใบรับรองจากสถาบันที่ให้การรับรองจำนวน 1 คน และผู้เชี่ยวชาญด้านการใช้งานสกรัมซึ่งเป็น จำนวน 1 คน เพื่อตรวจสอบความถูกต้อง ทั้งในแง่ความเหมาะสมและถูกต้อง มีการนำคำแนะนำดังกล่าวมาปรับปรุงแก้ไขงานวิจัย เพื่อความถูกต้องและเหมาะสมมากขึ้น

ตารางที่ 3.4 ประเภทผู้เชี่ยวชาญวิพากษ์ผลการสังเคราะห์ของงานวิจัย

#	ประเภทผู้เชี่ยวชาญ	รายละเอียด
1.	ผู้มีใบรับรองจากสถาบันด้านสกรัมในระดับนานาชาติ	ได้รับ Certified ด้านสกรัมอย่างน้อย 1 ประเภท จาก <ul style="list-style-type: none"> ● Scrum Alliance ● Scrum.org ● International Scrum Institute ● หรือจากสถาบันที่ออกใบรับรองในระดับนานาชาติ
2.	ผู้นำสกรัมไปใช้งาน	<ul style="list-style-type: none"> ● ผ่านการอบรม หรือเข้าร่วม workshop ด้านสกรัม ● เป็นส่วนหนึ่งของสมาชิกในทีมสกรัมแบบเต็มเวลา ● มีประสบการณ์ในการทำงานโดยใช้สกรัมมากกว่า 5 ปี

บทที่ 4 สรุปผลการดำเนินงาน

4.1 ผลการวิพากษ์โดยผู้เชี่ยวชาญ

4.1.1 ข้อมูลผู้เชี่ยวชาญประเภทผู้มีใบรับรองจากสถาบันด้านสกรัมในระดับนานาชาติ

ได้รับใบรับรอง Scrum Master Accredited Certification จาก International Scrum Institute ตั้งแต่ปี 2019 – ปัจจุบัน และใบรับรอง PROFESSIONAL SCRUM MASTER จาก Scrum.org ตั้งแต่ปี 2019 - ปัจจุบัน

4.1.2 ข้อมูลผู้เชี่ยวชาญประเภทผู้นำสกรัมไปใช้งาน

ผ่านการอบรมด้านสกรัม และเข้าร่วมกิจกรรมด้านสกรัมที่จัดขึ้นในประเทศไทย ตั้งแต่ปี 2558 เคยเป็นส่วนหนึ่งของสมาชิกในทีมสกรัมแบบเต็มเวลา ทั้งในบทบาท Scrum Master และ Scrum Member และมีประสบการณ์ในการทำงานโดยใช้สกรัมมากกว่า 6 ปี ซึ่งปัจจุบันดำรงตำแหน่งผู้บริหารระดับสูงสุดของบริษัท (CEO) ของบริษัท Start up

4.1.3 ข้อมูลที่ได้รับจากการวิพากษ์โดยผู้เชี่ยวชาญ

ตารางที่ 4.1 ข้อมูลจากผู้เชี่ยวชาญ

#	ประเภทผู้เชี่ยวชาญ	ผลการวิพากษ์
1.	ผู้มีใบรับรองจากสถาบันด้านสกรัมในระดับนานาชาติ	<ul style="list-style-type: none">การรวบรวมทักษะทั้ง Technical skills, Non-technical skills และ Mind set ครบถ้วน และเพียงพอต่อการนำมาใช้ในงานวิจัยการยุบรวมกลุ่มของทักษะ บางทักษะอาจถูกจำแนกอยู่ในกลุ่มทักษะได้หลายกลุ่มเปลี่ยนชื่อทักษะที่ผู้วิจัยนำเสนอจาก “Courtesy” เป็น Thai courtesyการจัดกลุ่มทักษะที่นำเสนอ มีความเหมาะสมแล้ว แต่ควรจัดเรียงลำดับการนำเสนอ แต่ละทักษะใน Technical skills ตามกระบวนการในการพัฒนาซอฟต์แวร์ (Systems Development Life Cycle : SDLC)หากผู้วิจัยนำเสนอแนวทางในการวัดความสามารถในแต่ละทักษะด้วย จะก่อให้เกิดประโยชน์ต่อสถานประกอบการในการคัดเลือกบุคคลเข้าทำงานหากสัมภาษณ์หรือเก็บข้อมูลเพิ่มเติมจากกลุ่มตัวอย่างที่เป็นสถานประกอบการในประเทศไทยจะส่งเสริมให้ผลของการวิจัยมีประสิทธิภาพมากขึ้นสามารถนำความรู้จากการวิจัยไปใช้ในการพัฒนาทักษะนิสิตในระดับมหาวิทยาลัยได้
2.	ผู้นำสกรัมไปใช้งาน	<ul style="list-style-type: none">เห็นด้วยกับการจำแนกทักษะในกลุ่มต่างๆ ที่ผู้วิจัยนำเสนอบางทักษะอาจถูกจัดอยู่ในได้หลายกลุ่ม เช่น Respect and trust อาจเป็นส่วนหนึ่งของ Teamwork / teambuilding ได้

4.2 สรุปทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์

ผู้วิจัยนำผลการวิพากษ์จากผู้เชี่ยวชาญมาปรับปรุงผลงานวิจัย และสรุปเป็นภาพรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์

<ol style="list-style-type: none"> 1. ทักษะกระบวนการพัฒนาโดยรวม (Overall development process) 2. ทักษะการบริหารจัดการความต้องการ (Requirement management) 3. ทักษะการประมาณการแรงงาน (Effort Estimation) 4. ทักษะการวางแผน (Planning) 5. ทักษะการลำดับความสำคัญของงาน (Advance prioritization) 6. ทักษะการเตรียมสภาพแวดล้อมที่เหมาะสมกับการพัฒนาซอฟต์แวร์ (Environment preparation) 7. ทักษะการใช้ดีไซน์แพทเทิร์น (Design pattern) 8. ทักษะการหากรณีทดสอบให้ครอบคลุม (Code coverage) 9. ทักษะในการทวนสอบโค้ด (Code review) 10. ทักษะการระบุโค้ดที่นำไปสู่ปัญหา (Code smelling) 11. ทักษะการเขียนโค้ดตามมาตรฐาน (Coding standard) 12. ทักษะการออกแบบและการเขียนโปรแกรมเชิงวัตถุ (OO-concepts) 13. ทักษะการเขียนโปรแกรม (Programming) 14. ทักษะการเขียนโปรแกรมเป็นคู่ (Pair programming) 15. ทักษะฐานข้อมูล (Database) 16. ทักษะในการโค้ดให้ง่ายต่อการเขียน อ่าน และดูแลรักษา (Clean Code) 17. ทักษะการเปลี่ยนแปลงโครงสร้างภายในโค้ด เพื่อให้ง่ายต่อการทำความเข้าใจ (Code refactoring) 18. ทักษะการทดสอบ (Testing) 19. ทักษะการตรวจสอบความผิดพลาด (Debugging) 20. ทักษะการรวบรวมโค้ดที่ได้รับการพัฒนาจากสมาชิกแต่ละคนในทีมให้เป็นชิ้นเดียว (Continuous integration) 21. ทักษะการบริหารจัดการการนำผลงานเข้าสู่การให้บริการ (Release management) 22. ทักษะการบริหารความเสี่ยง (Risk management) 23. ทักษะการใช้เทคโนโลยีสารสนเทศ (Use ICT) 	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ทักษะด้านเทคนิค (Technical skills) </div>	<ol style="list-style-type: none"> 1. ทักษะการสื่อสาร (Communication) 2. ทักษะการทำงานเป็นทีม (Teamwork / teambuilding) 3. ทักษะการแก้ไขปัญหา/การคิดอย่างเป็นระบบ (Problems solving/ System thinking) 4. ทักษะการจัดการความเครียด (Handling stress) 5. ทักษะการเรียนรู้ด้วยตนเอง (Self-learning) 6. ทักษะการบริหารจัดการ (Management) 7. ทักษะการเจรจาต่อรอง (Negotiation) 8. ทักษะมารยาทไทย (Thai courtesy) 9. ทักษะการจัดการการเปลี่ยนแปลง (Dealing with change) 10. ทักษะความรับผิดชอบ (Responsibility) 11. ทักษะการสร้างสัมพันธภาพและเข้าใจกัน (Diplomacy) 12. ทักษะความเป็นมืออาชีพ (Professionalism) 13. ทักษะการเป็นผู้นำ (Leadership) 14. ทักษะการเป็นผู้ประกอบการ (Entrepreneurship) 15. ทักษะการสร้างแรงจูงใจ (Motivation) 16. ทักษะการอำนวยความสะดวก (Facilitation) 17. ทักษะการปรับตัว (Adaptability) 18. ทักษะการแบ่งปันข้อมูล (Share knowledge) 	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) </div>
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes) </div>		

ภาพที่ 4.1 สรุปทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์

4.3 ทักษะด้านเทคนิค (Technical skills)

เพื่อตอบคำถามงานวิจัยข้อที่ 1 “ทักษะด้านเทคนิค (Technical skills) ไต ที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบเอจิล (Agile)” ผู้วิจัยสังเคราะห์ข้อมูลที่ได้จากการรวบรวม และนำเสนอทักษะ จำนวน 23 ทักษะ ดังแสดงในตารางที่ 4.3

ตารางที่ 4.2 ทักษะด้านเทคนิค (Technical skills)

#	ทักษะด้านเทคนิค (Technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
1.	ทักษะกระบวนการพัฒนาโดยรวม (Overall development process)	ทักษะในการทำกิจกรรมต่างๆ ในกระบวนการของการพัฒนาซอฟต์แวร์นั้นๆ โดยมีความรู้ความเข้าใจไม่ว่าจะเป็นบทบาทใด (Role) กิจกรรมใด (Event) งานหรือผลงานจากกิจกรรมที่กระทำ (Artifact) หลักปฏิบัติ (Practice) ตั้งแต่เริ่มต้นจนถึงสิ้นสุดการพัฒนา ตัวอย่างเช่น ในการพัฒนาซอฟต์แวร์แบบสกรัม สมาชิกทีมต้องร่วมกันทำกิจกรรมหลักๆ เช่น การประชุมวางแผนรอบการทำงาน (Sprint planning meeting) รอบการทำงาน (Sprint) สกรัมประจำวัน (Daily scrum) การตรวจสอบรอบการทำงาน (Sprint review) และเข้าใจบทบาทหน้าที่ของเจ้าของผลิตภัณฑ์ (Product Owner) ทีมพัฒนา (Development team) และสกรัมมาสเตอร์ (Scrum Master)
2.	ทักษะการบริหารจัดการความต้องการ (Requirement management)	ทักษะในการทำความเข้าใจและเข้าถึงความต้องการของลูกค้าอย่างแท้จริง ด้วยการสกัดความต้องการ ตรวจสอบและนิยามความต้องการ เพื่อนำไปสร้างเป็นข้อกำหนดความต้องการด้านระบบหรือซอฟต์แวร์
3.	ทักษะการประมาณการแรงงาน (Effort Estimation)	ทักษะในการประมาณการแรงงาน (Effort) ที่ใช้ในการพัฒนาซอฟต์แวร์ โดยใช้เทคนิคต่างๆ เช่น การประมาณ

#	ทักษะด้านเทคนิค (Technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
		การจากประสบการณ์ การประมาณการโดยใช้หลักทางสถิติ และเทคนิคอื่นๆ
4.	ทักษะการวางแผน (Planning)	ทักษะการบริหารจัดการระบบความคิด หรือความสามารถในการคิดวางแผนหรือคาดการณ์ล่วงหน้า เพื่อให้บรรลุถึงเป้าหมาย
5.	ทักษะการลำดับความสำคัญของงาน (Advance prioritization)	ทักษะในการจัดลำดับความสำคัญของงาน (Task) รู้ว่างานอะไร ที่ต้องทำก่อน-ทำหลัง และงานใจต้องรอให้งานอื่นเสร็จเรียบร้อยก่อน ถึงจะสามารถทำงานนั้นได้
6.	ทักษะการเตรียมสภาพแวดล้อมที่เหมาะสมกับการพัฒนาซอฟต์แวร์ (Environment preparation)	ทักษะในการเตรียมสภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ให้พร้อมสำหรับการทำงาน
7.	ทักษะการใช้ดีไซน์แพทเทิร์น (Design pattern)	<p>ทักษะในการนำดีไซน์แพทเทิร์นมาประยุกต์ใช้ในการพัฒนาซอฟต์แวร์</p> <p style="text-align: center;">หมายเหตุ</p> <p>ดีไซน์แพทเทิร์น เป็นรูปแบบหรือแบบแผนที่ถูกสร้างขึ้น และเป็นที่รู้จักกันอย่างแพร่หลายที่ถูกนำมาใช้ในการแก้ปัญหาการออกแบบพัฒนาซอฟต์แวร์ โดยแนวทางเหล่านี้ไม่ได้มีรูปแบบตายตัวที่สามารถนำไปใช้งานได้โดยตรง แต่เป็นเพียงการอธิบายแนวทางที่สามารถนำไปประยุกต์ใช้ในสถานการณ์ต่างๆ ซึ่งนิยมนำมาใช้กับการพัฒนาแบบ Object Oriented</p>
8.	ทักษะการหากรณีการทดสอบให้ครอบคลุม (Code coverage)	ทักษะในการหาเซตของกรณีการทดสอบ (Set of Test Cases) ที่สามารถครอบคลุมการทดสอบทั้งหมด โดยการลดกรณีทดสอบที่มีความซ้ำซ้อน
9.	ทักษะในการทวนสอบโค้ด (Code review)	ทักษะในการทวนสอบความสอดคล้องระหว่างโค้ดที่พัฒนา และการออกแบบ การทวนสอบโค้ดมีวัตถุประสงค์เพื่ออ่านและทำความเข้าใจโค้ด ดูความถูกต้องของตรรกะ

#	ทักษะด้านเทคนิค (Technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
		ความเป็นเหตุและผล รวมถึงโครงสร้างของโค้ดดังกล่าวว่าสามารถอ่าน หรือทำความเข้าใจได้หรือไม่
10.	ทักษะการระบุโค้ดที่น่าจะนำไปสู่ปัญหา (Code smelling)	ทักษะในการพิจารณาหาโค้ดที่น่าจะนำไปสู่ปัญหาในอนาคต ทั้งจากการออกแบบหรือพัฒนาซอฟต์แวร์ รวมถึงส่วนที่ยากต่อการทำความเข้าใจหรือยากต่อการบำรุงรักษาภายหลัง
11.	ทักษะการเขียนโค้ดตามมาตรฐาน (Coding standard)	ทักษะในการเขียนโค้ดให้เป็นไปตามมาตรฐานที่ทีมกำหนดไว้
12.	ทักษะการออกแบบและการเขียนโปรแกรมเชิงวัตถุ (OO-concepts)	ทักษะในการออกแบบและพัฒนาซอฟต์แวร์เชิงวัตถุ โดยใช้แนวคิดต่างๆ เช่น การห่อหุ้มส่วนต่าง ๆ (Encapsulation) เพื่อบดบังรายละเอียด (Information Hiding) ส่วนที่ไม่ต้องการให้เห็นไว้และยังมีแนวคิดอื่น ๆ ที่เกี่ยวข้อง เช่น การสืบทอดคลาส (Inheritance) และโพลิมอร์ฟิซึม (Polymorphism) เป็นต้น
13.	ทักษะการเขียนโปรแกรม (Programming)	ทักษะการเขียนโค้ดตามไวยากรณ์ของแต่ละภาษาคอมพิวเตอร์ ให้โปรแกรมหรือซอฟต์แวร์สามารถทำงานได้อย่างถูกต้องตามที่กำหนดไว้
14.	ทักษะการเขียนโปรแกรมเป็นคู่ (Pair programming)	ทักษะในการพัฒนาซอฟต์แวร์ โดยที่นักพัฒนาซอฟต์แวร์ 2 คน ช่วยเหลือและสื่อสารกัน เพื่อทำงานเดียวกันให้ประสบความสำเร็จโดยใช้คอมพิวเตอร์เครื่องเดียวกัน มีหนึ่งคนเป็นคนทำหน้าที่ควบคุมเมาส์และคีย์บอร์ดเรียกว่า "Driver" ขณะที่อีกคนคอยสังเกตการณ์ดูโค้ดที่กำลังถูกพัฒนาและให้คำแนะนำในการพัฒนาซึ่งเรียกว่า "Navigator" และมีการสลับตำแหน่งกันไปมาในการทำงาน
15.	ทักษะฐานข้อมูล (Database)	ทักษะในการใช้และบริหารจัดการข้อมูลในฐานข้อมูล
16.	ทักษะในการโค้ดให้ง่ายต่อการเขียนอ่าน และดูแลรักษา	ทักษะในการโค้ดให้ง่ายต่อการเขียน อ่าน และดูแลรักษา

#	ทักษะด้านเทคนิค (Technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
	(Clean Code)	
17.	ทักษะการเปลี่ยนแปลงโครงสร้างภายในโค้ด เพื่อให้ง่ายต่อการทำความเข้าใจ (Code refactoring)	ทักษะในการเปลี่ยนแปลงโครงสร้างภายในของโค้ด เพื่อให้ง่ายต่อการทำความเข้าใจและลดความซับซ้อนของโค้ด แต่ไม่ได้มีการเปลี่ยนแปลงพฤติกรรมภายนอกของระบบหรือเพิ่มฟังก์ชันการทำงานใหม่
18.	ทักษะการทดสอบ (Testing)	ทักษะในการทดสอบซอฟต์แวร์ว่าสามารถทำงานได้อย่างถูกต้องตามที่ได้กำหนดไว้ โดยใช้กรณีทดสอบ ระยะเวลาในการทดสอบ และค่าใช้จ่ายในการทดสอบที่น้อยที่สุด
19.	ทักษะการตรวจสอบความผิดพลาด (Debugging)	ทักษะในการค้นหาส่วนที่มีปัญหาในโปรแกรมและทำการแก้ไข
20.	ทักษะการรวบรวมโค้ดที่ได้รับการพัฒนาจากสมาชิกแต่ละคนในทีมให้เป็นชิ้นเดียว (Continuous integration)	ทักษะในการรวบรวมโค้ดที่ได้รับการพัฒนาจากสมาชิกแต่ละคนในทีมให้เป็นชิ้นเดียว จากนั้นทำการทดสอบ เพื่อตรวจสอบว่าโค้ดแต่ละส่วนเข้ากันได้หรือไม่
21.	ทักษะการบริหารจัดการการนำผลงานเข้าสู่การให้บริการ (Release management)	ทักษะปล่อยซอฟต์แวร์เข้าสู่การให้บริการจริง ต้องวางแผนร่วมกับผู้ที่เกี่ยวข้องทั้งหมดเพื่อกำหนดขั้นตอนปฏิบัติงาน
22.	ทักษะการบริหารความเสี่ยง (Risk management)	ทักษะในการวิเคราะห์ ประเมิน จัดการ ติดตาม และสื่อสารความเสี่ยงที่เกี่ยวข้องกับกิจกรรมหน่วยงาน/ฝ่ายงาน เพื่อช่วยลดความสูญเสียในการไม่บรรลุเป้าหมายให้เหลือน้อยที่สุด
23.	ทักษะการใช้เทคโนโลยีสารสนเทศ (Use ICT)	ทักษะในการนำเครื่องมือ อุปกรณ์ และเทคโนโลยีดิจิทัลที่มีอยู่ในปัจจุบัน อาทิ คอมพิวเตอร์ โทรศัพท์ แท็บเล็ต โปรแกรมคอมพิวเตอร์ และสื่อออนไลน์ มาใช้ให้เกิดประโยชน์สูงสุด ในการสื่อสาร การปฏิบัติงาน และการทำงานร่วมกัน หรือใช้เพื่อพัฒนากระบวนการทำงาน หรือระบบงานในองค์กรให้มีความทันสมัยและมีประสิทธิภาพ

4.4 ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills)

เพื่อตอบคำถามงานวิจัยข้อที่ 2 “ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) ใด ที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบเอจิล (Agile)” ผู้วิจัยสังเคราะห์ข้อมูลที่ได้จากการรวบรวม และนำเสนอทักษะ จำนวน 18 ทักษะ ดังแสดงในตารางที่ 4.4

ตารางที่ 4.3 ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills)

#	ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
1.	ทักษะการสื่อสาร (Communication)	ทักษะในการถ่ายทอดข้อมูล ข่าวสาร หรือแลกเปลี่ยนความรู้ความคิดเห็น จากบุคคลหนึ่งไปยังอีกบุคคลหนึ่ง มีวัตถุประสงค์เพื่อให้ผู้รับสารเข้าใจสิ่งที่ต้องการสื่อสาร
2.	ทักษะการทำงานเป็นทีม (Teamwork / teambuilding)	ทักษะในการทำงาน การให้ความร่วมมือ และช่วยเหลือสมาชิกภายในทีม ในการทำกิจกรรม หรืองานใด ๆ เพื่อให้บรรลุวัตถุประสงค์หรือเป้าหมายของทีมที่ได้วางไว้
3.	ทักษะการแก้ไขปัญหา/การคิดอย่างเป็นระบบ (Problems solving/ System thinking)	ทักษะการคิด วิเคราะห์ อย่างเป็นเหตุเป็นผล เพื่อหาแนวทางแก้ไขปัญหา โดยใช้เทคนิคต่างๆ โดยการนำข้อมูลมาเชื่อมโยงกับประสบการณ์เดิม
4.	ทักษะการจัดการความเครียด (Handling stress)	ทักษะในการจัดการ ควบคุม และบำบัดความเครียดให้น้อยลงหรือให้หมดสิ้นไป ในระหว่างการพัฒนาซอฟต์แวร์
5.	ทักษะการเรียนรู้ด้วยตนเอง (Self-learning)	ทักษะในการเรียนรู้ ทำความเข้าใจสิ่งใหม่ ๆ ทั้งที่สิ่งดังกล่าวเป็นสิ่งที่ ตนยังไม่เคยมีความรู้ หรือไม่มีประสบการณ์มาก่อน
6.	ทักษะการบริหารจัดการ (Management)	ทักษะในการบริหารงานตามภารกิจที่ได้รับมอบหมายอย่างรวดเร็วและถูกต้อง โดยใช้ทรัพยากรที่มีอยู่ให้น้อยที่สุด
7.	ทักษะการเจรจาต่อรอง (Negotiation)	ทักษะในการเจรจาต่อรองกับบุคคลใดๆ เพื่อให้ได้ข้อตกลงร่วมกัน โดยก่อนหน้านั้นความคิดเห็นของทั้งสองฝ่ายมีความแตกต่างกัน

#	ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
8.	ทักษะมารยาทไทย (Thai Courtesy)	ทักษะหรือแนวทางในการปฏิบัติ การแสดงวาจา ภาษา ท่าทาง และพฤติกรรมต่าง ๆ ออกมาให้ปรากฏแก่สายตา ของผู้อื่นคนทั่ว ๆ ไป
9.	ทักษะการจัดการการเปลี่ยนแปลง (Dealing with change)	ทักษะในการจัดการกับการเปลี่ยนแปลงที่เกิดขึ้นใน กระบวนการพัฒนา เรียกว่าเป็นการตอบรับการ เปลี่ยนแปลง ในขณะที่ต้องนำส่งซอฟต์แวร์ให้ได้อย่าง รวดเร็ว ทันตามกำหนดเวลา และมีความยืดหยุ่นในการ ทำงาน
10.	ทักษะความรับผิดชอบ (Responsibility)	ทักษะความสำนึกได้อยู่เสมอในการปฏิบัติงาน เอาใจใส่ ติดตามผลไม่ทอดทิ้ง ยอมรับผิดชอบผลของหน้าที่การงาน ที่ตนเองกระทำทั้งดีและไม่ดี
11.	ทักษะการสร้างสัมพันธภาพและ เข้าใจกัน (Diplomacy)	ทักษะในการสร้างความเข้าใจ และประสานสัมพันธ์ไมตรี ระหว่างกัน เน้นการจัดการความสัมพันธ์อย่างชาญฉลาด
12.	ทักษะความเป็นมืออาชีพ (Professionalism)	ทักษะความรู้ เชี่ยวชาญชำนาญพิเศษในวิชาชีพของตน สามารถปฏิบัติงานได้ตามมาตรฐานอย่างถูกต้อง มีไหวพริบ ในการจัดการแก้ไขปัญหาได้อย่างรวดเร็วแม่นยำ และลง มือทำอย่างจริงจัง
13.	ทักษะการเป็นผู้นำ (Leadership)	ทักษะหรือความสามารถของบุคคลที่มีอิทธิพลกับกลุ่มคน เพื่อช่วยนำ และสนับสนุนกลุ่มคนดังกล่าวให้ทำงานตามที ได้รับมอบหมายได้
14.	ทักษะการเป็นผู้ประกอบการ (Entrepreneurship)	ทักษะการคิดแบบผู้ประกอบการ มองโอกาสและความ เสี่ยง วิเคราะห์ต้นทุน หาช่องทางในการประกอบธุรกิจ เลือกลงและเอาใจใส่ลูกค้า
15.	ทักษะการสร้างแรงจูงใจ (Motivation)	ทักษะในการโน้มน้าว สร้างแรงจูงใจในการปฏิบัติงาน
16.	ทักษะการอำนวยความสะดวก (Facilitation)	ทักษะในการให้บริการที่ดี ให้ความช่วยเหลือ และแสดง ไมตรีจิต มีความกระตือรือร้น ทำให้ผู้ใช้บริการรู้สึกอบอุ่น

#	ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
17.	ทักษะการปรับตัว (Adaptability)	ทักษะในการยอมรับการเปลี่ยนแปลง และปรับปรุงตนเอง ให้สามารถตอบสนองต่อการเปลี่ยนแปลง
18.	ทักษะการแบ่งปันข้อมูล (Share knowledge)	ทักษะการนำเสนอและแบ่งปันข้อมูลคือการเผยแพร่ข้อมูล ไปยังผู้ที่สนใจ โดยใช้เทคนิคต่างๆ อย่างเหมาะสม

4.5 ทักษะที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)

เพื่อตอบคำถามงานวิจัยข้อที่ 3 “ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes) ใด ที่จำเป็นสำหรับ ผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern software development process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) และแบบ เอจิลล์ (Agile)” ผู้วิจัยสังเคราะห์ข้อมูลที่ได้จากการรวบรวม และนำเสนอทัศนคติ จำนวน 7 ทักษะ ดังแสดง ในตารางที่ 4.5

ตารางที่ 4.4 ทักษะที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)

#	ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
1.	เคารพและไว้วางใจกัน (Respect and trust)	มีความไว้วางใจและการเคารพซึ่งกันและกัน เชื่อว่า สมาชิกของทีมแต่ละคนมีความรู้และความสามารถเพียงพอที่จะทำงานให้สำเร็จจูล่วงลงได้ เพราะสมาชิกของทีมมีความหลากหลายทั้งในแง่ของประสบการณ์พื้นฐาน และ วัฒนธรรม
2.	ตระหนักใจเรื่องคุณภาพ (Focus on quality)	มีมุมมองหรือการตระหนักถึงเรื่องคุณภาพอยู่ตลอดเวลา โดยเชื่อว่าการพัฒนาซอฟต์แวร์ให้มีคุณภาพตลอดเวลาและทุกขั้นตอน จะช่วยทำให้ซอฟต์แวร์มีข้อผิดพลาดน้อยลง
3.	มีทัศนคติในแง่บวก (Positive attitude)	มีมุมมองด้านดีกับผู้อื่น โดยเริ่มจากการคิดดี ด้วยความรู้สึที่ดี และทำสิ่งที่ดี
4.	ให้ความสำคัญกับเวลา (Focus on time boxing)	มีมุมมองในเรื่องการเคารพกรอบเวลา (Time Boxing) ที่ได้ตั้งไว้ โดยเมื่อถึงเวลาที่กำหนดไม่ว่างานจะเสร็จตามกำหนดเวลาหรือไม่ จำเป็นต้องหยุดทันที เพื่อทำกิจกรรมถัดไป
5.	ให้ความสำคัญกับซอฟต์แวร์ที่ทำงานได้จริง (Focus on working software)	มีมุมมองหรือการตระหนักว่าซอฟต์แวร์ที่ทำงานได้จริงมีความสำคัญมากกว่าเอกสารหรือไดอะแกรม ที่ต้องนำเสนอลูกค้า และสามารถนำมาใช้ในการวัดความก้าวหน้าของงานได้

#	ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
6.	เชื่อในความเรียบง่าย (Belief in simplicity)	มีมุมมองในการทำสิ่งต่าง ๆ ให้ง่ายและทำเท่าที่จำเป็น แต่ ยังต้องเพียงพอกับความต้องการ มุมมองดังกล่าวจะทำให้ งานที่ต้องทำมีจำนวนไม่มากเกินไป
7.	มีความกล้า (Courage)	มีมุมมองที่กล้าที่จะเผชิญหน้ากับปัญหาต่าง ๆ ที่เกิดขึ้น ในช่วงของการพัฒนา ทำให้กิจกรรมสามารถสำเร็จไปได้ รวมถึงกล้าที่จะสื่อสารหรือบอกความผิดพลาดของตนเองที่ เกิดขึ้น

บทที่ 5 สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

งานวิจัยนี้ศึกษา รวบรวม และนำเสนอทักษะที่จำเป็น สำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ เพื่อให้ผู้ปฏิบัติงานสามารถเลือกทักษะดังกล่าวไปใช้ในการพัฒนาตนเอง โดยใช้วิธีวิจัยแบบการทบทวนอย่างเป็นระบบ (Systematic review) ซึ่งเป็นวิธีการที่ใช้เก็บรวบรวมและเรียบเรียงความรู้ในการสืบค้นข้อมูล และสังเคราะห์เป็นข้อมูลใหม่ โดยภายในงานวิจัยนี้แบ่งขั้นตอนออกเป็น 4 ขั้นตอนหลักๆ ได้แก่ 1) การกำหนดปัญหาและรวบรวมข้อมูลเกี่ยวกับทักษะที่จำเป็น 2) การสังเคราะห์ข้อมูลทั้งหมดที่ได้รับ 3) การแปลผล และ 4) การวิพากษ์ผลลัพธ์โดยผู้เชี่ยวชาญ ทั้งนี้ผู้วิจัยได้รวบรวมทักษะจากบทความวิจัยได้ทั้งหมด 123 ทักษะ และนำมาสังเคราะห์ร่วมกันพบว่า บางทักษะมีความคล้ายคลึงกันอย่างชัดเจน ผู้วิจัยจึงยุบรวมบางทักษะเข้าด้วยกัน โดยเหลือทักษะทั้งสิ้น 48 ทักษะ ซึ่งทักษะดังกล่าวถูกแบ่งออกเป็น 3 ประเภทหลัก ได้แก่ ทักษะด้านเทคนิค (Technical skills) จำนวน 23 ทักษะ ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical skills) จำนวน 18 ทักษะ และทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes) จำนวน 7 ทักษะ ทั้งนี้ผู้วิจัยได้เสนอผลงานวิจัยต่อผู้เชี่ยวชาญด้านสกริมที่ได้รับใบรับรองจากสถาบันที่ให้การรับรอง และผู้เชี่ยวชาญด้านการใช้งานสกริมซึ่งเป็นผู้บริหารระดับสูงสุดของบริษัท (CEO) เพื่อตรวจสอบความถูกต้อง นอกจากนี้ผู้วิจัยนำเสนอผลวิจัยสุดท้ายให้แก่คณาจารย์หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา เพื่อให้ผู้สอนสามารถนำทักษะในงานวิจัยนี้ ไปใช้ในการจัดการเรียนการสอน เพื่อส่งเสริมให้ผู้เรียนมีทักษะที่ตรงต่อความต้องการกับภาคอุตสาหกรรมพัฒนาซอฟต์แวร์

5.2 ข้อจำกัดของงานวิจัย

- 1) เนื่องจากกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ ไม่ได้กำหนดกรอบทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานไว้อย่างชัดเจน งานวิจัยนี้พยายามศึกษา รวบรวม และนำเสนอทักษะจากบทความวิจัยที่เกี่ยวข้อง ดังนั้นหากมีการระบุมอบทักษะอย่างเป็นทางการในภายหลัง อาจมีความคลาดเคลื่อนจากความเป็นจริงระดับหนึ่ง
- 2) เนื่องจากบทความวิจัยที่มีความน่าเชื่อถือเกี่ยวกับทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ โดยเฉพาะอย่างยิ่งการพัฒนาซอฟต์แวร์สมัยใหม่ยังมีค่อนข้างน้อย หากยังมีปริมาณงานวิจัยที่มากขึ้น จะส่งผลดีต่อความแม่นยำในการวิจัย

5.3 ข้อเสนอแนะในอนาคต

- 1) เพื่อให้สะท้อนภาพลักษณ์ของบริบทประเทศไทยได้อย่างชัดเจน ควรสอบถามความคิดเห็นเพิ่มเติมกับผู้มีส่วนได้ส่วนเสียในการใช้กระบวนการพัฒนาซอฟต์แวร์สมัยใหม่
- 2) เนื่องจากในทีมสกรีมมีทั้งบทบาท สกรีมมาสเตอร์ และสมาชิกภายในทีม ควรพิจารณาจำแนกทักษะตามบทบาทหน้าที่ในอนาคต

บรรณานุกรม

- [1] พระราชบัญญัติส่งเสริมการพัฒนาฝีมือแรงงาน, (2019), บทบาท ภารกิจกรมพัฒนาฝีมือแรงงาน ตามพระราชบัญญัติส่งเสริมการพัฒนาฝีมือแรงงาน พ.ศ. 2545 และที่แก้ไขเพิ่มเติม, [Online], Available: <https://www.dsd.go.th/DSD/Doc/Download/18299>.
- [2] Spencer, L.M., Spencer, S.M. (1993). **Competency at Work: Models for Superior Performance**. New York: John Wiley & Sons.
- [3] พจนานุกรมฉบับราชบัณฑิตยสถาน, **ความหมายของคำว่าทักษะ**, (2022), [Online], Available: <https://dictionary.orst.go.th>.
- [4] Axel, L., (2009), **Requirements Engineering From System Goals to UML Models to Software Specifications (Chapter 1)** , Wiley, ISBN 978-0-470-01270-3.
- [5] Neon Rain Interactive, **AGILE SCRUM FOR WEB DEVELOPMENT**, (2022), [Online], Available: <https://www.neonrain.com/agile-scrum-web-development/>.
- [6] Kenneth, R., (2012), **Essential Scrum: a practical guide to the most popular agile process**, First printing, Pearson Education, Inc., USA, 198- 211.
- [7] Nils, B., Torgeir, D., (2008), **Scrum and team effectiveness: theory and practice**, Agile Processes in Software Engineering and Extreme Programming Lecture Notes in Business Information Processing, Vol. 9, 11-20.
- [8] Hidayati, A., Budiardjo, E., Purwandari, B. (2020). **Hard and Soft Skills for Scrum Global Software Development Teams**, ICSIM '20: The 3rd International Conference on Software Engineering and Information Management, 110-114.
- [9] Prasad, P., (2010), **Skills for Scrum Agile Teams**, Agile Record, July 2010, Issue 3, 55-57.
- [10] Alistair, C., Jim, Hi., (2001), **Agile Software Development: The People Factor**, Computer, Vol. 34, Issue 11, 131-133.
- [11] Tan, C., Teo, H., (2007), **Training Future Software Developers to Acquire Agile Development Skills**, Communication of The ACM, Vol. 50, No. 12, 97-98.
- [12] Matturro, G., Raschetti, F., Fontán, C., (2015). **Soft Skills in Software Development Teams. A Survey of the Points of View of Team Leaders and Team Members**.

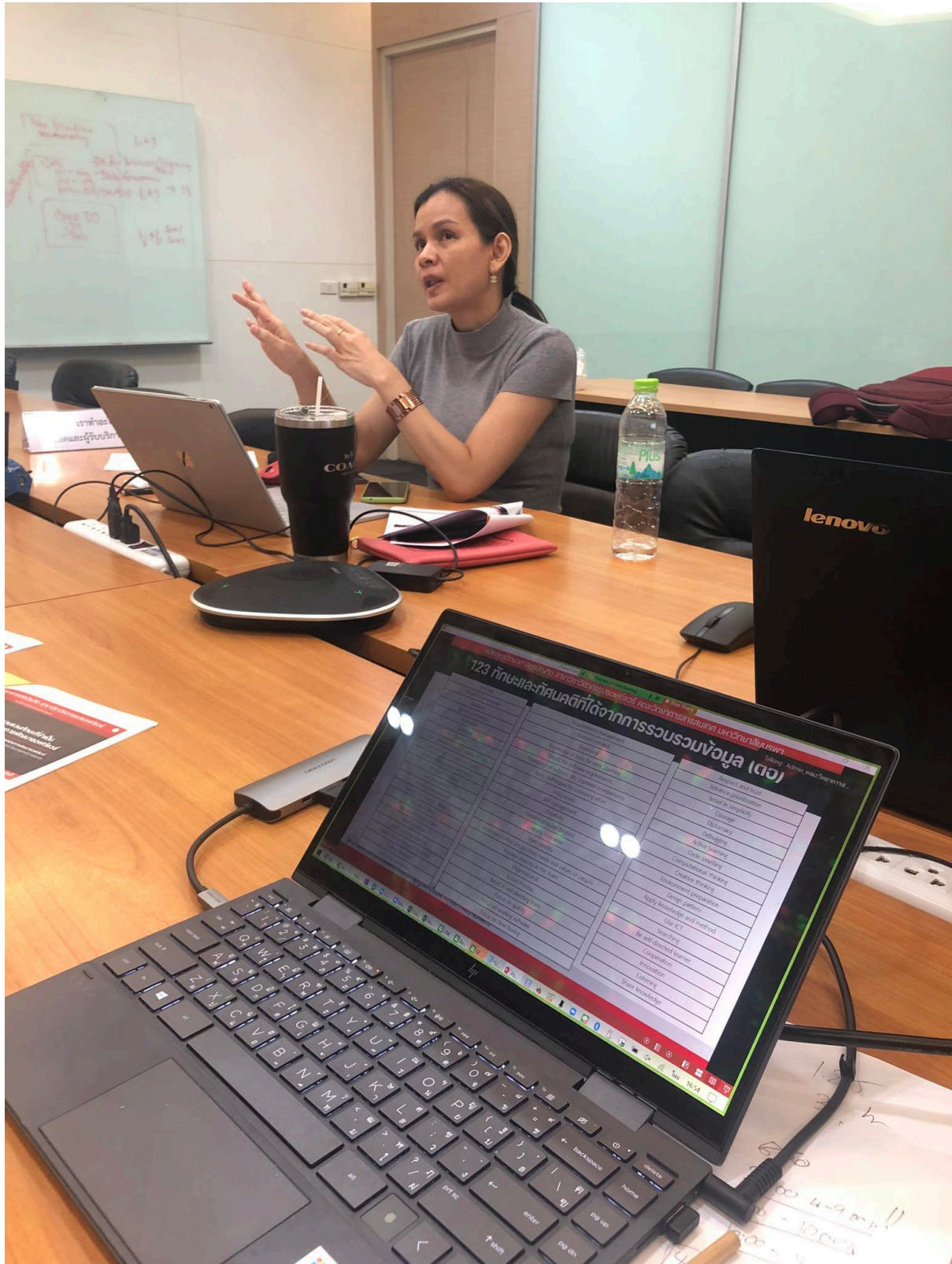
- 8th International Workshop on Cooperative and Human Aspects of Software Engineering, Italy.
- [13] Gonzalez-Morales, D., Moreno, L., Roda-García, J., (2011). **Teaching “soft” skills in Software Engineering**. Global Engineering Education Conference (EDUCON), 630 - 637.
- [14] Paasivaara, M., Lassenius, C., Damian, D., Raty, P., & Schroter, A. (2013). **Teaching students global software engineering skills using distributed scrum**. Paper presented at the Proceedings - International Conference on Software Engineering, 1128-1137.
- [15] Chassidim, H., Almog, D., & Mark, S. (2018). **Fostering soft skills in project-oriented learning within an agile atmosphere**. European Journal of Engineering Education, 43(4), 638-650.
- [16] Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2020). **From agile to DevOps: Smart skills and collaborations**. Information Systems Frontiers, 22(4), 927-945.
- [17] Kropp, M., Meier, A., & Biddle, R. (2016). **Teaching agile collaboration skills in the classroom**. Paper presented at the Proceedings - 2016 IEEE 29th Conference on Software Engineering Education and Training, CSEEdT 2016, 118-127.
- [18] Oyetoyan, T. D., Cruzes, D. S., & Jaatun, M. G. (2016). **An empirical study on the relationship between software security skills, usage and training needs in agile settings**. Paper presented at the Proceedings - 2016 11th International Conference on Availability, Reliability and Security, ARES 2016, 548-555.
- [19] Valentin, E., Carvalho, J. R. H., & Barreto, R. (2015). **Rapid improvement of students' soft-skills based on an agile-process approach**. Paper presented at the Proceedings - Frontiers in Education Conference, FIE, , 2015
- [20] Read, A., Derrick, D. C., & Ligon, G. S. (2014). **Developing entrepreneurial skills in IT courses: The role of agile software development practices in producing successful student initiated products**. Paper presented at the Proceedings of the Annual Hawaii International Conference on System Sciences, 201-209.
- [21] Diel, E., Bergmann, M., Marczak, S., & Luciano, E. (2017). **What is agile, which practices are used, and which skills are necessary according to brazilian**

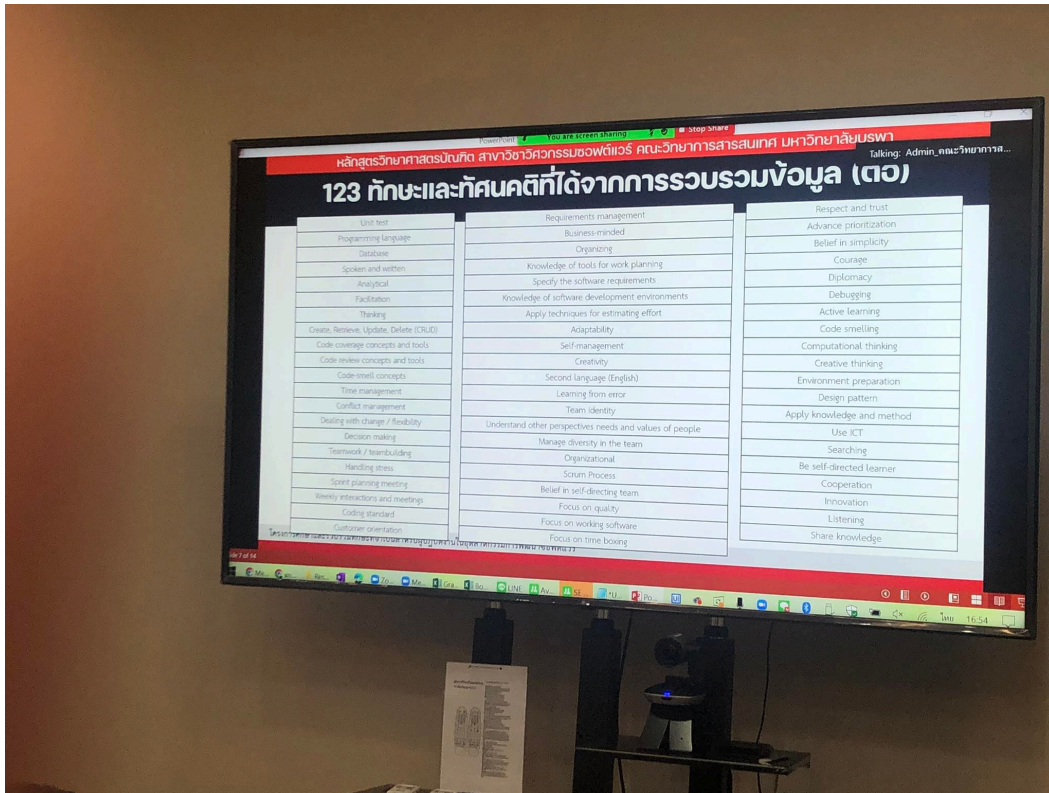
- professionals: Findings of an initial survey.** Paper presented at the Proceedings - 6th Brazilian Workshop on Agile Methods, WBMA 2015, 18-24.
- [22] Hidayati, A., Budiardjo, E. K., & Purwandari, B. (2020). **Hard and soft skills for scrum global software development teams.** Paper presented at the ACM International Conference Proceeding Series, 110-114.
- [23] Tolfo, C., Wazlawick, R. S., Ferreira, M. G. G., & Forcellini, F. A. (2018). **Agile practices and the promotion of entrepreneurial skills in software development.** Journal of Software: Evolution and Process, 30(9)
- [24] Gren, L., Knauss, A., & Stettina, C. J. (2018). **Non-technical individual skills are weakly connected to the maturity of agile practices.** Information and Software Technology, 99, 11-20.
- [25] Cornide-Reyes, H., Campillay, S., Alfaro, A., & Villarroel, R. (2019). **Identification of skills for the formation of agile high performance teams: A systematic mapping.** Information and Communication Technologies of Ecuador (TIC.EC), 141-152.
- [26] Maturro, G., Fontán, C., & Raschetti, F. (2015). **Soft skills in scrum teams: A survey of the most valued to have by product owners and scrum masters.** Paper presented at the Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, , 2015-January 42-45.
- [27] Omar, M., Khasasi, N. L. A., Abdullah, S. L. S., Hashim, N. L., Romli, R., & Katuk, N. (2018). **Defining skill sets requirements for agile scrum team formation.** Journal of Engineering and Applied Sciences, 13(3), 784-789.
- [28] Bootla, P., Rojanapornpun, O., & Mongkolnam, P. (2015). **Necessary skills and attitudes for development team members in scrum: Thai experts' and practitioners's perspectives.** Paper presented at the Proceedings of the 2015 12th International Joint Conference on Computer Science and Software Engineering, JCSSE 2015, 184-189.
- [29] Lutsenko, G., & Lucenko, G. (2020). **Work in progress: Fostering soft-skills of engineering students within scrum projects.** Paper presented at the IEEE Global Engineering Education Conference, EDUCON, , 2020-April 1723-1727.

- [30] Garcia, G. D., Pardo, C., & Alvarez, F. (2022). **Society 5.0 and soft skills in agile global software development**. Revista Iberoamericana De Tecnologias Del Aprendizaje, doi:10.1109/RITA.2022.316696

ภาคผนวก

ภาพการนำเสนอผลวิจัยสุดท้ายให้แก่คณาจารย์หลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
วันที่ 2 มิถุนายน 2565







สไลด์การนำเสนอผลวิจัยสุดท้ายให้แก่คณาจารย์หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

วันที่ 2 มิถุนายน 2565

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ●
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

การนำเสนอผลการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์
The Study and Collect of Necessary Skills for Practitioners in The Software Development Industry

Soft-En ดีเด่น เลิศล้ำ นวัตกรรม มิวินัย ร่วมใจ สามัคคี

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

ประเด็นการนำเสนอ

Research Objectives
วัตถุประสงค์โครงการและแผนการดำเนินงาน

Data Collection
การศึกษาและรวบรวมข้อมูล

Necessary Skills
ทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ หน้า 2

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ●
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

วัตถุประสงค์โครงการและแผนการดำเนินงาน

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์
The Study and Collect of Necessary Skills for Practitioners in The Software Development Industry

Soft-En ดีเด่น เลิศล้ำ นวัตกรรม มิวินัย ร่วมใจ สามัคคี

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ●
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

วัตถุประสงค์โครงการ

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์

ข้อที่ 1 เพื่อทราบทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์

ข้อที่ 2 เพื่อให้ผู้สอนสามารถนำทักษะในมาวิจัยนี้ ไปใช้ในการจัดการเรียนการสอน เพื่อส่งเสริมให้ผู้เรียนมีทักษะที่ตรงต่อความต้องการกับภาคอุตสาหกรรมพัฒนาซอฟต์แวร์

ข้อที่ 3 เพื่อให้ผู้เรียนสามารถนำทักษะในมาวิจัยนี้ ไปพัฒนาทักษะของตนเองที่ยังไม่เชี่ยวชาญ เพื่อเพิ่มความชำนาญในการปฏิบัติงาน

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ หน้า 4

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

แผนการดำเนินงาน

● ดำเนินการเสร็จสิ้น ● กำลังดำเนินการ ● รอดำเนินการ

หัวข้อดำเนินงาน/รายละเอียดกิจกรรม	2564					2565				
	ค.ค.	ค.ค.	พ.ค.	ค.ค.	ค.ค.	ค.ค.	ค.ค.	พ.ค.	ค.ค.	ค.ค.
1. วางแผนจัดทำแผนงาน/โครงการวิจัย	■	■								
2. ศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ ของกรมวิชาการพัฒนาซอฟต์แวร์ (Modern Software Development Process) โดยเอกสารอ้างอิง การสัมภาษณ์ผู้มีส่วนเกี่ยวข้อง	■	■	■	■	■					
3. วิเคราะห์ ปรับปรุงแก้ไข จัดกลุ่ม และจำแนกทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์						■	■	■	■	■
4. กำหนดนิยามความหมาย (Definition of terms) ของผลวิจัยที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ เพื่อตรวจสอบความหมายของผลวิจัย										■
5. วิเคราะห์ผลการวิเคราะห์ โดยผู้เชี่ยวชาญและผู้ทรงคุณวุฒิ และนำเสนอผลการวิเคราะห์ที่มหาวิทยาลัยบูรพา สภาวิชาการมหาวิทยาลัยบูรพา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา อย่างน้อย 1 ครั้ง										■
6. เขียนรายงานฉบับสมบูรณ์										■

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ หน้า 5

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ●
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

การศึกษาและรวบรวมข้อมูล

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์
The Study and Collect of Necessary Skills for Practitioners in The Software Development Industry

Soft-En ดีเด่น เลิศล้ำ นวัตกรรม มิวินัย ร่วมใจ สามัคคี

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

ขั้นตอนการวิจัย

1. กำหนดปัญหาและรวบรวมข้อมูล (ศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ ของกรมวิชาการพัฒนาซอฟต์แวร์)
2. สังเคราะห์ (วิเคราะห์ ปรับปรุงแก้ไข จัดกลุ่ม และจำแนกทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์)
3. แปลผล (กำหนดนิยามความหมาย (Definition of terms) ของผลวิจัยที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์)
4. วิจารณ์ผล (เขียนรายงานฉบับสมบูรณ์)

Keywords: "TITLE ((SCUM OR AGILE) AND SKILLS))" Type: (Journal, Conference)
* First Filter (F1) = Only articles published between 2013 - 2022
** Second Filter (F2) = 10 Citations (min or equal to 3, and 25 Complete read)

SCOPUS: 54 Accepted, 10 Discarded
Others (SpringerLink, IEEE, ...): 225 Accepted, 200 Discarded
1F: 4 Accepted, 25 Discarded
2F: 12 Accepted, 20 Discarded
3F: 17 Articles

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ หน้า 7

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ●
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

123 ทักษะและทัศนคติที่ได้จากการรวบรวมข้อมูล

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์
The Study and Collect of Necessary Skills for Practitioners in The Software Development Industry

Soft-En ดีเด่น เลิศล้ำ นวัตกรรม มิวินัย ร่วมใจ สามัคคี

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

123 ทักษะและทัศนคติที่ได้จากการรวบรวมข้อมูล

Distributed communication	Integrity	System knowledge
Distributed teamwork	Interpersonal	Entrepreneurship
Building and negotiating trust	Positive attitude	Apply software development
Tool usage for distributed collaboration	Professionalism	Breaking down a project task into smaller tasks
Inter-cultural collaboration	Responsibility	Estimating effort needed to complete software development work
Team communication	Work ethic	Interacting with customers / project members
Overall agile development process	Continuous integration	Domain expertise
Problem solving	Clear Code	Backlog of the product
Self-learning	Collaboration	Behavior Object Development (BOD)
System thinking	Transparency and openness	Test Driven Development (TDD)
Coding and programming language	Code ownership	Short and frequent deliveries
Architecture	Quality	Burndown chart
Quality	Service design	Release planning
Release management	Static code analysis	Pair programming
Management	Security testing	Collective code ownership
Agile/lean	Threat modeling & risk management	Code re-use
Communication	Testing	Code refactoring
Courtesy	Dev operation	Scrum meetings
Flexibility	Punctuality	
	Leadership	

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ หน้า 9

● หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

123 ทักษะและทัศนคติที่ได้จากการรวบรวมข้อมูล (ต่อ)

Unit test	Requirements management	Respect and trust
Programming language	Business oriented	Advance prioritization
Database	Organizing	Belief in empathy
Speak and written	Knowledge of tools for work planning	Courage
Analytical	Specify the software requirements	Diplomacy
Facilitation	Knowledge of software development environments	Dealing
Thinking	Apply techniques for estimating effort	Active learning
Creates, Refines, Updates, Deletes (CRUD)	Adaptability	Code writing
Code coverage concepts and tools	Self-management	Computational thinking
Code review concepts and tools	Creativity	Creative thinking
Code-related concepts	Second language (English)	Environment preparation
Time management	Learning from error	Design pattern
Conflict management	Team identity	Apply knowledge and methods
Dealing with change / flexibility	Understand other perspectives needs and values of people	User U.I.
Decision making	Manage diversity in the team	Searching
Teamwork / team-building	Organizational	Be self-directed learner
Handling stress	Scrum Process	Cooperation
Scrum planning meeting	Build a self-directing team	Innovation
Weekly retrospectives and meetings	Focus on quality	Listening
Coding standard	Focus on working software	Share knowledge
Customer orientation	Focus on time losing	

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์ หน้า 10

การบูรณาการทักษะและทัศนคติที่ดีในแต่ละแหล่งข้อมูล

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์
The Study and Collect of Necessary Skills for Practitioners in The Software Development Industry

การบูรณาการทักษะและทัศนคติที่ดีในแต่ละแหล่งข้อมูล (2)

Problems solving/ System thinking	<ul style="list-style-type: none"> Problems solving System thinking Analytical Thinking Decision making Creativity Computational thinking Creative thinking Innovation 	Self-learning	<ul style="list-style-type: none"> Self-learning Learning from error Active learning Apply knowledge and method Searching Be self-directed learner
Testing	<ul style="list-style-type: none"> Testing Security testing Behavior Driven Development (BDD) Test Driven Development (TDD) Unit test 	Programming	<ul style="list-style-type: none"> Coding and programming languages Secure coding Static code analysis Programming language Create, Retrieve, Update, Delete (CRUD)
Planning	<ul style="list-style-type: none"> Knowledge of tools for work planning 	Environment preparation	<ul style="list-style-type: none"> Architecture Secure design Knowledge of software development environments Environment preparation
Adaptability	<ul style="list-style-type: none"> Adaptability 	Belief in simplicity	<ul style="list-style-type: none"> Belief in simplicity

การบูรณาการทักษะและทัศนคติที่ดีในแต่ละแหล่งข้อมูล

Communication	<ul style="list-style-type: none"> Distributed communication Team communication Communication Writing Oral presentation Interacting with customers / product owners Spoken and written Second language (English) 	Overall development process	<ul style="list-style-type: none"> Overall agile development process Scrum knowledge Agile software development Backlog of the product Burndown chart Kanban Scrum meeting Sprint planning meeting Weekly interactions and meetings Scrum Process Breaking down a project task into smaller
Teamwork / teambuilding	<ul style="list-style-type: none"> Distributed teamwork Tool usage for distributed collaboration Inter-cultural collaboration Teamwork Collaboration Teamwork / teambuilding Team identity Manage diversity in the team Belief in self-directing team 	Respect and trust	<ul style="list-style-type: none"> Building and maintaining trust Respect and trust
		Handling stress	<ul style="list-style-type: none"> Handling stress
		Requirement management	<ul style="list-style-type: none"> Requirement management Specify the software requirements
		Advance prioritization	<ul style="list-style-type: none"> Advance prioritization

การบูรณาการทักษะและทัศนคติที่ดีในแต่ละแหล่งข้อมูล (3)

Release management	<ul style="list-style-type: none"> Release management Short and frequently deliveries Release planning 	Responsibility	<ul style="list-style-type: none"> Integrity Responsibility Work ethic Transparency and openness Collective code ownership Organizational
Focus on quality	<ul style="list-style-type: none"> Quality Focus on quality 	Diplomacy	<ul style="list-style-type: none"> Interpersonal Diplomacy Understand other perspectives needs and values of people Listening
Management	<ul style="list-style-type: none"> Management Organizing Self-management 	Professionalism	<ul style="list-style-type: none"> Professionalism Craftsmanship Domain expertise
Negotiation	<ul style="list-style-type: none"> Negotiation Conflict management 	Positive attitude	<ul style="list-style-type: none"> Positive attitude
Dealing with change	<ul style="list-style-type: none"> Flexibility Dealing with change / flexibility 	Use ICT	<ul style="list-style-type: none"> Use ICT
Courtesy	<ul style="list-style-type: none"> Courtesy 		
Share knowledge	<ul style="list-style-type: none"> Share knowledge 		
Design pattern	<ul style="list-style-type: none"> Design pattern 		

การบูรณาการทักษะและทัศนคติที่ดีในแต่ละแหล่งข้อมูล (4)

Continuous integration	<ul style="list-style-type: none"> Continuous integration 	Motivation	<ul style="list-style-type: none"> Motivation
Clean Code	<ul style="list-style-type: none"> Clean Code 	Pair programming	<ul style="list-style-type: none"> Pair programming
OO-concepts	<ul style="list-style-type: none"> OO-concepts 	Code refactoring	<ul style="list-style-type: none"> Code refactoring
Risk management	<ul style="list-style-type: none"> Threat modeling & risk management 	Database	<ul style="list-style-type: none"> Database
Focus on time boxing	<ul style="list-style-type: none"> Punctuality Time management Focus on time boxing 	Facilitation	<ul style="list-style-type: none"> Facilitation
Leadership	<ul style="list-style-type: none"> Leadership 	Code coverage	<ul style="list-style-type: none"> Code coverage: concepts and tools
Entrepreneurship	<ul style="list-style-type: none"> Entrepreneurship Business-minded 	Code review	<ul style="list-style-type: none"> Code review: concepts and tools
Effort Estimation	<ul style="list-style-type: none"> Estimating effort needed to complete software development work Apply techniques for estimating effort 	Code smell	<ul style="list-style-type: none"> Code-smell concepts Code smelling
Debugging	<ul style="list-style-type: none"> Debugging 	Coding standard	<ul style="list-style-type: none"> Coding standard
		Focus on working software	<ul style="list-style-type: none"> Customer orientation Focus on working software
		Courage	<ul style="list-style-type: none"> Courage

ทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมพัฒนาซอฟต์แวร์
The Study and Collect of Necessary Skills for Practitioners in The Software Development Industry

23 ทักษะด้านเทคนิค (Technical Skills)

1. ทักษะการบูรณาการการพัฒนาโดยรวม (Overall development process)	12. ทักษะการเขียนโค้ดที่สะอาดง่ายในชนิด
2. ทักษะการเขียนโปรแกรม (Programming)	ที่ฝังในชุดภาษาที่ความเข้าใจ (Code refactoring)
3. ทักษะการเขียนสภาพแวดล้อมที่จำเป็นสำหรับการพัฒนาซอฟต์แวร์ (Environment preparation)	13. ทักษะการทดสอบ (Debugging)
4. ทักษะการทดสอบ (Testing)	14. ทักษะการทำการจัดการของโค้ดที่ครอบคลุม (Code coverage)
5. ทักษะการจัดการการควบคุมความเสี่ยงและการจัดการ (Risk management)	15. ทักษะในการตรวจสอบโค้ด (Code review)
6. ทักษะการจัดการการบูรณาการที่จำเป็นสำหรับการพัฒนาซอฟต์แวร์อย่างต่อเนื่อง (Continuous integration)	16. ทักษะการระบุกลิ่นที่ซ่อนอยู่ในโค้ด (Code smelling)
7. ทักษะในการใช้โค้ดที่สะอาดง่าย (Clean Code)	17. ทักษะการเขียนโค้ดตามมาตรฐาน (Coding standard)
8. ทักษะการออกแบบและการเขียนโปรแกรมเชิงวัตถุ (OO-concepts)	18. ทักษะการจัดการความเสี่ยงและการจัดการ (Requirement management)
9. ทักษะการจัดการความเสี่ยง (Risk management)	19. ทักษะการวางแผน (Planning)
10. ทักษะการประมาณการความพยายาม (Effort Estimation)	20. ทักษะการลำดับความสำคัญของการพัฒนา (Advance prioritization)
11. ทักษะการเขียนโปรแกรมคู่ (Pair programming)	21. ทักษะการตรวจสอบความผิดพลาด (Debugging)
	22. ทักษะการใช้รูปแบบที่ชัดเจน (Design pattern)
	23. ทักษะการใช้เครื่องมือที่ทันสมัย (Use ICT)

18 ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical Skills)

1. ทักษะการสื่อสาร (Communication)	10. ทักษะความรับผิดชอบ (Responsibility)
2. ทักษะการทำงานเป็นทีม (Teamwork / teambuilding)	11. ทักษะการรักษาสัมพันธภาพและเข้าใจกัน (Diplomacy)
3. ทักษะการแก้ไขปัญหา/การคิดอย่างเป็นระบบ (Problems solving/ System thinking)	12. ทักษะความเป็นมืออาชีพ (Professionalism)
4. ทักษะการจัดการความเครียด (Handling stress)	13. ทักษะการเป็นผู้นำ (Leadership)
5. ทักษะการเรียนรู้ด้วยตนเอง (Self-learning)	14. ทักษะการเป็นผู้ประกอบการ (Entrepreneurship)
6. ทักษะการบริหารจัดการ (Management)	15. ทักษะการริเริ่มแรงจูงใจ (Motivation)
7. ทักษะการเจรจาต่อรอง (Negotiation)	16. ทักษะการอำนวยความสะดวก (Facilitation)
8. ทักษะมารยาท (Courtesy)	17. ทักษะการปรับตัว (Adaptability)
9. ทักษะการจัดการการเปลี่ยนแปลง (Dealing with change)	18. ทักษะการแบ่งปันข้อมูล (Share knowledge)

7 ทัศนคติที่ดีต่อการพัฒนาซอฟต์แวร์ (Attitudes)

1. เคารพและไว้วางใจกัน (Respect and trust)
2. ตระหนักใจเรื่องคุณภาพ (Focus on quality)
3. มีทัศนคติในแง่บวก (Positive attitude)
4. ให้ความสำคัญกับเวลา (Focus on time boxing)
5. ให้ความสำคัญกับซอฟต์แวร์ที่ทำงานได้จริง (Focus on working software)
6. เชื่อในความเรียบง่าย (Belief in simplicity)
7. มีความกล้า (Courage)



ตัวอย่างผลการวิพากษ์โดยผู้เชี่ยวชาญ

112.	Computational thinking								/	
113.	Creative thinking								/	
114.	Environment preparation								/	
115.	Design pattern								/	
116.	Apply knowledge and method								/	
117.	Use ICT								/	
118.	Searching								/	
119.	Be self-directed learner								/	
120.	Cooperation								/	
121.	Innovation								/	
122.	Listening								/	
123.	Share knowledge								/	

ดิฉันคือผู้เชี่ยวชาญด้าน Soft skill, Technical Skill. ไม่
 ชอบครับ คือ Soft skill, Technical Skill. ไม่
 ชอบให้เขามีทักษะวิพากษ์ของครูหรือไม่
 ทักษะการเขียนของครูที่อาจ
 มีมันของ สังคมไทย

โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ 10

เทคนิค (Technical Skills) ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical Skills) และทัศนคติที่ต่อการพัฒนาซอฟต์แวร์ (Attitudes)

1.2.1 การยุบรวมทักษะและทัศนคติที่ได้ในแต่ละแหล่งข้อมูล

ผู้วิจัยสังเคราะห์ทักษะ/ทัศนคติที่ได้จากการรวบรวมข้อมูล จากทั้งหมด 123 ทักษะ ดังแสดงในตารางที่ 1.1 ข้างต้น เหลือเพียง 48 ทักษะ นอกจากนี้ผู้วิจัยได้เสนอชื่อทักษะใหม่ เพื่อเพิ่มความชัดเจนมากขึ้น ดังแสดงในตารางที่ 1.2

ตารางที่ 1.2 การยุบรวมทักษะและทัศนคติที่ได้ในแต่ละแหล่งข้อมูล

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
1.	Communication <i>Internal / Team External * Transpiracy</i>	<ul style="list-style-type: none"> ● Distributed communication ● Team communication ● Communication ● Writing ● Oral presentation ● Interacting with customers / product owners ● Spoken and written ● Second language (English)
2.	Teamwork / teambuilding	<ul style="list-style-type: none"> ● Distributed teamwork ● Tool usage for distributed collaboration

ไม่พอใจอาจมองว่าเป็น SET
 ของ Factor 3 ของ Model
 ขาดทักษะ

Group ๖๐๕ ฟังก์ชันผู้วิจัย โดยใช้เวลาพอสมควร
 ฝึกฝน

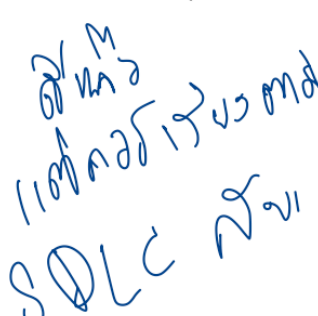
#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ : จากแต่ละแหล่งข้อมูล (Skills/attitudes name in articles)
46.	Design pattern	● Design pattern
47.	Use ICT	● Use ICT
48.	Share knowledge	● Share knowledge

1.2.2 ประเภทของทักษะและทัศนคติที่ผู้วิจัยเสนอเพื่อตอบคำถามงานวิจัย

#	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	ชื่อทักษะ/ทัศนคติ (Skills/attitude)
		● Self-management
14.	Negotiation	● Negotiation ● Conflict management
15.	Courtesy	● Courtesy
16.	Dealing with change	● Flexibility ● Dealing with change
17.	Responsibility	● Integrity ● Responsibility ● Work ethic

That's Custom, 99

ตารางที่ 1.3 ประเภทของทักษะและทัศนคติที่ผู้วิจัยเสนอเพื่อตอบคำถามงานวิจัย

#	ประเภททักษะ/ทัศนคติ (Skill type)	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)
1.	ทักษะด้านเทคนิค (Technical Skills) 	<ol style="list-style-type: none"> 1. Overall development process 2. Programming 3. Environment preparation 4. Testing 5. Release management 6. Continuous integration 7. Clean Code 8. OO-concepts 9. Risk management 10. Effort Estimation 11. Pair programming 12. Code refactoring 13. Database 14. Code coverage 15. Code review

เพื่อตอบคำถามงานวิจัยข้อที่ 2 “ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical Skills) ไต ที่จำเป็น สำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ของกระบวนการในการพัฒนาซอฟต์แวร์สมัยใหม่ (Modern Software Development Process) โดยเฉพาะอย่างยิ่ง การพัฒนาซอฟต์แวร์แบบสกรีม (Scrum) และแบบเอจิล (Agile)” ผู้วิจัยสังเคราะห์ข้อมูลที่ได้จากการรวบรวม และนำเสนอทักษะ จำนวน 18 ทักษะ ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical Skills)

#	ทักษะการเป็นผู้ปฏิบัติงานที่ดี (Non-technical Skills)	
	ชื่อทักษะ/ทัศนคติ : ผู้วิจัยเสนอ (New skills/attitudes name)	นิยาม/ความหมาย (Definition of terms)
1.	ทักษะการสื่อสาร (Communication)	ทักษะในการถ่ายทอดข้อมูล ข่าวสาร หรือแลกเปลี่ยน ความรู้ความคิดเห็น จากบุคคลหนึ่งไปยังอีกบุคคลหนึ่ง มีวัตถุประสงค์เพื่อให้ผู้รับสารเข้าใจสิ่งที่ต้องการสื่อสาร
2.	ทักษะการทำงานเป็นทีม (Teamwork / teambuilding)	ทักษะในการทำงาน การให้ความร่วมมือ และช่วยเหลือ สมาชิกภายในทีม ในการทำกิจกรรม หรืองานใด ๆ เพื่อให้

เขียนใหม่ให้สอดคล้องกับ
หรือ Automate
หรือ AI
หรือ ML
หรือ DL
หรือ NLP
หรือ CV
หรือ OCR
หรือ RPA
หรือ IOT
หรือ Blockchain
หรือ AR
หรือ VR
หรือ MR
หรือ XR
หรือ Hologram
หรือ Metaverse
หรือ Digital Twin
หรือ Smart City
หรือ Smart Home
หรือ Smart Agriculture
หรือ Smart Manufacturing
หรือ Smart Transportation
หรือ Smart Energy
หรือ Smart Healthcare
หรือ Smart Education
หรือ Smart Retail
หรือ Smart Marketing
หรือ Smart Customer Service
หรือ Smart HR
หรือ Smart Finance
หรือ Smart Insurance
หรือ Smart Legal
หรือ Smart Government
หรือ Smart Infrastructure
หรือ Smart Urban Planning
หรือ Smart Environment
หรือ Smart Security
หรือ Smart Defense
หรือ Smart Space
หรือ Smart Exploration
หรือ Smart Colonization
หรือ Smart Interplanetary Travel
หรือ Smart Space Mining
หรือ Smart Space Manufacturing
หรือ Smart Space Agriculture
หรือ Smart Space Tourism
หรือ Smart Space Education
หรือ Smart Space Healthcare
หรือ Smart Space Retail
หรือ Smart Space Marketing
หรือ Smart Space Customer Service
หรือ Smart Space HR
หรือ Smart Space Finance
หรือ Smart Space Insurance
หรือ Smart Space Legal
หรือ Smart Space Government
หรือ Smart Space Infrastructure
หรือ Smart Space Urban Planning
หรือ Smart Space Environment
หรือ Smart Space Security
หรือ Smart Space Defense
หรือ Smart Space Exploration
หรือ Smart Space Colonization
หรือ Smart Space Interplanetary Travel
หรือ Smart Space Mining
หรือ Smart Space Manufacturing
หรือ Smart Space Agriculture
หรือ Smart Space Tourism
หรือ Smart Space Education
หรือ Smart Space Healthcare
หรือ Smart Space Retail
หรือ Smart Space Marketing
หรือ Smart Space Customer Service
หรือ Smart Space HR
หรือ Smart Space Finance
หรือ Smart Space Insurance
หรือ Smart Space Legal
หรือ Smart Space Government
หรือ Smart Space Infrastructure
หรือ Smart Space Urban Planning
หรือ Smart Space Environment
หรือ Smart Space Security
หรือ Smart Space Defense

สรุป		(Testing)	ถูกต้องตามที่กำหนดไว้ โดยเชอร์มทดสอบ ระยะเวลา ในการทดสอบ และค่าใช้จ่ายในการทดสอบที่น้อยที่สุด
	5.	ทักษะการบริหารจัดการการนำ ผลงานเข้าสู่การให้บริการ (Release management)	ทักษะปล่อยซอฟต์แวร์เข้าสู่การให้บริการจริง ต้องวางแผน ร่วมกับผู้ที่เกี่ยวข้องทั้งหมดเพื่อกำหนดขั้นตอนปฏิบัติงาน
	6.	ทักษะการรวบรวมโค้ดที่ได้รับการ พัฒนาจากสมาชิกแต่ละคนในทีม ให้เป็นชิ้นเดียว (Continuous integration)	ทักษะในการรวบรวมโค้ดที่ได้รับการพัฒนาจากสมาชิกแต่ละคนในทีมให้เป็นชิ้นเดียว จากนั้นทำการทดสอบ เพื่อ ตรวจสอบว่าโค้ดแต่ละส่วนเข้ากันได้หรือไม่
	7.	ทักษะในการโค้ดให้ง่ายต่อการเขียน อ่าน และดูแลรักษา (Clean Code)	ทักษะในการโค้ดให้ง่ายต่อการเขียน อ่าน และดูแลรักษา
	8.	ทักษะการออกแบบและการเขียน โปรแกรมเชิงวัตถุ (OO-concepts)	ทักษะในการออกแบบและพัฒนาซอฟต์แวร์เชิงวัตถุ โดยใช้ แนวคิดต่างๆ เช่น การห่อหุ้มส่วนต่าง ๆ (Encapsulation) เพื่อบดบังรายละเอียด (Information Hiding) ส่วนที่ไม่ ต้องการให้เห็นไว้และยังมีแนวคิดอื่น ๆ ที่เกี่ยวข้องเช่น การสืบทอดคลาส (Inheritance) และพอลิมอร์ฟิซึม (Polymorphism) เป็นต้น
	9.	ทักษะการบริหารความเสี่ยง	ทักษะในการวิเคราะห์ ประเมิน จัดการ ติดตาม และ

18.	ทักษะการแบ่งปันข้อมูล (Share knowledge)	ทักษะการนำเสนอและแบ่งปันข้อมูลคือการเผยแพร่ข้อมูล ไปยังผู้สนใจ โดยใช้เทคนิคต่างๆ อย่างเหมาะสม
<p>สอน SPDC เป็นตัวอย่าง แล้วสอนว่าทำไม feed forward / MICE to have.</p>		

	(Belief in simplicity)	ยังต้องเพียงพอกับความต้องการ มุมมองดังกล่าวจะทำให้ งานที่ต้องทำมีจำนวนไม่มากเกินไป
7.	มีความกล้า (Courage)	มีมุมมองที่กล้าที่จะเผชิญหน้ากับปัญหาต่าง ๆ ที่เกิดขึ้น ในช่วงของการพัฒนา ทำให้กิจกรรมสามารถสำเร็จไปได้ รวมถึงกล้าที่จะสื่อสารหรือบอกความผิดพลาดของตนเองที่ เกิดขึ้น

Next Research.

๑. Interview กับผู้ประกอบการ
 ๒. Interview กับ ผู้เชี่ยวชาญ
 ๓. Interview กับ ผู้เชี่ยวชาญ
 ๔. Interview กับ ผู้เชี่ยวชาญ
 ๕. Interview กับ ผู้เชี่ยวชาญ
 ๖. Interview กับ ผู้เชี่ยวชาญ
 ๗. Interview กับ ผู้เชี่ยวชาญ
 ๘. Interview กับ ผู้เชี่ยวชาญ
 ๙. Interview กับ ผู้เชี่ยวชาญ
 ๑๐. Interview กับ ผู้เชี่ยวชาญ

รายงานการเงิน

สัญญาเลขที่ 8/2564

โครงการวิจัยประเภทเงินรายได้ คณะวิทยาการสารสนเทศ

ประจำปีงบประมาณ พ.ศ. 2564

ชื่อโครงการ โครงการศึกษาและรวบรวมทักษะที่จำเป็นสำหรับผู้ปฏิบัติงานในอุตสาหกรรมการพัฒนาซอฟต์แวร์

ชื่อหัวหน้าโครงการวิจัยผู้รับทุน นายอภิสิทธิ์ แสงใส

รายงานในช่วงตั้งแต่วันที่ 30 กรกฎาคม 2564 ถึงวันที่ 20 กรกฎาคม 2565

ระยะเวลาดำเนินการ 1 ปี ตั้งแต่วันที่ 30 กรกฎาคม 2564 ถึงวันที่ 20 กรกฎาคม 2565

รายรับ

จำนวนเงินที่ได้รับ

งวดที่ 1 (50%) 16,500บาท เมื่อ 30 กรกฎาคม 2564

งวดที่ 2 (40%)13,200บาท เมื่อ 7 กุมภาพันธ์ 2565

งวดที่ 3 (10%) 3,300 บาท เมื่อ กรกฎาคม 2565

รวม 33,000 บาท

รายจ่าย

รายการ	งบประมาณที่ตั้งไว้	งบประมาณที่ใช้จริง	จำนวนเงินคงเหลือ
1. ค่าตอบแทน	29,600	29,600	0
2. ค่าใช้สอย	1,750	1,750	0
3. ค่าวัสดุ	1,650	1,650	0
รวมทั้งสิ้น	33,000	33,000	0