

ทักษะที่จำเป็นสำหรับสมาชิกทีมพัฒนาในสกรัม

Necessary Skills for Scrum's Development Team Members

เพ็ญประภา บุตรละ (Penprapa Bootla)* โอปาร โรจนพรพันธุ์ (Olam Rojanapornpun)*
และ พรชัย มงคลนาม (Pornchai Mongkolnam)*

บทคัดย่อ

ลักษณะเด่นของทีมพัฒนาซอฟต์แวร์แบบสกรัม คือ เป็นทีมที่สามารถจัดการตนเองได้ สมาชิกของทีมไม่ได้ถูกกำหนดหน้าที่กันอย่างชัดเจนตายตัว ส่งผลให้สมาชิกทุกคนจำเป็นต้องมีทักษะที่เพียงพอสำหรับการทำงาน จากงานวิจัยที่เกี่ยวข้องพบว่า การขาดทักษะที่จำเป็นส่งผลต่อความสำเร็จในการพัฒนาซอฟต์แวร์แบบสกรัม แต่ในขณะเดียวกัน กรอบการทำงานของสกรัมไม่ได้มีการกำหนดทักษะที่จำเป็นดังกล่าวไว้อย่างแน่นอน มีเพียงแนวทางคร่าวๆ ที่จะนำไปสู่การเป็นทีมที่สามารถจัดการตนเองได้ไว้เท่านั้น วัตถุประสงค์ของงานวิจัยนี้ เพื่อรวบรวมและนำเสนอทักษะที่มีความจำเป็น สำหรับสมาชิกทีมพัฒนาในการพัฒนาซอฟต์แวร์แบบสกรัม เพื่อให้สมาชิกของทีมสามารถเลือกทักษะดังกล่าวไปใช้ในการพัฒนาตนเองได้ โดยทำการรวบรวมทักษะต่างๆ จากงานวิจัยที่เกี่ยวข้อง และสอบถามความคิดเห็นจากผู้เชี่ยวชาญ รวมถึงผู้นำสกรัมไปใช้จริงในประเทศไทย ซึ่งทักษะดังกล่าวถูกจำแนกออกเป็น 3 ประเภท คือ ทักษะทางด้านเทคนิค ทักษะทางด้านบุคคล และทัศนคติ ผลจากการศึกษาพบว่า ทักษะทางด้านบุคคล และทัศนคติส่วนใหญ่ที่นำเสนอในงานวิจัยนี้ ผู้เชี่ยวชาญและผู้ใช้สกรัมเห็นด้วยว่ามีความจำเป็นสำหรับสมาชิกทีมพัฒนาของสกรัม

คำสำคัญ: ทักษะทางด้านเทคนิค ทักษะทางด้านบุคคล ทัศนคติ สกรัม

Abstract

Scrum team is a cross-functional team which is composed of multi-disciplinary individuals, and there are no clearly specified positions. It means that the team members must perform many tasks in both functional area and aptitude area toward a common goal. Team members should have enough skill set in order to carry out their own work and also help the team to meet the goal. Lacking necessary skills can affect the Scrum team's success. This study proposes the necessary skills of the development team, which has been collected from relevant skills found in Scrum sources and subsequently opined by Thai Scrum experts and practitioners. Our skills are categorized into 3 categories: technical skills, people or soft skills, and attitudes. The results show that most of the proposed soft skills and attitudes are necessary for Scrum's development team members.

Keyword: Attitude, People Skills, Scrum, Soft Skills, Technical Skills.

1. บทนำ

สกรัม (Scrum) เป็นหนึ่งในวิธีการพัฒนาซอฟต์แวร์แบบอไจล์ (Agile Software Development) ที่ได้รับความนิยมมากที่สุด ผลการสำรวจในปี ค.ศ. 2010 พบว่ามีผู้ใช้สกรัมในการพัฒนาซอฟต์แวร์ถึง 54% และเพิ่มขึ้นในปี ค.ศ. 2013 เป็น

*ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

55% ใช้สกรัมรวมกับการพัฒนาซอฟต์แวร์แบบอื่น 18% และที่เหลือคือใช้การพัฒนาซอฟต์แวร์แบบอไจล์วิธีอื่นๆ [1-2]

สกรัมเป็นกรอบสำหรับการจัดการโครงการ ทุกกิจกรรมอยู่ภายใต้กรอบเวลา (Time Box) ที่มีการตกลงไว้ร่วมกันระหว่างสมาชิกภายในทีม โดยสกรัมจะเน้นการตรวจสอบและการปรับเปลี่ยนตามสถานการณ์ (Inspection and Adaptation) ให้มีความสำคัญกับเรื่องคุณภาพ และติดตามความคืบหน้างานผ่านซอฟต์แวร์ที่ทำงานได้จริง (Working Software) [3-6] ทีมพัฒนาซอฟต์แวร์แบบสกรัม เป็นทีมที่สามารถจัดการตนเองได้ (Self-Organizing Team) กล่าวคือสมาชิกของทีมรับผิดชอบหน้าที่ของตนเอง และทำงานแบบ “ข้ามฟังก์ชันงาน” (Cross-Functional) ไม่มีการกำหนดหน้าที่หรือตำแหน่งงานภายในทีมกันอย่างชัดเจนตายตัว มีเพียงแค่การกำหนดบทบาทหลักๆ ไว้ 3 บทบาทเท่านั้น คือ ทีมพัฒนา (Development Team) สกรัมมาสเตอร์ (Scrum Master) และเจ้าของผลิตภัณฑ์ (Product Owner) ส่งผลให้สมาชิกภายในทีมต้องทำงานที่หลากหลาย ทั้งที่ตนเองถนัดและไม่ถนัด ดังนั้นสมาชิกของทีมจำเป็นต้องมีทักษะที่เพียงพอเพื่อให้สามารถทำงานตามวัตถุประสงค์ที่วางไว้ร่วมกันได้

ปัจจุบันซอฟต์แวร์มีความซับซ้อนมากขึ้น ในขณะที่ปัจจัยที่ส่งผลต่อการปล่อยซอฟต์แวร์ออกสู่ตลาดก็มีมากขึ้นเช่นกัน อาทิ คุณภาพซอฟต์แวร์การแข่งขันทางการตลาด ความต้องการที่ไม่แน่นอน แรงงานและกรอบเวลาที่มีอย่างจำกัด เป็นต้น ทำให้บางครั้งการพัฒนาซอฟต์แวร์แบบดั้งเดิม อาจไม่สนับสนุนปัจจัยดังกล่าว ในขณะที่สกรัมมีความยืดหยุ่นตามสถานการณ์ ทั้งเรื่องการนำส่ง เวลา ขนาดทีม และมีการตรวจสอบงานที่บ่อยขึ้น [7] ทำให้หลายบริษัทพยายามนำสกรัมไปปรับใช้ในการพัฒนาซอฟต์แวร์ของตน

ผลการวิจัยพบว่า ไม่ใช่ทุกทีมที่นำสกรัมไปใช้ในการพัฒนาซอฟต์แวร์แล้วจะประสบความสำเร็จได้ ปัจจัยหนึ่งส่งผลให้ทีมล้มเหลว คือ ปัจจัยทางด้านบุคคล เช่น สมาชิกของทีมขาดการสื่อสารที่มีประสิทธิภาพ ขาดทักษะที่จำเป็น รวมถึงขาดความรู้ทางด้านเทคโนโลยีและธุรกิจนั้นๆ [8] Dingsoyr และ Dyba ทำการศึกษาทีมที่สามารถจัดการตนเองได้ในสกรัม พวกเขาได้อธิบายว่า “ทักษะที่เกี่ยวข้องเป็นสิ่งจำเป็นต่อการพัฒนาซอฟต์แวร์แบบสกรัม และยังเป็นกุญแจสำคัญที่จะนำทีมไปสู่การเป็นทีมที่สามารถจัดการตนเองได้” [9] แต่เนื่องจากกรอบการทำงานของสกรัม ไม่ได้

มีการกำหนดทักษะที่จำเป็นไว้อย่างชัดเจน งานวิจัยนี้จึงทำการศึกษา รวบรวมและนำเสนอทักษะที่มีความจำเป็นสำหรับสมาชิกทีมพัฒนาเท่านั้น ไม่ได้รวมถึงสกรัมมาสเตอร์และเจ้าของผลิตภัณฑ์ โดยทักษะที่นำเสนอดังกล่าว ถูกนำไปสอบถามความคิดเห็นจากผู้เชี่ยวชาญ และผู้นำสกรัมไปใช้จริงในประเทศไทย

2. งานวิจัยที่เกี่ยวข้อง

หัวใจของการพัฒนาซอฟต์แวร์แบบสกรัม คือ การเป็นทีมที่สามารถจัดการตนเองได้ โดยแนวทางในการไปสู่เป้าหมายดังกล่าวได้แฝงอยู่ในหลักปฏิบัติของอไจล์ 12 ข้อ อาทิ เช่น ความถี่และต่อเนื่องในการนำส่งซอฟต์แวร์ที่ทำงานได้จริง การยินดีและพร้อมตอบรับการเปลี่ยนแปลงความสามารถในการจัดการงาน การสื่อสารที่มีประสิทธิภาพ การมีแรงผลักดันในการทำงาน ซึ่งสิ่งเหล่านี้สามารถที่จะสนับสนุนทีมให้ประสบความสำเร็จได้ ทีมพัฒนาซอฟต์แวร์แบบอไจล์ไม่ว่าจะเป็นการพัฒนาด้วยวิธีการใด สมาชิกของทีมต้องมีจุดมุ่งหมายเดียวกัน มีความไว้วางใจกัน เคารพข้อสัตย์ต่อกัน รวมถึงมีความสามารถที่จะทำงานที่ตนได้รับมอบหมายได้ในทุกสถานการณ์ที่เปลี่ยนแปลงไป [10-14]

ในสกรัมไม่เคยมีการกำหนดทักษะที่จำเป็นขึ้นมาอย่างเป็นทางการ ดังที่ Moe และคณะ [15] ได้ชี้ว่า “สกรัมเน้นการเป็นทีมที่สามารถจัดการตนเองได้ แต่ก็ไม่ได้กำหนดแนวทางหรือทักษะที่จำเป็นไว้อย่างชัดเจน” แต่จากการศึกษาพบว่า มีบางงานวิจัยที่ได้กล่าวถึงทักษะที่ใช้ในสกรัมให้เห็นอยู่บ้าง

ทักษะหลักที่มีความสำคัญในการพัฒนาซอฟต์แวร์แบบสกรัม คือ ทักษะการสื่อสาร (Communication) สกรัมเน้นการทำงานในสถานที่เดียวกัน เพื่อให้ได้การสื่อสารที่รวดเร็ว หากเกิดข้อสงสัยหรือปัญหาจะได้แก้ได้ทันทีที่ การสื่อสารที่ไม่มีประสิทธิภาพและประสิทธิผล อาจทำให้เกิดการทำงานที่ซ้ำซ้อน และอาจเกิดความผิดพลาดขึ้นมาได้ [16-18]

ในปี ค.ศ. 2010 Prabhakaran [19] ได้เขียนบทความในนิตยสาร “Agile Record, the Magazine for Agile Developers and Agile Testers” เกี่ยวกับทักษะของสกรัมทีม โดยมีการแบ่งทักษะที่เกี่ยวข้องออกเป็น 2 ประเภท คือ ทักษะทางเทคนิค (Technical Skills) และทักษะที่เห็นได้ทางด้านพฤติกรรม (Behavioural Skills) สำหรับนักพัฒนา (Developer)

นักควบคุมคุณภาพ (Quality Assurance: QA) และสกรัมมาสเตอร์ (Scrum Master) ขอสังเกตพบว่า Prabhakaran มีการแยกนักควบคุมคุณภาพออกมาเป็นอีก 1 บทบาท ซึ่งการแยกดังกล่าวขึ้นอยู่กับทีมพัฒนานั้นๆ ด้วย โดยที่ทักษะส่วนใหญ่ของนักพัฒนาและนักควบคุมคุณภาพไม่ได้มีความแตกต่างกัน ในขณะที่สกรัมมาสเตอร์ต้องมีทักษะการอำนวยความสะดวกให้แก่ทีมเพิ่มขึ้น แต่เนื่องจากบางทักษะ ไม่ได้มีการอธิบายความหมายเอาไว้ในบทความ อาจส่งผลให้เกิดความสับสน หรือเกิดความเข้าใจที่ไม่ตรงกันได้

เนื่องจากงานวิจัยที่กล่าวถึงทักษะของสกรัมโดยตรงมีอยู่ค่อนข้างน้อย ผู้วิจัยจึงได้พิจารณาทักษะที่นิยมใช้ในการพัฒนาซอฟต์แวร์แบบอไจล์วิธีอื่นๆ มาร่วมในการนำเสนอทักษะที่จำเป็นสำหรับทีมพัฒนาซอฟต์แวร์แบบสกรัม เพราะเชื่อว่า ทั้งหมดตั้งอยู่บนหลักปฏิบัติ (Principles) และการให้คุณค่า (Value) ร่วมกัน

ในปี ค.ศ. 2012 Lorber และ Tieszen [20] ได้ทำการศึกษาพัฒนาวิธีในการสร้างความสมดุลงานก่อนที่จะเข้าพัฒนาในแต่ละสปรินท์ งานวิจัยดังกล่าวให้ความสำคัญกับการนำส่งซอฟต์แวร์ให้ตรงตามความต้องการ และทันเวลาในแต่ละสปรินท์ ซึ่งพวกเขาได้อธิบายว่า “ทีมที่ดีควรมีความพยายามที่จะปรับปรุงความสามารถ ในการทำงานแบบข้ามหน้าที่การทำงาน โดยใช้หลักปฏิบัติที่ดี เช่น แพร์โปรแกรมมิ่ง (Pair Programming) เพียร์โคดรีวิว (Peer Code Review) และทำงานอยู่ในสถานที่เดียวกัน ควรมีทักษะด้านการประมาณการ (Effort Estimation) เพื่อประเมินทรัพยากรที่ต้องใช้ในการพัฒนา อีกทั้งยังต้องมีความสามารถในการเจรจาต่อรอง (Negotiation) กรณีที่ต้องต่อรองเพื่อลดขอบเขตงานกับเจ้าของผลิตภัณฑ์อีกด้วย”

Kovitz [21] นำเสนอทักษะสำหรับการพัฒนาความต้องการของลูกค้า (Requirement Engineering Phase) ในการพัฒนาซอฟต์แวร์แบบอไจล์ ซึ่งทักษะที่ถูกกล่าวถึง เช่น ทักษะการแตกงานให้เป็นงานย่อยๆ (Breaking Big Thing into Tiny Things) เพื่อให้ทีมสามารถเลือกไปใช้ในการพัฒนาได้ ทักษะการมีส่วนร่วมและการสื่อสาร (Collaboration and Communication) ทักษะการเขียนโค้ดแบบใช้การทดสอบในการขับเคลื่อน (Test-Driven) และทักษะการออกแบบเชิงวัตถุ (Object Oriented Design) เป็นต้น โดยมีงานของ Viljakainen

และ Turunen [22] กล่าวสนับสนุนทักษะดังกล่าวด้วยเช่นกัน

Wirfs-Brock [23], [24] ทำการศึกษาเกี่ยวกับทักษะสำหรับนักออกแบบในการพัฒนาซอฟต์แวร์แบบอไจล์ (Agile Designer) ซึ่งใช้สกรัมเป็นตัวอย่างของการศึกษา โดยอธิบายว่า การออกแบบจะถูกทำในสปรินท์ ซึ่งสกรัมต้องการทราบปัญหาและหาแนวทางแก้ไขให้ได้ภายในเวลาอันรวดเร็ว ฉะนั้นการออกแบบจะอย่างไรพอเพียงเท่านั้น เน้นไปที่ความเรียบง่ายแต่สามารถนำไปแก้ปัญหาได้ หากในกรณีที่แนวทางดังกล่าวไม่สามารถตอบโจทย์ของลูกค้า การออกแบบจะถูกทำซ้ำ โดยมีการร่วมมือและสื่อสารกันภายในทีม (Collaboration and Communication) ทุกคนต้องกล้าที่จะเปิดใจ มีความจริงใจต่อกัน รวมถึงการมีทัศนคติที่ดี ซึ่งถือว่าเป็นสิ่งสำคัญเหนือกว่าเทคนิคและกระบวนการใดๆ อันจะส่งผลให้การทำงานประสบความสำเร็จได้

Tan และ Teo [25] ทำการทดลองสอนนักศึกษาเกี่ยวกับทักษะการพัฒนาซอฟต์แวร์แบบอไจล์ และมีการเปรียบเทียบผลลัพธ์ก่อนและหลังการสอน ผลพบว่า หลังจากที่นักศึกษาได้เรียนรู้ทักษะดังกล่าว ส่งผลทางด้านบวกกับคุณภาพของซอฟต์แวร์ (Software Quality) รวมถึงความคืบหน้าของการพัฒนาโครงการด้วย (Work Progress) โดยทักษะที่ถูกนำเสนอ เช่น ทักษะการรับมือกับการเปลี่ยนแปลง (Dealing with Changes) การจัดการความขัดแย้ง (Conflict Management) รวมไปถึงทักษะที่สะท้อนออกมาจากบทบาทหน้าที่ของสมาชิกภายในทีม เป็นต้น

3. วิธีการดำเนินการวิจัย

3.1 การรวบรวมทักษะที่เกี่ยวข้อง

จากการศึกษาข้างต้น ผู้วิจัยได้ทำการรวบรวมทักษะทั้งที่ได้จากงานวิจัยที่เกี่ยวข้องโดยตรง และเกิดจากการพิจารณาลักษณะของสกรัม อาทิ เช่น กิจกรรมที่เกี่ยวข้อง ถ้อยแถลง อไจล์ (Agile Manifesto) หลักปฏิบัติของอไจล์ (Agile Principle) เป็นต้น [26-27] รวมถึงมีการพิจารณาทักษะในการพัฒนาซอฟต์แวร์ และทักษะทางด้านบุคคลทั่วไป ที่น่าจะมีประโยชน์กับทีมพัฒนาในสกรัมร่วมด้วย จากนั้นทำการจำแนกทักษะดังกล่าวออกเป็น 3 ประเภท คือ (1) ทักษะทางด้านเทคนิค (Technical Skills) (2) ทักษะทางด้านบุคคล (People/ Soft Skills) และ (3) ทัศนคติ (Attitude) ดังตารางที่ 1 และตารางที่ 2

3.2 การประเมินความคิดเห็นของผู้เชี่ยวชาญและผู้ใช้สกรัม

ผู้วิจัยได้ทำแบบสอบถาม เพื่อประเมินความคิดเห็นเกี่ยวกับทักษะที่นำเสนอ ว่ามีความจำเป็นกับทีมพัฒนาในสกรัมมากน้อยเพียงใด โดยผู้ตอบแบบสอบถามสามารถให้ความคิดเห็น เกี่ยวกับทักษะนั้นๆ และยังสามารถนำเสนอทักษะอื่นๆ ที่เกี่ยวข้องได้ ซึ่งกลุ่มตัวอย่างในการสำรวจครั้งนี้คือ ผู้เชี่ยวชาญทางด้านสกรัมในประเทศไทย จำนวน 7 ท่าน ซึ่งเป็นผู้มีส่วนผลักดันให้การพัฒนาซอฟต์แวร์แบบอไจล์เป็นที่รู้จักกันมากขึ้นในประเทศไทย โดยที่ผู้วิจัยใช้วิธีการสุ่มแบบเฉพาะเจาะจง (Purposive Sampling) ในการเลือกผู้เชี่ยวชาญ และมีผู้ตอบแบบสอบถามอีกจำนวน 31 คน ที่เป็นผู้ใช้สกรัมจริงในการพัฒนาซอฟต์แวร์ในประเทศไทย (Scrum Practitioners) ซึ่งแบบสอบถามประกอบ

ไปด้วย (1) ข้อมูลพื้นฐานของผู้ตอบแบบสอบถาม (2) ความคิดเห็นเกี่ยวกับทักษะและทัศนคติที่จำเป็นต่อความสำเร็จของทีม สำหรับสมาชิกทีมพัฒนาของสกรัม (3) ข้อเสนอแนะ

โดยแบบสอบถามในส่วนที่ 2 ผู้วิจัยได้นำมาตรวัดของลิเคิร์ต (Likert Scale) มาใช้เป็นระดับคะแนนในการประเมินความคิดเห็น ซึ่งมีค่าดังต่อไปนี้ 5 = เห็นด้วยอย่างยิ่ง 4 = เห็นด้วย 3 = ไม่มั่นใจ 2 = ไม่เห็นด้วย และ 1 = ไม่เห็นด้วยอย่างยิ่ง

เนื่องจากทักษะส่วนใหญ่ที่นำเสนอ เป็นทักษะทางด้านการพัฒนาซอฟต์แวร์และทักษะทางด้านบุคคลต่างๆ ไปที่นักพัฒนาซอฟต์แวร์ส่วนใหญ่ทราบกันดีอยู่แล้ว ผู้วิจัยจึงขออธิบายเพิ่มเติมเฉพาะในส่วนทักษะที่เกี่ยวข้องโดยตรงกับการพัฒนาซอฟต์แวร์แบบสกรัม ดังนี้

ตารางที่ 1 ทักษะทางด้านเทคนิค และทักษะทางด้านบุคคลที่ผู้วิจัยนำเสนอ

ทักษะทางด้านเทคนิค	ทักษะทางด้านบุคคล
<p>ทักษะที่เกี่ยวข้องกับสกรัมโดยตรง (Scrum Skills)</p> <ul style="list-style-type: none"> - ทักษะการทำการระบวนการสกรัม (Scrum Process) - ทักษะการแตกงาน (Breaking the Product Backlog) - ทักษะการประมาณการแรงงาน (Effort Estimation) - ทักษะการลำดับความสำคัญของงาน (Advance Prioritization) <p>ทักษะที่ได้รับความนิยมจากการพัฒนาซอฟต์แวร์แบบอื่น ๆ (Popular Skills)</p> <ul style="list-style-type: none"> - ทักษะการพัฒนาซอฟต์แวร์แบบใช้การทดสอบเป็นตัวขับเคลื่อน (Test Driven Practice) - ทักษะการทำแพร์โปรแกรมมิ่ง (Pair Programming) <p>ทักษะเกี่ยวกับการพัฒนาซอฟต์แวร์ทั่วไป (Related Software Development Skills)</p> <ul style="list-style-type: none"> - ทักษะการคิดเชิงประมวลผล (Computational Thinking) - ทักษะการออกแบบโปรแกรมเชิงวัตถุ (Object Oriented Design : OOD) - ทักษะการใช้ดีไซน์แพทเทิร์น (Design Pattern) - ทักษะการเตรียมสภาพแวดล้อมที่เหมาะสมในการพัฒนาซอฟต์แวร์ (Environment Preparation) - ทักษะการเขียนโค้ดตามมาตรฐาน (Coding Standard) - ทักษะการตรวจสอบโค้ด (Code Review) - ทักษะการหากรณีการทดสอบให้ครอบคลุม (Code Coverage) - ทักษะการทดสอบหน่วยย่อย (Unit Testing) - ทักษะการระบุโค้ดที่น่าจะนำไปสู่ปัญหา (Code Smelling) - ทักษะการทำรีแฟคเตอร์ (Refactoring) - ทักษะการตรวจสอบความผิดพลาด (Debugging) 	<ul style="list-style-type: none"> - ทักษะในการเรียนรู้สิ่งใหม่ๆ (Active Learning) - ทักษะการคิดอย่างสร้างสรรค์ (Creative Thinking) - ทักษะในการตัดสินใจ/การคิดแบบมีวิจารณญาณ (Decision Making/ Critical Thinking) - ทักษะการแก้ปัญหา (Problem Solving) - ทักษะการจัดการเวลา/การวางแผน (Time Management/ Planning) - ทักษะการจัดการกับการเปลี่ยนแปลง (Dealing with Change) - ทักษะการจัดการความเครียด (Stress Handling) - ทักษะการสื่อสาร (Communication) - ทักษะความเป็นผู้นำ (Leadership) - ทักษะการทำงานเป็นทีม/การสร้างทีม (Teamwork/ Team Building) - ทักษะการสร้างสัมพันธภาพและความเข้าใจกัน (Diplomacy) - ทักษะการจัดการความขัดแย้ง (Conflict Management) - ทักษะการเจรจาต่อรอง (Negotiation) - ทักษะในการนำเสนองาน (Presentation)

ทักษะการทำการกระบวนการสกรัม คือ ทักษะในการทำกิจกรรมต่างๆ ในกระบวนการของการพัฒนาซอฟต์แวร์แบบสกรัม (Scrum) ต้องมีความรู้ความเข้าใจในกระบวนการพัฒนา ไม่ว่าจะเป็นบทบาท (Scrum Role) กิจกรรม (Scrum Activity) งานหรือผลจากกิจกรรมที่กระทำ (Artifact) หลักปฏิบัติสำคัญที่เกี่ยวข้อง (Practice) รวมถึงความรู้อื่นๆ ที่เกี่ยวข้องตั้งแต่เริ่มต้นจนถึงสิ้นสุดการพัฒนาแบบ

ทักษะการลำดับความสำคัญของงาน คือ ทักษะในการจัดลำดับความสำคัญของงานที่ทำ (Task) รู้ว่างานอะไรที่ต้องทำก่อน-ทำหลัง และงานใดต้องรอให้งานอื่นเสร็จเรียบร้อยก่อน ถึงจะสามารถทำงานนั้นได้

ทักษะการประเมินแรงงาน คือ ทักษะในการประมาณการแรงงาน (Effort) ที่ใช้ในการพัฒนา การพัฒนาซอฟต์แวร์แบบ อัจฉริยะกำหนดให้ทีมแสดงความคิดเห็น และตัดสินใจร่วมกันในการประมาณการแรงงานของแต่ละโปรดักแบคล็อก (Product Backlog) หากมีสมาชิกคนใดมีความเห็นที่แตกต่างสมาชิกคนอื่นๆ ต้องอธิบายและแสดงความคิดเห็น จนกว่าสมาชิกคนดังกล่าว จะเห็นด้วยกับการประมาณการนั้นๆ

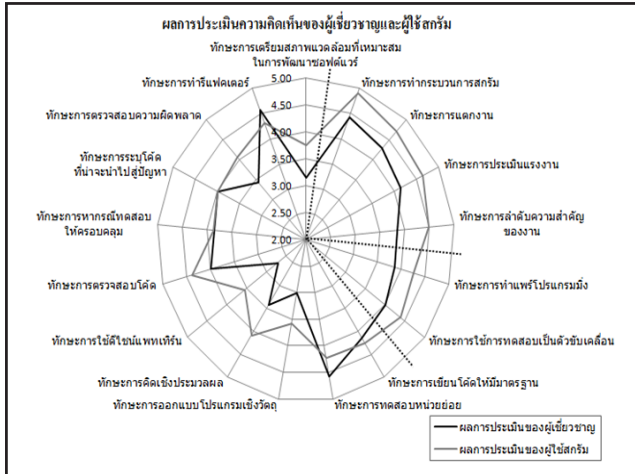
ทักษะการแตกงาน คือ ทักษะในการคิดวิเคราะห์ แยกโปรดักแบคล็อกให้เป็นงานย่อยๆ เพื่อให้สมาชิกภายในทีมนำไปพัฒนาหรือแก้ปัญหาที่ละส่วนได้

ตารางที่ 2 ทศนคติที่ผู้วิจัยนำเสนอ

ทักษะทางด้านเทคนิค	ทักษะทางด้านบุคคล
ให้ความสำคัญกับซอฟต์แวร์ที่ทำงานได้จริง (Focus on Working Software)	การมีมุมมองหรือการตระหนักว่า ซอฟต์แวร์ที่ทำงานได้จริงมีความสำคัญมากกว่าเอกสารหรือไอเดียแกรมที่ต้องนำส่งลูกค้า และสามารถนำมาใช้ในการวัดความก้าวหน้าของงานได้
ตระหนักในเรื่องคุณภาพ (Focus on Quality)	การมีมุมมองหรือการตระหนักถึงเรื่องคุณภาพอยู่ตลอดเวลา โดยเชื่อว่าการพัฒนาซอฟต์แวร์ให้มีคุณภาพตลอดเวลาและทุกขั้นตอน จะช่วยทำให้ซอฟต์แวร์มีข้อผิดพลาดน้อยลง
เชื่อในความเรียบง่าย (Belief in Simplicity)	การมีมุมมองในการทำสิ่งต่างๆ ให้ง่ายและทำเท่าที่จำเป็น แต่ยังคงเพียงพอกับความต้องการ มุมมองดังกล่าวจะทำให้งานที่ต้องทำมีจำนวนไม่มากเกินไป
เชื่อในการเป็นเจ้าของโค้ดร่วมกัน (Belief in Collective Code Ownership)	การมีมุมมองที่ว่า “โค้ดหรือผลิตภัณฑ์เป็นของทุกคนภายในทีม” ทุกคนเป็นเจ้าของร่วมกัน ดังนั้นในกรณีที่เกิดปัญหาขึ้น และผู้ที่มีหน้าที่รับผิดชอบหลักไม่อยู่ สามารถให้คนอื่น ๆ ภายในทีมทำการแก้ไข และทำงานต่อไปได้
ความเคารพและไว้วางใจซึ่งกันและกัน (Respect and Trust)	การมีความไว้วางใจ และการเคารพซึ่งกันและกัน เชื่อว่าสมาชิกของทีมแต่ละคนมีความรู้ และความสามารถเพียงพอที่จะทำงานให้สำเร็จลุล่วงลงได้ เพราะสมาชิกของทีมมีความหลากหลาย ทั้งในแง่ของประสบการณ์พื้นฐานและวัฒนธรรม
ความกล้า (Courage)	การมีมุมมองที่กล้าที่จะเผชิญหน้ากับปัญหาต่างๆ ที่เกิดขึ้นในช่วงของการพัฒนา ทำให้กิจกรรมสามารถลุล่วงไปได้ รวมถึงกล้าที่จะสื่อสารหรือบอกความผิดพลาดของตนเองที่เกิดขึ้น สมาชิกของทีมจะได้หารือร่วมกันในการแก้ปัญหาดังกล่าว
ให้ความสำคัญกับกรอบเวลาที่กำหนด (Focus on Time Boxing)	การมีมุมมองในเรื่องการเคารพกรอบเวลา (Time Boxing) ที่ได้กำหนดไว้ร่วมกัน โดยเมื่อถึงเวลาที่กำหนดดังกล่าว ไม่ว่าจะงานจะเสร็จตามกำหนดเวลาหรือไม่ จำเป็นต้องหยุดทันที เพื่อทำกิจกรรมถัดไป
เชื่อในทีมที่สามารถจัดการตนเองได้ (Belief in Self-Directing Team)	การมีมุมมองที่ว่า หากสมาชิกทุกคนภายในทีมรับบทบาทหน้าที่ของตน สามารถตัดสินใจได้ด้วยตนเอง รู้ว่าต้องทำอะไร ตอนไหนต้องนำหรือตอนไหนต้องตาม จะทำให้เกิดเป็นทีมที่สามารถจัดการตนเองได้
มีมุมมองในการทำบูรณาการโปรแกรมอย่างต่อเนื่อง (Continuous Integration)	การมีมุมมองในการบูรณาการ (Integration) โปรแกรมที่ได้พัฒนาขึ้นอย่างต่อเนื่องและบ่อยครั้ง รวมถึงเล็งเห็นความสำคัญของการทำบูรณาการบ่อยๆ จะทำให้ได้ทดสอบ และเห็นซอฟต์แวร์ที่ทำงานได้จริงได้เร็วขึ้น

4. ผลการวิจัย

การตีความคะแนนการประเมินผล แบ่งออกเป็น 5 ระดับ คือ 4.51–5.00 หมายถึง เห็นด้วยอย่างยิ่ง 3.51–4.50 หมายถึง เห็นด้วย 2.51–3.50 หมายถึง ไม่มั่นใจ 1.51–2.50 หมายถึง ไม่เห็นด้วย และ 1.00–1.50 หมายถึง ไม่เห็นด้วยอย่างยิ่ง ผลการประเมินเป็นดังภาพที่ 1 ภาพที่ 2 และภาพที่ 3



ภาพที่ 1 ผลการประเมินเกี่ยวกับทักษะทางด้านเทคนิค

จากภาพที่ 1 จะเห็นได้ว่ามีการใช้เส้นประแบ่งทักษะทางด้านเทคนิคออกเป็น 3 กลุ่ม ได้แก่

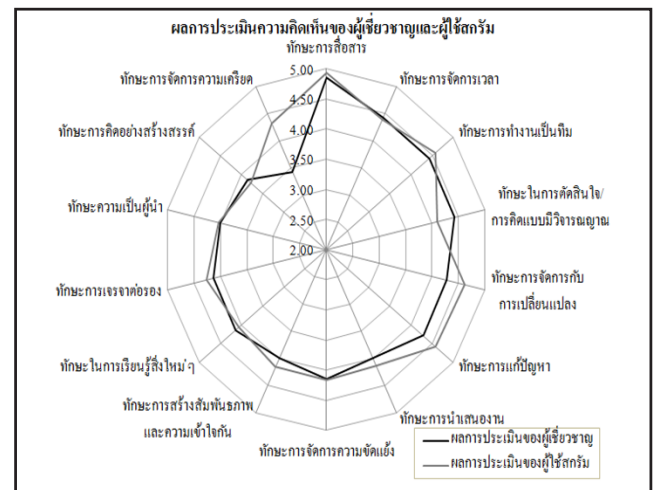
กลุ่มที่ 1 ทักษะที่เกี่ยวข้องกับสกริมโดยตรง ผู้ประเมินทั้งผู้เชี่ยวชาญและผู้ใช้สกริม เห็นด้วยว่ามีความจำเป็นสำหรับสมาชิกทีมพัฒนาในสกริม ในขณะที่มีผู้เชี่ยวชาญท่านหนึ่งได้ให้ความเห็นเพิ่มเติมว่า “ปกติเจ้าของผลิตภัณฑ์เป็นผู้จัดลำดับโปรดัคแบคคี่ลิกตามความต้องการ ในขณะที่เดียวกันทีมพัฒนาต้องสามารถแตกโปรดัคแบคคี่ลิกนั้นๆ ออกเป็นงานย่อยๆ ให้ได้ พร้อมทั้งกำหนดความสำคัญ รู้ว่างานใดมาก่อนมาหลัง ก่อนที่จะนำไปพัฒนาจริง”

กลุ่มที่ 2 ทักษะที่ได้รับความนิยมจากการพัฒนาซอฟต์แวร์แบบอื่นๆ กล่าวคือ ไม่ได้เกี่ยวข้องกับสกริมโดยตรง แต่ถูกกล่าวถึงว่าเป็นกระบวนการที่ดี ที่มีการนำมาประยุกต์ใช้ในสกริม ทักษะกลุ่มนี้ผู้ประเมินเห็นด้วยว่ามีความจำเป็นกับทีมพัฒนาในสกริม และผู้เชี่ยวชาญได้อธิบายว่า “ในสกริมเน้นการทำงานเป็นทีม การใช้แพร์โปรแกรมมิ่ง จะช่วยในการตรวจสอบโค้ด และเป็นการแชร์ความรู้ร่วมกันระหว่างสมาชิกของทีม อีกทั้งการใช้ Test Driven Practice ยังเป็นการช่วยให้โค้ดได้ถูกตรวจสอบได้บ่อยครั้งอีกด้วย”

กลุ่มที่ 3 ทักษะที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ทั่วไป

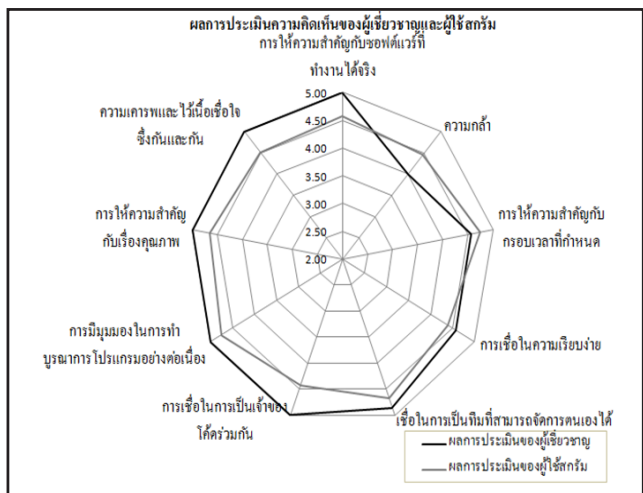
(Related Software Development Skills) โดยทักษะกลุ่มนี้ผู้ใช้สกริมเห็นด้วยว่า มีความจำเป็นกับสมาชิกทีมพัฒนาในสกริม และยังมีความสำคัญกับการพัฒนาซอฟต์แวร์แบบอื่นๆ เพราะเป็นทักษะพื้นฐานของการพัฒนาซอฟต์แวร์ ในขณะที่ผู้เชี่ยวชาญทางด้านสกริมเห็นว่า ทักษะการใช้ดีไซน์แพทเทิร์น และทักษะการออกแบบโปรแกรมเชิงวัตถุ ไม่จำเป็นสำหรับสมาชิกทีมสกริม เพราะขึ้นอยู่กับการตกลงของทีม อาทิ เช่น จะออกแบบอย่างไร เป็นโปรแกรมเชิงวัตถุหรือไม่ จะมีการใช้ดีไซน์แพทเทิร์นเข้าช่วยในการพัฒนาครั้งนั้นหรือไม่ เป็นต้น

ในทักษะกลุ่มที่ 3 นี้ผู้เชี่ยวชาญยังไม่มั่นใจว่า ทักษะการเตรียมสภาพแวดล้อมที่เหมาะสมในการพัฒนาซอฟต์แวร์ ทักษะการตรวจสอบความผิดพลาด และทักษะการคิดเชิงประมวลผล มีความจำเป็นสำหรับทีมพัฒนาในสกริม เพราะบางครั้งการเตรียมสภาพแวดล้อมในการพัฒนาซอฟต์แวร์ ถูกทำโดยเจ้าหน้าที่สนับสนุนไอที (IT Support) ไม่ใช่ทีมพัฒนาของสกริม ส่วนทักษะการตรวจสอบความผิดพลาด ผู้เชี่ยวชาญได้อธิบายว่า ถือว่าเป็นส่วนหนึ่งของทักษะการใช้การทดสอบเป็นตัวขับเคลื่อนแล้ว ไม่จำเป็นต้องแยกทักษะดังกล่าวออกมาต่างหาก



ภาพที่ 2 ผลการประเมินเกี่ยวกับทักษะทางด้านบุคคล

จากภาพที่ 2 ผู้ใช้สกริมเห็นด้วยว่าทักษะทางด้านบุคคลที่นำเสนอ จำเป็นกับทีมพัฒนาในสกริม และมีเพียงทักษะการจัดการความเครียดเท่านั้น ที่ผู้เชี่ยวชาญไม่เห็นด้วยว่าจำเป็นกับทีมพัฒนาในสกริม เนื่องจากความเครียดส่วนใหญ่เกิดกับโครงการที่มีเวลาจำกัด ขอบเขตของงานขนาดใหญ่ แต่สกริมมีการเลือกจำนวนงานตามความสามารถของทีม ที่



ภาพที่ 3 ผลการประเมินเกี่ยวกับทัศนคติ

จะนำมาพัฒนาในรอบสปริ้นท์นั้น ๆ รวมถึงสามารถที่จะต่อตรงกับเจ้าของผลิตภัณฑ์กรณีที่ต้องการลดขอบเขตของงาน ส่งผลให้ความเครียดที่เกิดขึ้นมีค่อนข้างน้อย

จากภาพที่ 3 ผู้เชี่ยวชาญและผู้ใช้สกรัมต่างเห็นด้วยว่าทัศนคติทั้งหมดที่นำเสนอจำเป็นสำหรับสมาชิกทีมพัฒนาของสกรัม และมีผู้เชี่ยวชาญท่านหนึ่งได้อธิบายเพิ่มเติมเกี่ยวกับ “ความกล้า” ว่าบ่อยครั้งที่ต้องใช้ความกล้าในการตัดสินใจ หากทีมไม่มีความเชื่อมั่นและเชื่อใจกัน ความกล้าดังกล่าวอาจถูกตีความมาเป็นการบ่นของคนที่มีทักษะดังกล่าวได้

จากการวิเคราะห์ข้อมูล ทักษะทางด้านเทคนิคที่จำเป็นสำหรับทีมพัฒนาซอฟต์แวร์แบบสกรัม ไม่ได้แตกต่างจากการพัฒนาซอฟต์แวร์แบบดั้งเดิมมากนัก จะมีบ้างตามลักษณะ เฉพาะของกิจกรรมของสกรัม ในขณะที่ทักษะทางด้านบุคคลและทัศนคติ ดูเหมือนจะเป็นสิ่งที่สกรัมมุ่งเน้นให้สมาชิกในทีม ต้องมีเพิ่มขึ้น มากกว่าที่จะมองทักษะทางด้านเทคนิคเป็นหลักเพียงด้านเดียว

5. สรุปผล

เนื่องจากปัจจัยทางด้านบุคคลอันหมายถึงสมาชิกของทีมได้กลายมาเป็นกุญแจสำคัญ ที่ส่งผลต่อความสำเร็จของการพัฒนาซอฟต์แวร์แบบสกรัม การที่สมาชิกของทีมมีความรู้ ความสามารถ รวมถึงการมีทักษะที่เพียงพอ จะช่วยนำทีมไปสู่เป้าหมายที่วางไว้ร่วมกันได้ งานวิจัยนี้เสนอทักษะที่จำเป็นสำหรับสมาชิกทีมพัฒนาของสกรัม ผู้วิจัยได้ทำการรวบรวมทักษะที่เกี่ยวข้อง และทำการประเมินความคิดเห็นจากผู้เชี่ยวชาญด้านสกรัมและผู้ที่นำสกรัมไปใช้

พัฒนาซอฟต์แวร์จริงในประเทศไทย โดยแบ่งทักษะดังกล่าวออกเป็น 3 ประเภท คือ ทักษะทางด้านเทคนิค ที่ส่งผลโดยตรงต่อการพัฒนาซอฟต์แวร์, ทักษะทางด้านบุคคล เป็นทักษะที่ช่วยในการสนับสนุนให้การพัฒนาซอฟต์แวร์มีประสิทธิภาพมากยิ่งขึ้น รวมถึงสร้างความสัมพันธ์ระหว่างคนในทีม, และทัศนคติ ซึ่งเป็นมุมมองที่ดีที่สมาชิกของทีมควรมี ปัจจุบันมีงานวิจัยไม่มากนักที่ทำการศึกษาเกี่ยวกับทักษะที่เกี่ยวข้องในการพัฒนาซอฟต์แวร์แบบดังกล่าว งานวิจัยนี้ช่วยให้ทราบถึงทักษะที่จำเป็นสำหรับสมาชิกทีมพัฒนา ในการพัฒนาซอฟต์แวร์แบบสกรัม ซึ่งสมาชิกของทีมสามารถเลือกทักษะที่ยังไม่เชี่ยวชาญ มาใช้ในการพัฒนาตนเองได้ ส่วนงานที่สามารถทำได้ในอนาคต คือ การศึกษาหาวิธีการวัดระดับความสามารถในแต่ละทักษะดังกล่าว เพื่อให้สมาชิกของทีมรู้ว่า ณ ปัจจุบัน ตนเองมีความสามารถในระดับใด และต้องการที่จะก้าวไปสู่ระดับใดในอนาคต

6. เอกสารอ้างอิง

- [1] G. Azizyan, M. N. Magarian, and M. Kajko-Mattson. “Survey of Agile Tool Usage and Needs.” *In Proceedings of the Agile Conference (AGILE)*, pp. 29-38, August 7-13, 2011.
- [2] “State of Agile Survey.” Available Online at <http://stateofagile.versionone.com>, accessed on Jul. 20, 2014.
- [3] “The Scrum Primer Version 2.0” Available Online at http://www.scrumprimer.org/scrupprimer20_small.pdf, accessed on Oct. 20, 2014.
- [4] K. S. Rubin. *Essential Scrum: a practical guide to the most popular agile process*. First printing, Pearson Education, Inc., USA, pp. 198- 211, 2012.
- [5] S. Viscardi. *The Professional Scrum Master’s Handbook*. First printing, Packt Publishing Ltd., 35 Livery Street Birmingham B3 2PB, UK, p. 99, 2013.
- [6] K. Schwaber. “Scrum Guide.” Available Online at <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>, accessed on Jul. 15, 2014
- [7] K. Schwaber. *Scrum Development Process*. Springer London, pp. 117-134, 1997.

- [8] N. Chantachaimongkol and P. Sincharoenpanich. *Critical factors for implementing the Scrum software development methodology*. Master Thesis in IT Management, School of Business, Society, and Engineering, Malardalen University, 2013.
- [9] N. B. Moe, T. Dingsoyr, and T. Dyba. "Understanding Self-organizing Teams in Agile Software Development." *In Proceedings of the Australian Conference on Software Engineering*, pp. 76-85, March 26- 28, 2008.
- [10] M. Fowler and J. Highsmith. "The agile manifesto." *Software Development*, August, 2001.
- [11] A. Cockburn and J. Highsmith. "Agile Software Development: The People Factor," *Computer*, Vol. 34, Issue 11, pp. 131-133, 2001.
- [12] R. Hoda, J. Noble and S. Marshall. "Organizing self-organizing teams." *In Proceedings of the ACM/IEEE 32nd International Conference on Software Engineering*, Vol. 1, pp. 285-294, 2010.
- [13] R. Hoda, J. Noble and S. Marshall. "Self-Organizing Roles on Agile Software Development Teams." *The IEEE Transactions on Software Engineering*, Vol. 39, Issue 3, pp. 422-444, 2013.
- [14] T. Chow and D. Cao. "A Survey Study of Critical Success Factors in Agile Software Projects." *The Journal of Systems and Software*, Vol. 81, No. 6, pp. 961-971, 2008.
- [15] N. B. Moe and T. Dingsoyr. "Scrum and team effectiveness: theory and practice." *Agile Processes in Software Engineering and Extreme Programming Lecture Notes in Business Information Processing*, Vol. 9, pp. 11-20, 2008.
- [16] J. Cho. "Issue and Challenges of Agile Software Development with Scrum." *Issues in Information Systems*, Vol. 9, Issue 2, pp. 188-195, 2008.
- [17] Y. I. Alzoubi and A. Q. Gill. "Agile global software development communication challenges: A systematic review." Available Online at http://pacis2014.org/data/PACIS_mainconference/pdf/pacis2014_submission_352.pdf, accessed on Aug. 05, 2014.
- [18] D. Kumlander. "Semi-and Fully Self-Organised Teams." *Advanced Techniques in Computing Sciences and Software Engineering*. Springer Netherlands, pp. 257-261, 2010.
- [19] P. Prabhakaran, "Skills for Scrum Agile Teams." *Agile Record*, July 2010, Issue 3, pp. 55-57, 2010.
- [20] A. A. Lorber and S. R. Tieszen. "A Starting Point for Negotiations - Delivering with a Heterogeneous Team." *In Proceedings of the Agile Conference (AGILE)*, pp. 148-155, August 13-17, 2012.
- [21] B. Kovitz. "Hidden Skills that Support Phased and Agile Requirements Engineering." *Requirements Engineering*, pp. 135-141, April 8, 2003.
- [22] A. Viljakainen and T. Turunen. "Requirements Engineering in Agile software projects." Available Online at <http://www.cs.joensuu.fi/~teturun/RE/REandAgile-final.pdf>, accessed on Aug. 20, 2014.
- [23] R. J. Wirfs-Brock. "Skills for the agile designer: seeing, shaping and discussing design ideas." *In Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, pp. 323-326, 2010.
- [24] R. J. Wirfs-Brock. "Designing with an Agile Attitude." *IEEE Software Magazine*, Vol. 26, Issue 2, pp. 68-69, 2009.
- [25] C. Tan and H. Teo. "Training Future Software Developers to Acquire Agile Development Skills." *Communication of The ACM*, Vol. 50, No. 12, pp. 97-98, 2007.
- [26] M. Cohn. *Succeeding with Agile: Software development using Scrum*. Second printing, Pearson Education, Inc., USA, pp. 31, 155-173, 2010.
- [27] L. Williams, G. Brown, A. Meltzer, and N. Nagappan. "Scrum + Engineering Practices: Experiences of Three Microsoft Teams." *In Proceedings of the International Symposium on Empirical Software Engineering and Measurement*, pp. 463- 471, September 22-23, 2011.