

การทดสอบแบบอัตโนมัติด้วยหุ่นเฟรมเวิร์ค
สำหรับการพัฒนาระบบแบบออนไลน์

นายอภิสิทธิ์ เชนาม



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช

พ.ศ. 2563

การทดสอบแบบอัตโนมัติด้วยหุ่นเฟรมเวิร์ค
สำหรับการพัฒนาระบบแบบออนไลน์

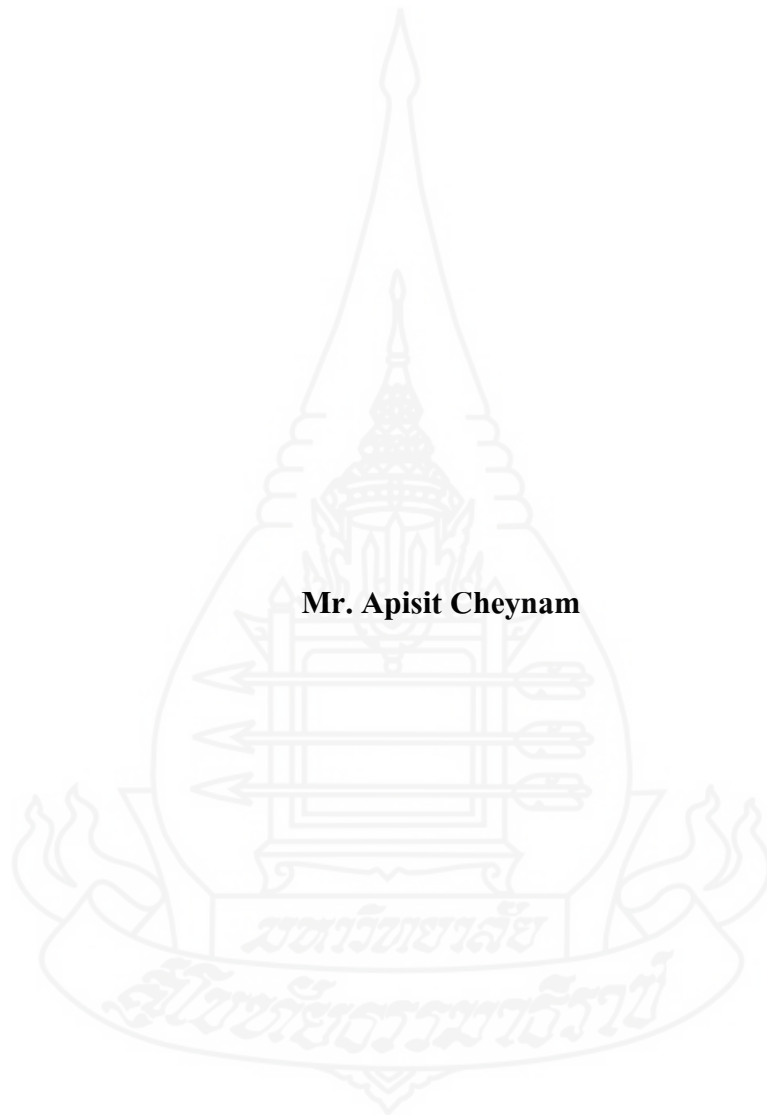


นายอภิสิทธิ์ เชนาม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช
พ.ศ. 2563

Automation Testing with a Robot Framework for Agile Software Development

Mr. Apisit Cheynam



A Thesis Submitted in Partial Fulfillment of the Requirement for the
Degree of Master of Science in Information and Communication Technology


School of Science and Technology
Sukhothai Thammathirat Open University

2020


หัวข้อวิทยานิพนธ์ การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบ
แบบออนไลน์
ชื่อและนามสกุล นายอภิสิทธิ์ เซชนาม
แขนงวิชา เทคโนโลยีสารสนเทศและการสื่อสาร
สาขาวิชา วิทยาศาสตร์และเทคโนโลยี
อาจารย์ที่ปรึกษา 1. อาจารย์ ดร.สรันย์ นาคถนอม
2. รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์

วิทยานิพนธ์นี้ ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรระดับ
ปริญญาโท เมื่อวันที่ 25 สิงหาคม 2564


..... ประธานกรรมการ
(รองศาสตราจารย์ ดร. สมชาย ปรากฏเจริญ)


..... กรรมการ
(อาจารย์ ดร.สรันย์ นาคถนอม)


..... กรรมการ
(รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์)


..... ประธานกรรมการบัณฑิตศึกษา
(รองศาสตราจารย์ ดร.เทพศักดิ์ นุชรัตพันธุ์)

ชื่อวิทยานิพนธ์ การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบ
แบบออนไลน์

ผู้วิจัย นายอภิสิทธิ์ เชนนาม

รหัสนักศึกษา 2619600220

ปริญญา วิทยาศาสตร์มหาบัณฑิต (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษา 1. อาจารย์ ดร.ศรัณย์ นาคถนอม
2. รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์

ปีการศึกษา 2563

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ (1) พัฒนาการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์ (2) ทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์ (3) ประเมินประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์ โดยมีขั้นตอนการดำเนินการวิจัยคือ (1) ขั้นตอนการพัฒนาการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์ (2) การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์ (3) ประเมินประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์

โดยการวิจัยครั้งนี้วัดจากจำนวนการทำงานทั้งหมด 88 งาน จากกลุ่มตัวอย่าง 151 คน โดยแบ่งเป็นการทดสอบซอฟต์แวร์แบบอัตโนมัติเปรียบเทียบกับ การทดสอบแบบมีผู้ปฏิบัติงานแบบการทำงานแบบออนไลน์ และ ประสิทธิภาพของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์โดยผลการวิจัย

ในส่วนของ การทดสอบแบบอัตโนมัติโดยเปรียบเทียบกับ การทดสอบแบบมีผู้พบว่า การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คมีค่าปริมาณผิดพลาดเชิงสัมพัทธ์ เท่ากับ 19.76, 19.22 และ 9.61 ตามลำดับ และค่าปริมาณผิดพลาดเชิงสัมพัทธ์ของประสิทธิภาพของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบออนไลน์ในด้านของขนาดซอฟต์แวร์ เท่ากับ 11.65 ในด้านของข้อผิดพลาดในการทดสอบ เท่ากับ 12.11 และ ในด้านของระยะเวลาในการทดสอบ เท่ากับ 14.73

คำสำคัญ: การทดสอบซอฟต์แวร์แบบอัตโนมัติ, โรบอทเฟรมเวิร์ค, การพัฒนาระบบแบบออนไลน์

Thesis: Automation Testing with a Robot Framework for Agile Software Development

Researcher: Mr. Apisit Cheynam

ID: 2619600220

Degree: Master of Science (Information and Communication Technology)

Advisor: 1. Dr. Sarun Nakthanom

2. Dr. Waranya Poonnawat, Associate Professor

Academic year: 2020

Abstract

The objectives of this research were to (1) develop automation test scripts using robot framework for agile software development (2) automate testing using robot framework for agile software development (3) evaluate performance for automation testing using robot framework for agile software development. Research methods consisted of the processes for developing automation test scripts using robot framework for agile software development, automation testing using robot framework for agile software development, and evaluating performance for automation testing using robot framework for agile software development.

This research measured 88 testing tasks with 151 representative samples which divided into 2 types comparison between manual testing and automation testing for agile software development and automation testing for agile software development performance.

The research shown that the automation testing compared with manual testing for agile software development proved that automation testing had higher quality than manual testing with MMRE equaled 19.76, 19.22 and 9.61 respectively. In addition MMRE for performance of automation testing using robot framework for agile software development regarding software sizes equaled 11.65, for testing error equaled 12.11, and for testing time using equaled 14.73.

Keywords: Automation Testing, Robot Framework, Agile Development

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี เนื่องจากได้รับความกรุณาเป็นอย่างอย่างสูงจาก อาจารย์ ดร.ศรันย์ นาคถนอม อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก และ รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์ อาจารย์ที่ปรึกษาร่วม ที่ให้ความรู้ คำแนะนำ คำปรึกษาด้านแนวคิด แนวทางและ หลักการในการจัดทำวิทยานิพนธ์ฉบับนี้ ทั้งยังเป็นส่วนหนึ่งช่วยในการตรวจสอบ รวมถึงแก้ไข ความถูกต้องในการจัดทำวิทยานิพนธ์ ตลอดจนให้คำแนะนำทางวิชาการที่ดี ผู้วิจัยขอขอบพระคุณไว้เป็นอย่างสูงไว้ ณ ที่นี้ อีกทั้งขอขอบพระคุณประธานกรรมการสอบวิทยานิพนธ์และ กรรมการสอบวิทยานิพนธ์ทุกท่าน ที่ได้ให้ข้อเสนอแนะและการชี้แนะเพื่อการปรับปรุงวิทยานิพนธ์ ให้ดียิ่งขึ้น

ขอขอบพระคุณประชากรและกลุ่มตัวอย่างทุกท่านที่ให้ความร่วมมือเป็นอย่างดีในการทำ ให้ข้อมูลเพื่อสร้างฐานความรู้ การทดสอบซอฟต์แวร์แบบอัตโนมัติ รวมถึงการช่วยเหลือในการเก็บ ข้อมูลในส่วนต่าง ๆ เพื่อให้วิทยานิพนธ์ฉบับนี้สมบูรณ์และบรรลุเป้าหมายตามแนวคิดและ สมมุติฐานที่วางไว้ ทั้งนี้ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์แก่ผู้ที่สนใจ และศึกษา ค้นคว้า หรือถูกนำไปใช้ในการอ้างอิงในการวิจัยด้านการทดสอบซอฟต์แวร์แบบ อัตโนมัติ หากแต่วิทยานิพนธ์ฉบับนี้มีข้อบกพร่องประการใด ผู้วิจัยต้องขออภัยและน้อมรับความ ผิดพลาดนี้ไว้หากมีโอกาสครั้งต่อไปจะได้นำไปปรับปรุงแก้ไขให้ดียิ่งขึ้นในอนาคต

อภิสิทธิ์ เชนนาม
พฤษภาคม 2564

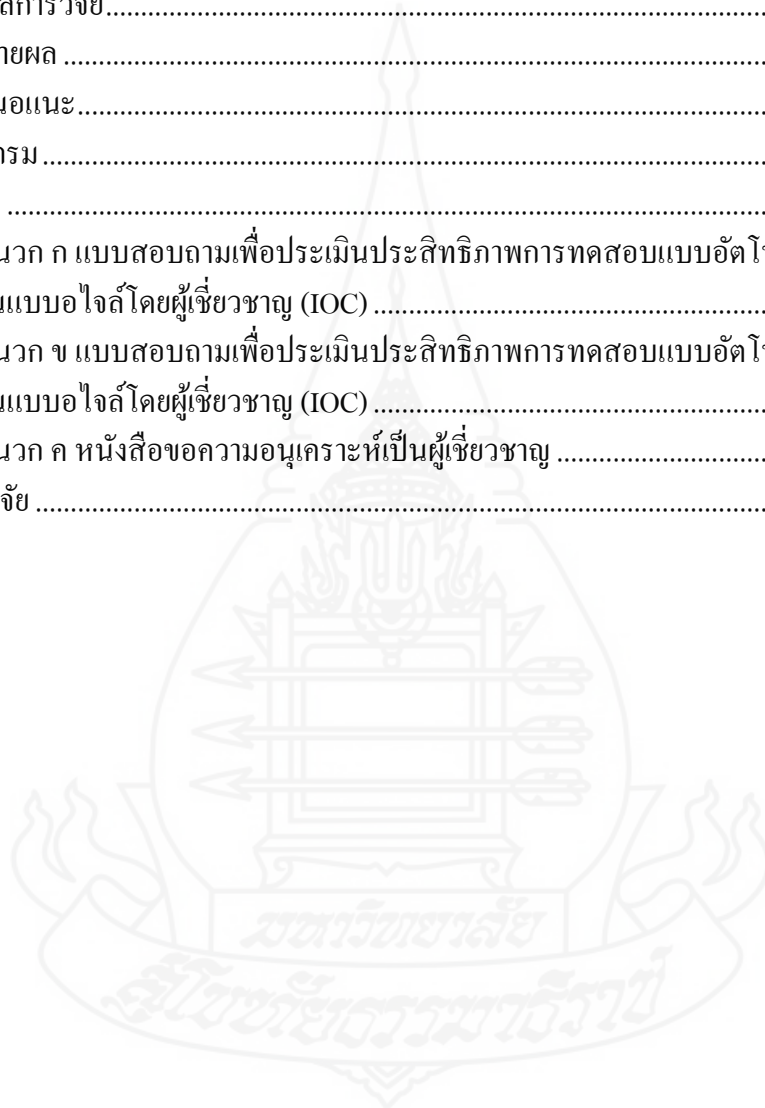


สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญตาราง	ฅ
สารบัญภาพ	ฉ
บทที่ 1 บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
วัตถุประสงค์การวิจัย.....	5
กรอบแนวคิดการวิจัย	5
ขอบเขตการวิจัย.....	6
นิยามศัพท์เฉพาะ	8
ประโยชน์ที่คาดว่าจะได้รับ	9
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง	10
การทดสอบระบบ.....	10
การทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์	16
โรบอทเฟรมเวิร์ค (Robot Framework).....	21
เครื่องมือทางสถิติ.....	23
งานวิจัยอื่น ๆ ที่เกี่ยวข้อง.....	30
บทที่ 3 วิธีดำเนินการวิจัย.....	36
พัฒนาการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและกรณีทดสอบแบบมือภายใต้รูปแบบการทำงานแบบออนไลน์.....	36
การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบออนไลน์	43
ประเมินประสิทธิภาพการทดสอบภาพซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบออนไลน์.....	44
บทที่ 4 ผลการวิจัย.....	68
ผลการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์	68
ผลการวิเคราะห์องค์ประกอบประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์	77

สารบัญ (ต่อ)

การประมาณค่าความแม่นยำ	81
บทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะ	86
สรุปผลการวิจัย.....	86
อภิปรายผล	87
ข้อเสนอแนะ.....	90
บรรณานุกรม.....	92
ภาคผนวก	92
ภาคผนวก ก แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์โดยผู้เชี่ยวชาญ (IOC)	92
ภาคผนวก ข แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์โดยผู้เชี่ยวชาญ (IOC)	111
ภาคผนวก ค หนังสือขอความอนุเคราะห์เป็นผู้เชี่ยวชาญ	129
ประวัติผู้วิจัย	135



สารบัญตาราง

	หน้า
ตารางที่ 2.1 เอกสารและงานวิจัยที่เกี่ยวข้อง	33
ตารางที่ 3.1 ตัวอย่างกรณีทดสอบ.....	38
ตารางที่ 3.2 ค่า Cronbach's Alpha ของกลุ่มตัวอย่าง 30 ราย	45
ตารางที่ 3.3 ค่า Cronbach's Alpha ของกลุ่มตัวอย่าง 88 ราย	46
ตารางที่ 3.4 ผลการตรวจสอบค่าเบ้ (Skewness)	50
ตารางที่ 3.5 ค่าสัมพันธประสิทธิ์สหสัมพันธ์ของตัวแปรต่าง ๆ (บางส่วน).....	50
ตารางที่ 3.6 ค่า KMO.....	51
ตารางที่ 3.7 ผลการสกัดปัจจัยแบบไม่หมุนแกน	52
ตารางที่ 3.8 ค่าความผันแปรของทุกตัวแปรสะสมที่อธิบายได้.....	54
ตารางที่ 3.9 ผลการหมุนแกน (Rotation).....	56
ตารางที่ 3.10 ตัวชี้วัดในแต่ละปัจจัย	58
ตารางที่ 3.11 เกณฑ์การประเมินความสอดคล้องของโมเดล	65
ตารางที่ 3.12 ค่าประมาณการ (Estimate) ของสัมประสิทธิ์ความสัมพันธ์ (Regression Weight) (Regression Weight)	65
ตารางที่ 3.13 ค่า Model Fit Summary	65
ตารางที่ 3.14 ค่า Modification indices.....	66
ตารางที่ 3.15 ค่า CMIN	66
ตารางที่ 3.16 ค่า GFI และ AGFI	67
ตารางที่ 3.17 ค่า RMSEA	67
ตารางที่ 4.1 ผลการทดสอบของกลุ่มที่ทำการทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โป้รเจค.....	70
ตารางที่ 4.2 ผลการทดสอบของกลุ่มทดสอบซอฟต์แวร์มากกว่า 1 ครั้ง ต่อ 1 โป้รเจค หรือเรียกว่า การทดสอบแบบถดถอย (Regression Testing).....	70
ตารางที่ 4.3 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โป้รเจค	72
ตารางที่ 4.4 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 2 ครั้ง ต่อ 1 โป้รเจค	73
ตารางที่ 4.5 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 3 ครั้ง ต่อ 1 โป้รเจค	75
ตารางที่ 4.6 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 4 ครั้ง ต่อ 1 โป้รเจค	76

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 4.7 ค่าสถิติที่ตรวจสอบโมเดลของค่าสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบบอใจล์จากขนาดซอฟต์แวร์ (ZTC_EFF)	78
ตารางที่ 4.8 ค่าสถิติที่ตรวจสอบโมเดลของค่าสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบบอใจล์จากข้อผิดพลาดของการทดสอบ (ZTC_ERR).....	80
ตารางที่ 4.9 ค่าสถิติที่ตรวจสอบโมเดลของค่าสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบบอใจล์จากเวลาของการทดสอบ (TIME_ERR)	80
ตารางที่ 4.10 ข้อมูลที่นำมาทดสอบ.....	82
ตารางที่ 4.11 ผลการคำนวณค่า MRE และ MMRE ของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบบอใจล์จากขนาดซอฟต์แวร์.....	82
ตารางที่ 4.12 ผลการคำนวณค่า MRE และ MMRE ของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบบอใจล์จากข้อผิดพลาดของการทดสอบ	83
ตารางที่ 4.13 ผลการคำนวณค่า MRE และ MMRE ของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบบอใจล์จากเวลาของการทดสอบ	84



สารบัญภาพ

	หน้า
ภาพที่ 1.1 ปัญหาของการทดสอบระบบแบบมือ ในกรณีที่ทีมพัฒนาได้มีการพัฒนาระบบใหม่.....	2
ภาพที่ 1.2 สถาปัตยกรรมของโรบอทเฟรมเวิร์ค (Mart, 2017)	4
ภาพที่ 1.3 กรอบแนวคิดการวิจัย.....	6
ภาพที่ 1.4 ขั้นตอนการพัฒนาระบบบนกระบวนการทำงานแบบอไจล์.....	8
ภาพที่ 2.1 ภาพแบบการทดสอบระบบ	10
ภาพที่ 2.2 ขั้นตอนการทดสอบระบบแบบมือ (Manual Testing).....	11
ภาพที่ 2.3 โครงสร้างของ Automation framework.....	13
ภาพที่ 2.4 วงจรชีวิตการทดสอบซอฟต์แวร์ (STLC : Software Testing Life Cycle).....	15
ภาพที่ 2.5 วงจรชีวิตการทำงานแบบอไจล์.....	18
ภาพที่ 2.6 กระบวนการพัฒนาที่ขับเคลื่อนด้วยการทดสอบ	20
ภาพที่ 2.7 โครงสร้างของโรบอทเฟรมเวิร์ค (robotframework, 2021).....	21
ภาพที่ 2.8 ตัวอย่างโมเดลหลักของโมเดลสมการโครงสร้าง.....	28
ภาพที่ 3.1 แผนภาพแสดงขั้นตอนการพัฒนาระบบทดสอบแบบมือและกรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค.....	37
ภาพที่ 3.2 ตัวอย่าง Setting ของโรบอทเฟรมเวิร์ค	40
ภาพที่ 3.3 ตัวอย่าง Variable ของโรบอทเฟรมเวิร์ค.....	40
ภาพที่ 3.4 ตัวอย่าง Keywords ของโรบอทเฟรมเวิร์ค	40
ภาพที่ 3.5 ตัวอย่าง กรณีทดสอบ ของโรบอทเฟรมเวิร์ค.....	40
ภาพที่ 3.6 ตัวอย่างชุดทดสอบกรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค.....	41
ภาพที่ 3.7 การรายงานผลการทดสอบด้วยโรบอทเฟรมเวิร์ค.....	42
ภาพที่ 3.8 ค่าสังเกตที่ผิดปกติของทุกตัวชี้วัดหลังจากปรับปรุงข้อมูลที่เป็น Outlier.....	47
ภาพที่ 3.9 ค่ามาตรฐานของทุกตัวแปร (บางส่วน).....	48
ภาพที่ 3.10 สมมุติฐานแบบจำลองวัดองค์ประกอบการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอไจล์	58
ภาพที่ 3.11 ตัวแปรที่อยู่ในปัจจัย F1	60
ภาพที่ 3.12 ตัวแปรที่อยู่ในปัจจัย F2	60
ภาพที่ 3.13 ตัวแปรที่อยู่ในปัจจัย F3	60
ภาพที่ 3.14 ตัวแปรที่อยู่ในปัจจัย F4	61
ภาพที่ 3.15 ตัวแปรที่อยู่ในปัจจัย F5	62
ภาพที่ 3.17 Model Specification ของ ZTC_Eff.....	64

สารบัญญภาพ (ต่อ)

	หน้า
ภาพที่ 3.17 Model Specification ของ ZTC_ERR	64
ภาพที่ 3.18 Model Specification ของ ZTC_TIME	64
ภาพที่ 3.19 แบบจำลองสมการ โครงสร้างตั้งต้น	64
ภาพที่ 4.1 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรม เวิร์คสำหรับการพัฒนาระบบแบบออนไลน์.....	77
ภาพที่ 4.2 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรม เวิร์คสำหรับการพัฒนาระบบแบบออนไลน์จากข้อผิดพลาดของการทดสอบ.....	78
ภาพที่ 4.3 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรม เวิร์คสำหรับการพัฒนาระบบแบบออนไลน์จากเวลาของการทดสอบ	80
ภาพที่ 5.1 แผนภูมิแสดงผลการทดสอบระบบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบ การทำงานแบบออนไลน์.....	86
ภาพที่ 5.2 แผนภูมิแสดงผลค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) ของแบบจำลองสมการ โครงสร้าง.....	87



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันมีการแข่งขันทางด้านธุรกิจเทคโนโลยีที่สูงและหลากหลาย ธุรกิจที่มีซอฟต์แวร์ที่ทันสมัยใช้งานง่าย ตอบโจทย์ผู้ใช้งานในทุกกลุ่มเป้าหมาย ออกสู่ตลาดของธุรกิจต่าง ๆ เป็นตัวสร้างความได้เปรียบทางการแข่งขันเหนือคู่แข่ง โดยภายใต้ซอฟต์แวร์ที่ดีนั้นต้องมีพื้นฐานมาจากกระบวนการทำงานที่ดี มีหลักเกณฑ์และมาตรฐาน ซึ่งการทำงานแบบอไจล์ (Agile Methodology) ถือเป็นแนวคิดที่ทำให้การทำงานมีประสิทธิภาพมากขึ้น (Team, 2018) ซึ่งในการทำงานด้านเทคโนโลยี มักมีการนำแนวการทำงานแบบอไจล์ เข้ามาใช้ในการทำงานตามวงจรพัฒนาระบบ โดยกระบวนการทดสอบระบบถือเป็นหนึ่งในกระบวนการที่สำคัญต่อวงจรการพัฒนาระบบ ที่สามารถนำกระบวนการทำงานแบบอไจล์ มาใช้ในการบริหารจัดการงานทดสอบระบบ

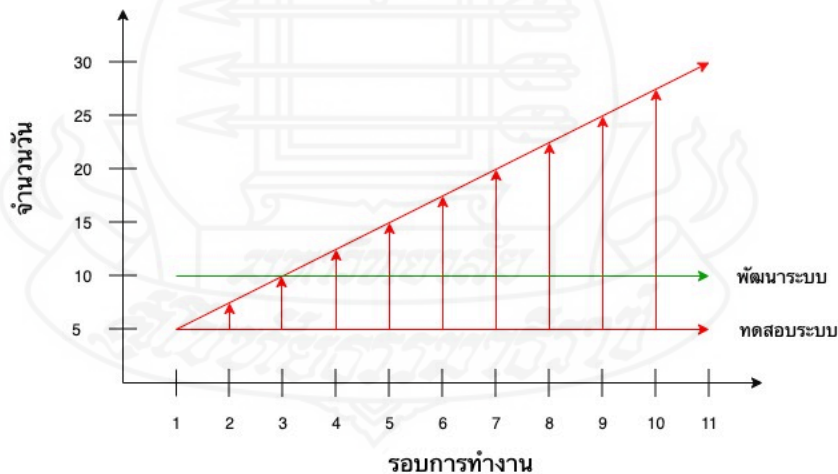
ในกระบวนการทดสอบระบบที่เกิดขึ้นในแต่ละครั้งนั้น เกิดขึ้นโดยความต้องการหรือความต้องการเพิ่มเติมของผู้ใช้งานของแต่ละระบบ ซึ่งเมื่อทีมวิเคราะห์และออกแบบระบบได้รับข้อเสนอแนะจากผู้ใช้งานแล้ว จึงนำข้อเสนอแนะ มาเข้ากระบวนการทำงานแบบอไจล์ เพื่อปรับปรุงแก้ไขระบบให้ตรงความต้องการของผู้ใช้งานระบบจริงที่สุด และทำการส่งต่อระบบให้กับทีมพัฒนาระบบ เพื่อพัฒนาระบบตามที่ทีมวิเคราะห์และออกแบบระบบได้ออกแบบไว้ หลังจากนั้น ทีมพัฒนาระบบจะส่งระบบที่สำเร็จแล้ว ให้กับทีมทดสอบระบบเพื่อทดสอบความถูกต้องของระบบ โดยใช้บุคคลเป็นผู้ทดสอบ ซึ่งจะทำการทดสอบระบบทั้งหมดที่ทีมพัฒนาระบบได้พัฒนาขึ้น โดยอ้างอิงความถูกต้องจากทีมวิเคราะห์และออกแบบระบบที่ได้ออกแบบระบบไว้ เมื่อทำการทดสอบระบบสำเร็จแล้ว ทีมทดสอบระบบจะต้องทำการทดสอบระบบกับผู้ใช้งานจริง เมื่อทำการทดสอบระบบกับผู้ใช้งานจริงสำเร็จแล้ว จึงส่งผลการทดสอบให้กับทีมพัฒนาระบบ เพื่อทำการนำระบบไปใช้สภาพแวดล้อมจริง เพื่อให้ผู้ใช้งานได้ใช้ระบบใหม่หลังจากการแก้ไขระบบ

การทดสอบระบบ (Software Testing) เป็นกระบวนการในการประเมินและปรับปรุงคุณภาพของซอฟต์แวร์ โดยการค้นหาข้อผิดพลาดและ/หรือจุดบกพร่องของซอฟต์แวร์ที่ได้มาจากเอกสารความต้องการของลูกค้า (User Requirement) โดยทดสอบจากการพัฒนาซอฟต์แวร์ของทีมพัฒนาซอฟต์แวร์ (Developer) โดยสามารถระบุแนวทางที่ทำให้เกิดขึ้นได้และก็ทำการแก้ไข โดยการทดสอบระบบจะมีการจำลองหรือออกแบบกรณีทดสอบ เพื่อยืนยันความถูกต้องของระบบ โดยมีการใช้ Microsoft Excel , Microsoft Word หรือเครื่องมืออื่น ๆ เป็นเครื่องมือที่ใช้ในการจำลองหรือออกแบบกรณีทดสอบเพื่อให้ผู้ทดสอบซอฟต์แวร์ (Software Tester) ทำการทดสอบซอฟต์แวร์/เปรียบเทียบผลการทดสอบ (Test Execution) โดยผู้ทดสอบจะทำการทดสอบตามกรณีทดสอบ

(Test Cases) โดยผู้ทดสอบเอง โดยอ้างอิงความถูกต้องของการทำงานของระบบจากเอกสารความต้องการของลูกค้า (User Requirement)

การทดสอบตามกรณีทดสอบด้วยมือของผู้ทดสอบเองนั้นอาจมีประสิทธิภาพที่ดีสำหรับการทดสอบระบบที่ไม่ใหญ่และความซับซ้อนไม่มากนัก แต่ถ้าหากนำไปใช้กับระบบที่มีขนาดใหญ่มากนั้น ก็อาจจะไม่เป็นผลดีนักในเรื่องของการใช้เวลาการทดสอบ และโอกาสเสี่ยงที่จะเกิดข้อผิดพลาดที่เกิดจากผลกระทบของผู้ทดสอบระบบ (Human Error) ซึ่งอาจจะส่งผลเรื่องประสิทธิภาพของระบบที่ไม่ได้คุณภาพ

ในกรณีที่การทดสอบระบบแบบมือ (Manual Testing) นั้นมักพบปัญหาหรือข้อผิดพลาดที่เกิดจากทีมพัฒนาระบบทำการพัฒนาผิดพลาดและไม่ตรงตามที่ทีมวิเคราะห์และออกแบบระบบออกแบบไว้นั้น ผู้ทดสอบระบบแบบมือ (Manual Tester) จะต้องส่งรายงานข้อผิดพลาดที่เกิดขึ้นนั้นไปให้ทีมพัฒนาระบบ ทำการแก้ไขให้ถูกต้อง ซึ่งหลังจากที่ทีมพัฒนาระบบ ทำการแก้ไขระบบสำเร็จแล้ว ทีมทดสอบระบบแบบมือ จะต้องนำระบบที่ถูกแก้ไขนั้น มาทำการทดสอบใหม่ทั้งระบบ เพื่อหาข้อผิดพลาดที่เกิดจากผลกระทบของการแก้ไขระบบใหม่ ซึ่งกรณีที่ทีมทดสอบระบบนั้นพบข้อผิดพลาดหลายข้อก็จะต้องทำการทดสอบระบบแบบมือใหม่ (Regression) ซ้ำ ๆ หลาย ๆ รอบ ซึ่งส่งผลให้การทดสอบระบบนั้นใช้เวลานานเกินกำหนด และซึ่งทีมทดสอบระบบแบบมือ ทำการทดสอบระบบแบบมือใหม่ ซ้ำ ๆ หลาย ๆ ครั้งนั้นอาจส่งผลให้เกิดความผิดพลาดที่เกิดขึ้นจากทีมทดสอบระบบแบบมือซึ่งอาจเป็นเหตุให้มีข้อผิดพลาดเกิดขึ้นบนระบบจริงได้ (सानุตร์, 2561; Jasper, 2012)



ภาพที่ 1.1 ปัญหาของการทดสอบระบบแบบมือ ในกรณีที่ทีมพัฒนามีการพัฒนาใหม่

จากปัญหาดังกล่าว จึงได้มีกระบวนการทดสอบระบบแบบอัตโนมัติ เพื่อมาแก้ปัญหาที่เกิดขึ้น โดยทีมทดสอบแบบอัตโนมัติ ทำหน้าที่คิด วิเคราะห์ ออกแบบการสร้าง กรณีทดสอบของซอฟต์แวร์สำหรับการทดสอบระบบแบบอัตโนมัติ การทำการทดสอบแบบอัตโนมัตินั้นสามารถ

ช่วยลดเวลาและลดความเสี่ยงข้อผิดพลาดที่เกิดจากผลกระทบของการแก้ไขระบบใหม่ แบบไม่ต้องทำการทดสอบระบบแบบมือ (Paranun, 2016) โดยใช้เครื่องมือที่ชื่อว่าโรบอทเฟรมเวิร์คมาใช้ในการพัฒนาเทสต์สคริปต์ และใช้คำสั่งการทดสอบแบบอัตโนมัติ และยังสามารถนำเสนอเป็นภาพแบบรายงานผลการทดสอบแบบอัตโนมัติได้ด้วย

จากการที่ผู้ทำวิจัยทำการศึกษเปรียบเทียบเครื่องการทดสอบระบบแบบอัตโนมัติ ได้พบว่าโรบอทเฟรมเวิร์ค เครื่องมือสำหรับการทดสอบระบบแบบอัตโนมัติ (Test Automation Framework) ที่ไม่มีค่าใช้จ่าย โดยทำงานผ่านวิธีการสร้างคีย์เวิร์คฟังก์ชัน (Keyword-Driven Testing) ที่ใช้สำหรับทดสอบซึ่งมีทั้งที่อยู่ในไลบรารี (Library Keyword) หรือ สร้างขึ้นมาเพื่อให้เฉพาะกับระบบ (User Keyword) มาประกอบกันให้เป็นเทสต์สคริปต์ (Test Scripts) เพื่อใช้สำหรับการบวนการยอมรับการทดสอบ (Acceptance Test Driven Development (ATDD) และเป็นเครื่องมือที่ใช้งานในภาพแบบของ ไวยากรณ์ (Syntax) ที่ชื่อทาบูลา (Tabular) สำหรับการสร้างกรณีทดสอบแบบอัตโนมัติ โรบอทเฟรมเวิร์คยังมีความสามารถในการออกรายงานผลการทดสอบได้อย่างละเอียดโดย รายงานเป็นเวลาในการรันกรณีทดสอบเป็นรายข้อและบอกเวลาในการรันกรณีทดสอบ (benjawan T, 2017) โดยโรบอทเฟรมเวิร์ค มีองค์ประกอบหลัก ๆ ในการทำงานดังนี้

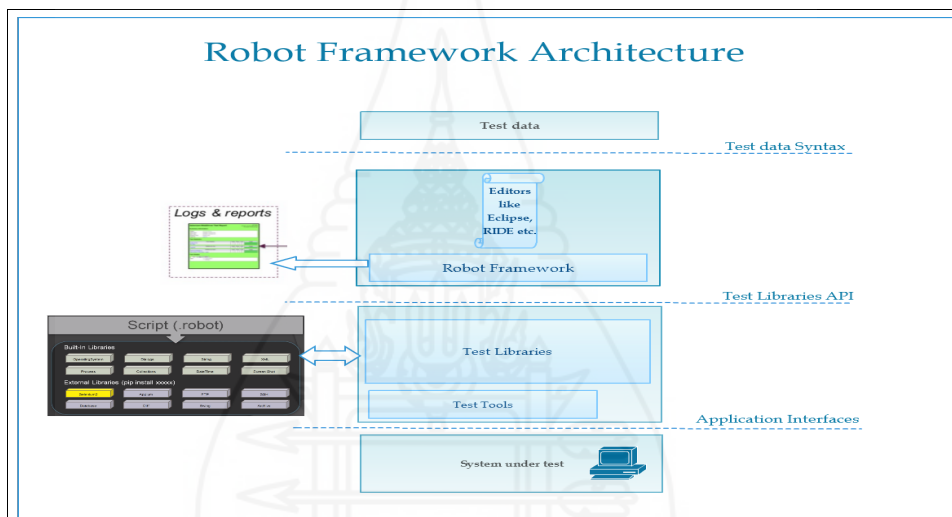
1. การตั้งค่า (Setting) ในส่วนของการตั้งค่า ใช้ในการ นำเข้าไฟล์ (Include File), การกำหนดคำอธิบายของไฟล์ (Document Description), การทำสูทเซทอัป (Suite Setup), การทำเทียร์ดาวน์ (Teardown) และ การนำเข้าไลบรารี (Import Library) ซึ่งการ การนำเข้าไลบรารีสามารถแบ่งไป 3 ส่วน นั่นคือ ไลบรารีมาตรฐานของโปรแกรม (Standard), ไลบรารีที่ติดตั้งจากภายนอก (External), ไลบรารีที่ผู้ใช้งานพัฒนาขึ้นเอง (Other) (robotframework, 2019)

2. วาเรียเบิล (Variables) คือการประกาศตัวแปรเก็บข้อมูลการทดสอบต่าง ๆ สำหรับการนำไปเขียนเทสต์สคริปต์

3. คีย์เวิร์ค (Keyword) เป็นฟังก์ชัน หรือชุดคำสั่งที่เขียนเพื่อนำไปเรียกใช้หลายๆ ครั้งโดยสามารถเขียนเป็นภาษาไทยหรืออังกฤษก็ได้

4. กรณีทดสอบ (Testcase) เป็นที่ที่ใช้ในการเขียน สิ่งที่ต้องการทดสอบลงไป เป็นการกำหนดขั้นตอนการทดสอบ (Test Step) และเงื่อนไขต่าง ๆ ของกรณีทดสอบ การเขียนกรณีทดสอบ นั้นจะมีการจัดเตรียมข้อมูล (Test data) เพื่อใช้กับกรณีทดสอบ เช่น ข้อมูลอีเมล, ข้อมูลเบอร์โทรศัพท์ และ รายละเอียดอื่น ๆ ที่จำเป็นสำหรับการทดสอบในหัวข้อการทดสอบ (Test Scenario) นั้น ๆ เพื่อให้การทดสอบใกล้เคียงหรือเปรียบเสมือนผู้ใช้ (User) มาใช้งานจริงมากที่สุด ซึ่งในกรณีที่ระบบมีขนาดใหญ่ การจัดเตรียมข้อมูลนั้นก็จะมีปริมาณที่มาก ซึ่งโรบอทเฟรมเวิร์ค นั้นมีการรองรับข้อมูลโดยสามารถเก็บข้อมูลไว้ที่ไฟล์ Excel และใช้ โรบอทเฟรมเวิร์ค-เอกซ์เซลไลบรารี (Robotframework – Excelibrary) เป็นไลบรารีที่ใช้เรียกข้อมูลจากเอกซ์เซลไปใช้งานในโรบอทเฟรมเวิร์ค โดยใช้โรบอทเฟรมเวิร์คเป็นเครื่องมือที่ช่วยในการพัฒนากรณีทดสอบแบบอัตโนมัติ เป็นเครื่องมือที่ใช้งานและเข้าใจได้ง่าย รองรับการเขียนกรณีทดสอบแบบอัตโนมัติ

สำหรับเว็บแอปพลิเคชัน (Web application) โมบายแอปพลิเคชัน (Mobile Application) และ ส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) โดยทำงานผ่านวิธีการคีย์เวิร์ด – ไดรฟ์เวน (Keyword-Driven) ที่ใช้ชุดคำสั่งทดสอบ (Test Library) เป็นแหล่งรวบรวมชุดคำสั่งการทดสอบต่าง ๆ เพื่อนำใช้งานกับโรบอทเฟรมเวิร์คโดยชุดคำสั่งทดสอบ ที่ โรบอทเฟรมเวิร์ค รองรับนั้นมีทั้งภาพแบบที่เป็นมาตรฐานอยู่แล้วและแบบที่ผู้ทดสอบพัฒนาขึ้นมาเองเพื่อปรับใช้ให้เข้ากับระบบที่ทดสอบ ตัวอย่างของชุดคำสั่งทดสอบ เช่น ซิลิเนียมทูลไอบรารี่ (Selenium2Library) ใช้สำหรับเก็บ ชุดคำสั่งการทดสอบที่ใช้ทดสอบกับเว็บแอปพลิเคชัน (Web Application) แอปเพียมไอบรารี่ (AppiumLibrary) ใช้สำหรับเก็บ ชุดคำสั่งการทดสอบที่ใช้ทดสอบกับโมบายแอปพลิเคชัน (Mobile Application) เป็นต้น



ภาพที่ 1.2 สถาปัตยกรรมของโรบอทเฟรมเวิร์ค (Mart, 2017)

จากการที่ผู้ทำวิจัยทำการศึกษาระบบแบบอัตโนมัติ และการทดสอบระบบแบบมือ ได้เล็งเห็นความสำคัญในการนำการทดสอบระบบแบบอัตโนมัติ นั้นอาจมีประสิทธิภาพมากกว่าการทดสอบระบบแบบมือ ในเรื่องของเวลาในการใช้ทดสอบระบบ และโอกาสที่จะเกิดความเสียหายผิดพลาดที่เกิดจากผลกระทบของผู้ทดสอบระบบ และผู้ทำวิจัยได้เล็งเห็นความสำคัญของภาพแบบการทำงานแบบออโต้ ที่ถือเป็นแนวคิดการทำงานที่ช่วยเพิ่มประสิทธิภาพในการทดสอบระบบ จึงได้นำการทดสอบระบบแบบอัตโนมัติ มาประยุกต์ใช้กับภาพแบบการทำงานแบบออโต้ เพื่อเพิ่มประสิทธิภาพสูงสุดต่อระบบ และด้วยการนำประสิทธิภาพของขั้นตอนการทดสอบระบบและกรณีการทดสอบมาผสมผสานเข้ากับเทคนิคการทำงานแบบอัตโนมัติด้วยการพัฒนาคำสั่งโปรแกรมการทำงานโรบอทเฟรมเวิร์ค เพื่อการทดสอบ จึงได้จัดทำเป็นงานวิจัย การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบออโต้

1.2 วัตถุประสงค์การวิจัย

1.2.1 เพื่อพัฒนาการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบออนไลน์

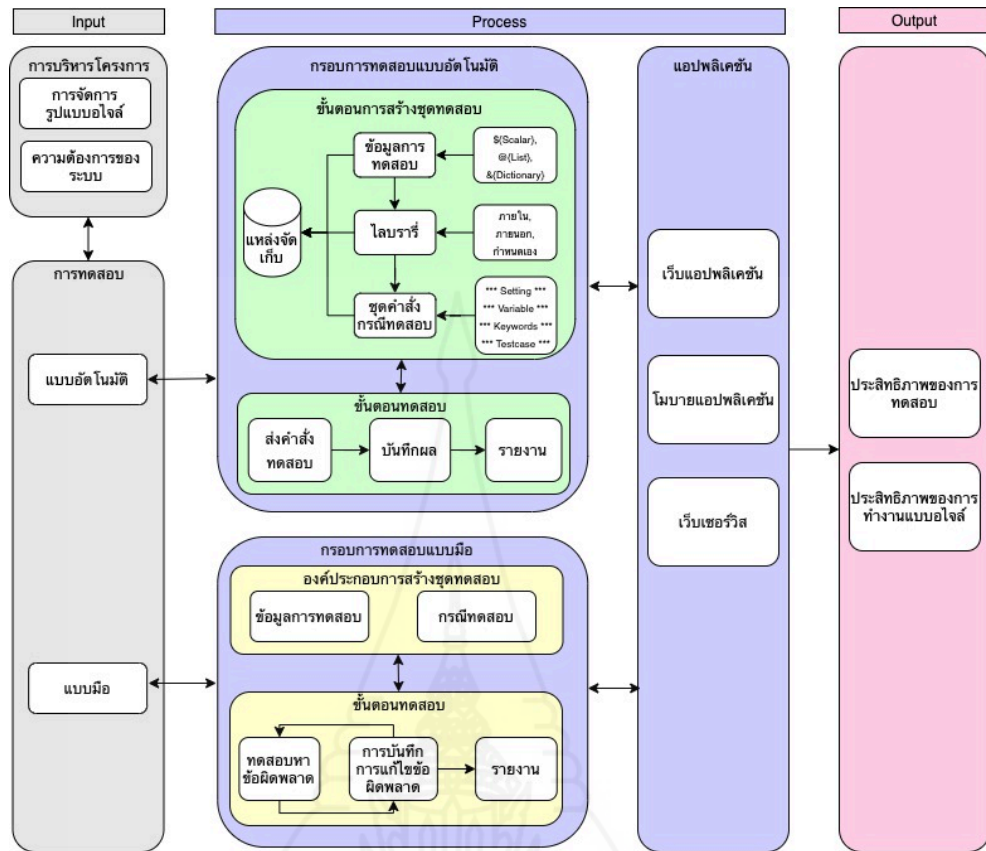
1.2.2 เพื่อทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบออนไลน์

1.2.3 เพื่อประเมินประสิทธิภาพการทดสอบภาพซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบออนไลน์

1.3 กรอบแนวคิดการวิจัย

สำหรับการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบออนไลน์ เป็นการวิจัยเชิงทดลองเปรียบเทียบระหว่างการทดสอบซอฟต์แวร์แบบอัตโนมัติและการทดสอบซอฟต์แวร์แบบมือ ในด้านประสิทธิภาพความแม่นยำของระยะเวลา ความถูกต้องระบบ และประสิทธิภาพของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบออนไลน์ โดยกระบวนการทำงานตามกรอบแนวคิดจะเน้นการพัฒนาระบบทดสอบแบบอัตโนมัติ ด้วยโรบอทเฟรมเวิร์ก สำหรับนำไปใช้ในการทดสอบระบบแบบอัตโนมัติ ดังภาพที่ 1.3





ภาพที่ 1.3 กรอบแนวคิดการวิจัย

1.4 ขอบเขตการวิจัย

1.4.1 เนื้อหาและทฤษฎีที่เกี่ยวข้อง

การวิจัยครั้งนี้ผู้วิจัยได้ดำเนินการวิจัยให้ครอบคลุม ดังนี้

1.4.1.1 การทดสอบระบบแบบอัตโนมัติโดยใช้โรบอทเฟรมเวิร์ค เป็นเครื่องมือสำหรับการทดสอบระบบแบบอัตโนมัติที่ไม่มีค่าใช้จ่าย โดยทำงานผ่านสร้างฟังก์ชัน ที่ใช้สำหรับทดสอบซึ่งมีทั้งที่อยู่ในไลบรารี หรือ สร้างขึ้นมาเพื่อให้เฉพาะกับระบบมาประกอบกันให้เป็นเทสต์สคริปต์เพื่อใช้สำหรับการทดสอบโดยใช้โรบอทเฟรมเวิร์ค และออกรายงานผลการทดสอบ โดย รายงานเป็นเวลาในการรันกรณีทดสอบเป็นรายข้อและบอกเวลาในการรันกรณีทดสอบ

1.4.1.2 แนวคิดและกระบวนการทำงานแบบบอใจล์เป็นแนวคิดในการทำงานที่เน้นการสื่อสารและปฏิสัมพันธ์กันระหว่างคนเพื่อให้การทำงานหรือการทำให้งานมีการตอบสนองต่อผู้ใช้งานที่ดีที่สุด โดยแนวคิดและกระบวนการทำงานแบบบอใจล์ มีขั้นตอนและกระบวนการทั้งหมด 5 ดังนี้

1) ปัญหาหรือความต้องการของระบบ (Requirement) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการรับปัญหาหรือความต้องการของระบบ คือ ผู้ใช้งาน สรุปได้ว่า ปัญหาหรือความต้องการของระบบ ได้มาจากการที่ผู้ใช้งานนั้นมีความต้องการหรือพบปัญหาเกี่ยวกับระบบที่ใช้งานอยู่และต้องการให้มีการแก้ไขหรือเพิ่มฟีเจอร์ (Feature) ให้กับระบบ ผู้ใช้งานจะให้เอกสารปัญหาหรือความต้องการมาในภาพแบบ ยูเซอร์สตอรี (User Story) ซึ่งทีมงานอโกลด์จะต้องรับปัญหาหรือความต้องการไปสร้างเป็นงานตามลักษณะงานของตัวเอง

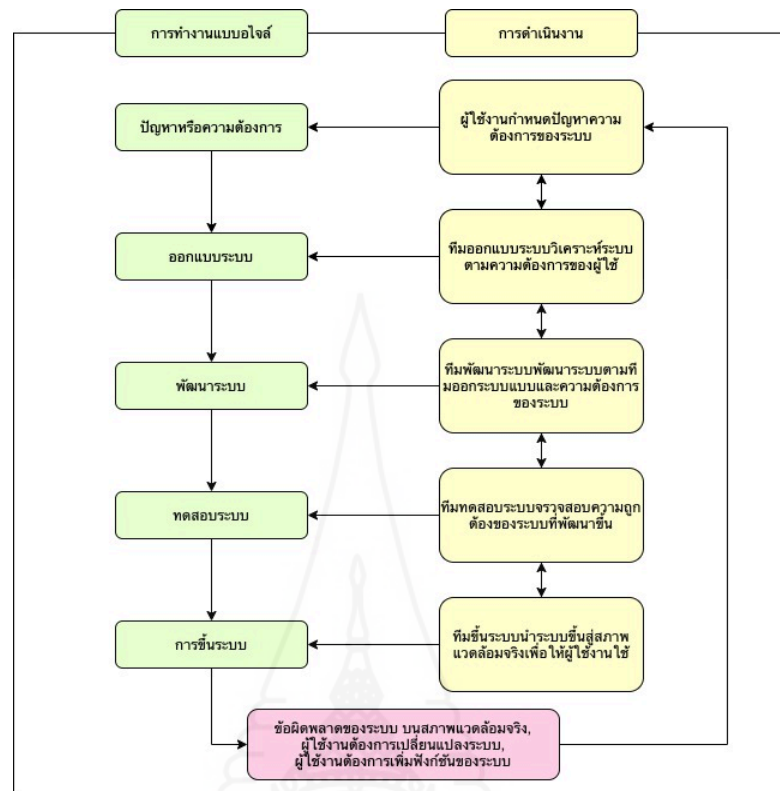
2) ออกแบบระบบ (Design) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการออกแบบระบบ คือ ผู้วิเคราะห์และออกแบบระบบ (System Analyst) เป็นผู้ที่ได้รับ ปัญหาหรือความต้องการจากผู้ใช้งานและมาทำการวิเคราะห์และออกแบบแนวทางการพัฒนาซอฟต์แวร์เพื่อให้ทีมพัฒนาระบบนำไปพัฒนาให้ตรงกับความต้องการของผู้ใช้งาน

3) พัฒนาระบบ (Development) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการพัฒนาระบบ คือ ผู้พัฒนาระบบ (Developer / Programmer) เป็นผู้ที่ทำหน้าที่พัฒนาซอฟต์แวร์ตามที่ผู้วิเคราะห์และออกแบบระบบ ได้ออกแบบไว้โดยสอดคล้องกับปัญหาหรือความต้องการของระบบของผู้ใช้งานให้สำเร็จถึงจะนำซอฟต์แวร์ที่เสร็จสมบูรณ์ส่งให้ทีมทดสอบระบบทำการตรวจสอบ

4) ทดสอบระบบ (Software Testing) เป็นการตรวจสอบความถูกต้องของซอฟต์แวร์หลังจากที่ ผู้พัฒนาระบบทำการพัฒนาเสร็จสมบูรณ์ ซึ่งการทดสอบระบบนั้นต้องใช้เวลาและความละเอียดรอบคอบ ซึ่งการทำงานแบบอโกลด์เป็นการทำงานที่ต้องอาศัยความรวดเร็วและถูกต้องแม่นยำ ผู้จัดทำวิจัยจึงได้นำการทดสอบแบบ โดยใช้โรบอทเฟรมเวิร์ค มาใช้ในการทดสอบ เพื่อเป็นการลดเวลาและความผิดพลาดที่จะเกิดขึ้นจากมนุษย์ โดยทดสอบแบบรวมหน่วย (SIT : System Integration Test) และเพื่อเป็นการส่งมอบซอฟต์แวร์ให้กับผู้ใช้ (End User) อย่างมีคุณภาพและรวดเร็ว โดยลดข้อผิดพลาดในกระบวนการทดสอบการใช้งาน โดยผู้ใช้งานจริง (UAT : User Acceptance Test)

5) การขึ้นระบบ (Deployment) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการการขึ้นระบบ คือ ระบบที่ผ่านกระบวนการทดสอบเรียบร้อยแล้วขึ้นไปให้ลูกค้าได้ใช้งาน

จะสังเกตได้ว่าการทำงานแบบอโกลด์ นั้นทุกฝ่ายมีการทำงานด้วยกันเพื่อมุ่งเน้นการส่งมอบซอฟต์แวร์ให้ผู้ใช้งาน ได้รวดเร็ว แม่นยำและมีประสิทธิภาพ โดยผู้วิจัยได้ทำขั้นตอนการพัฒนาระบบโดยใช้ภาพแบบการทำงานแบบอโกลด์ และความเกี่ยวข้องของการทดสอบ (Software Testing) ระบบกับภาพแบบการทำงานแบบอโกลด์ (Agile Methodology) ดังภาพที่ 1.4



ภาพที่ 1.4 ขั้นตอนการพัฒนากระบวนการทำงานแบบไจล์

1.4.1.3 แบบจำลองสมการโครงสร้าง (Structural Equation Modeling) หรือ SEM เป็นเทคนิควิธีการวิเคราะห์ทางสถิติโดยมุ่งเน้น โมเดลความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝงเชิงทฤษฎี ที่มีความสัมพันธ์ต่อกันหลาย ๆ ตัวแปร หรือใช้วิเคราะห์สำหรับโมเดลความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝง (Latent Variables) กับตัวแปรสังเกตได้ (Observed Variable) โดยทำการประมาณค่าพารามิเตอร์ของโมเดลพร้อมกันทั้งหมดด้วยระบบสมการโมเดลโครงสร้าง หรือ SEM เพื่อเป็นเทคนิคทางสถิติที่ใช้ในการยืนยัน (Confirmatory) ว่าผลงานวิจัยมีความถูกต้องมากน้อยเพียงใดเมื่อเปรียบเทียบกับข้อมูลจริงในปรากฏการณ์หรือข้อมูลเชิงประจักษ์

1.5 นิยามศัพท์เฉพาะ

1.5.1 โรบอทเฟรมเวิร์ค (Robot Framework) เป็นเครื่องมือสำหรับการทดสอบระบบแบบอัตโนมัติ ใช้ฟังก์ชันในการทำงานโดยใช้ข้อมูลการทดสอบในการทำงานและออกรายงานผลการทดสอบ

1.5.2 การทดสอบซอฟต์แวร์แบบอัตโนมัติ (Automation Testing) คือการทดสอบระบบโดยใช้ซอฟต์แวร์ในการทดสอบระบบแทนการใช้คน

1.5.3 การทดสอบซอฟต์แวร์แบบมือ (Manual Testing) คือการใช้คนทำการทดสอบระบบ เก็บผลการทดสอบ และรายงานผลการทดสอบ

1.5.4 อไจล์ (Agile Software Development) เป็นแนวคิดในการทำงานที่เน้นการสื่อสารและปฏิสัมพันธ์กันระหว่างคน เพื่อให้การทำงานหรือการทำให้งานมีการตอบสนองต่อผู้ใช้งานที่ดีที่สุด

1.5.5 ความต้องการของระบบ (Requirement) คือ เอกสารที่ผู้ใช้งาน นั้นมีความต้องการหรือพบปัญหาเกี่ยวกับระบบที่ใช้งานอยู่และต้องการให้มีการแก้ไขหรือเพิ่มฟีเจอร์ ให้กับระบบ

1.5.6 ออกแบบระบบ (Design) คือ การออกแบบหรือแนวทางการพัฒนาซอฟต์แวร์เพื่อให้ทีมพัฒนาระบบ นำไปพัฒนาให้ตรงกับความต้องการของผู้ใช้งาน

1.5.7 พัฒนาระบบ (Development) คือการพัฒนาซอฟต์แวร์ตามที่ผู้วิเคราะห์และออกแบบระบบ ได้ออกแบบไว้โดยสอดคล้องกับปัญหาหรือความต้องการของระบบของผู้ใช้งาน

1.5.8 ทดสอบระบบ (Software Testing) คือ การตรวจสอบความถูกต้องของซอฟต์แวร์ หลังจากที่ ผู้พัฒนาระบบทำการพัฒนาเสร็จสมบูรณ์

1.5.9 การทดสอบแบบรวมหน่วย (System Integration Test) คือ การทดสอบเพื่อทดสอบว่าระบบย่อยต่าง ๆ สามารถทำงานร่วมกันได้อย่างถูกต้อง ตรงตามวัตถุประสงค์

1.5.10 การขึ้นระบบ (Deployment) การนำระบบขึ้นใช้งานในสภาพแวดล้อมที่มีผู้ใช้จริง เป็นผู้ใช้งาน

1.5.11 ความผิดพลาด (Error) เป็นการคลาดเคลื่อนระหว่างการคำนวณ การสังเกต การวัดค่า หรือเงื่อนไขตามความเป็นจริงที่ระบุ หรือความแตกต่างระหว่างผลลัพธ์ที่แท้จริงและผลลัพธ์ที่คาดไว้ นอกจากนี้ยังหมายถึงข้อผิดพลาดที่เกิดจากมนุษย์ที่รวมไปถึงความเข้าใจผิด และการแปลความหมายผิดพลาดต่าง ๆ ที่ส่งผลให้เกิดปัญหาขึ้นในระบบ

1.5.12 ความคลาดเคลื่อน (Fault) ความไม่ถูกต้องตามขั้นตอน กระบวนการ หรือการนิยามข้อมูลภายในโปรแกรมคอมพิวเตอร์ ซึ่งส่งผลให้ทำงานไม่ถูกต้องตามที่คาดไว้

1.5.13 ข้อบกพร่อง (Defect / Bug) เป็นข้อบกพร่องภายในส่วนประกอบหรือระบบที่ส่งผลให้ระบบเกิดความล้มเหลวในการทำงานตามฟังก์ชันที่กำหนดไว้ ทั้งความล้มเหลว ความผิดพลาด และความคลาดเคลื่อนจะถูกเรียกรวม ๆ กันว่าข้อบกพร่อง

1.5.15 กรณีทดสอบ (Test Case) เป็นกลุ่มของค่าอินพุต เงื่อนไขในการประมวลผลและผลลัพธ์ที่คาดการณ์ไว้ ถูกพัฒนาโดยมีเป้าหมายเฉพาะ เช่น การทดสอบเส้นทางการทำงานของโปรแกรมหรือการทำงานตามความต้องการของระบบที่กำหนดไว้

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้กรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอไจล์

1.6.2 ได้ผลการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอไจล์

1.6.3 ได้ผลการประเมินประสิทธิภาพการทดสอบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค
ภายใต้รูปแบบการทำงานแบบไฮลิ์



บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการวิจัย เรื่อง การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอจี้ ได้มีการศึกษาค้นคว้า ทบทวนทฤษฎีและวรรณกรรมต่าง ๆ ที่เกี่ยวข้องกับเรื่อง การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอจี้ เพื่อนำรายละเอียดที่เกี่ยวข้องที่ได้ศึกษาและทบทวนวรรณกรรมมาใช้ประกอบการทำวิจัย โดยการวิจัยนี้มุ่งหมายเพื่อประเมินประสิทธิภาพของการทดสอบระบบโดยใช้โรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอจี้ โดยผู้วิจัยได้ศึกษาทฤษฎี รวบรวมเอกสาร และงานวิจัยที่เกี่ยวข้อง โดยครอบคลุมรายละเอียดดังต่อไปนี้

- 2.1 การทดสอบระบบ
- 2.2 การทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้
- 2.3 โรบอทเฟรมเวิร์ค
- 2.4 เครื่องมือทางสถิติ
- 2.5 งานวิจัยอื่น ๆ ที่เกี่ยวข้อง

2.1 การทดสอบระบบ

การทดสอบระบบ (Software Testing) เป็นขั้นตอนหลักของการพัฒนาซอฟต์แวร์ คือ กระบวนการตรวจสอบข้อผิดพลาดของซอฟต์แวร์ โดยมีวัตถุประสงค์เพื่อยืนยันว่าซอฟต์แวร์ที่ถูกพัฒนาขึ้นมานั้นเป็นไปตามมาตรฐานข้อกำหนด และความปลอดภัย ตามที่ลูกค้ากำหนดไว้ โดยประกอบด้วย การตรวจสอบกระบวนการพัฒนา (Verification) และการตรวจสอบผลลัพธ์ของการพัฒนา (Validation) (Smith, 2020) ได้จำแนกภาพแบบการทดสอบระบบได้เป็น 3 ประเภท ดังภาพที่ 2.1



ภาพที่ 2.1 ภาพแบบการทดสอบระบบ

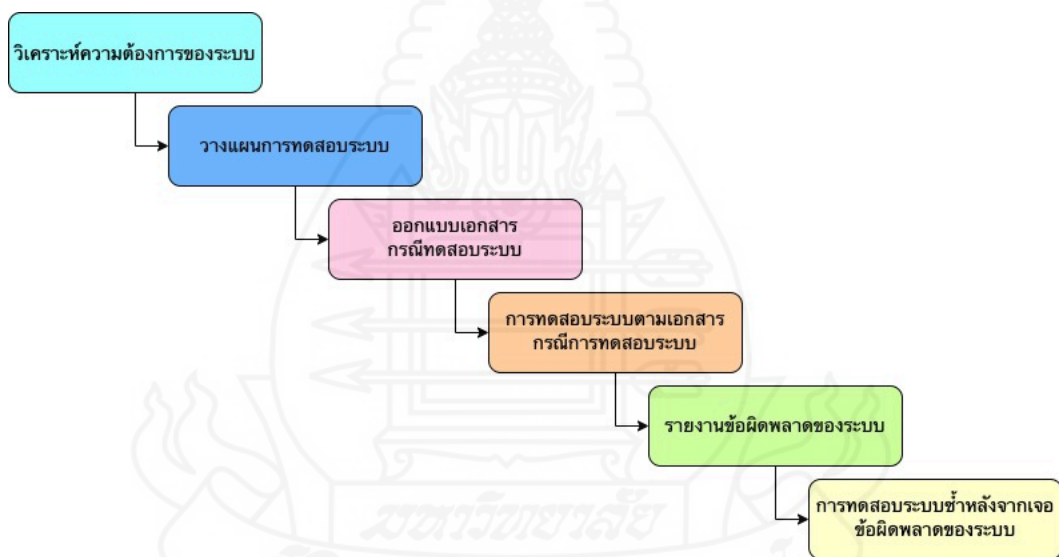
1 การทดสอบแบบกล่องดำ (Black Box Testing) คือการทดสอบระบบแบบไม่เห็นโครงสร้าง โดยจะเน้นส่วนของข้อมูลนำเข้า (Input) และ ผลลัพธ์ (Output) เท่านั้น

2 การทดสอบแบบกล่องขาว (White Box Testing) คือการทดสอบแบบเจาะระบบภายในเป็นการทดสอบระดับหน่วยของซอฟต์แวร์

3 การทดสอบแบบกล่องเทา (Gray Box Testing) เป็นวิธีการทดสอบแบบผสมผสานระหว่าง การทดสอบแบบกล่องดำ และ การทดสอบแบบกล่องขาว คือการทดสอบระบบแบบสนใจทั้ง ข้อมูลนำเข้า กระบวนการ และ ผลลัพธ์

2.1.1 วิธีการทดสอบระบบแบบมือ

(Naznin, 2018) ได้ให้ความหมายการทดสอบระบบแบบมือ คือการจำลองหรือออกแบบกรณีทดสอบ เพื่อยืนยันความถูกต้องของระบบ ผู้ทดสอบซอฟต์แวร์ ทำการทดสอบซอฟต์แวร์/เปรียบเทียบผลการทดสอบ โดยผู้ทดสอบจะทำการทดสอบตามกรณีทดสอบ โดยผู้ทดสอบเอง โดยอ้างอิงความถูกต้องของการทำงานของระบบจากเอกสารความต้องการของลูกค้า โดยแบ่งขั้นตอนการทดสอบระบบแบบมือ ออกเป็น 6 ขั้นตอน ดังภาพที่ 2.2



ภาพที่ 2.2 ขั้นตอนการทดสอบระบบแบบมือ (Manual Testing)

1) การวิเคราะห์ความต้องการของผู้ใช้ระบบ (Requirement Analysis) การวิเคราะห์ความต้องการของผู้ใช้ระบบ เป็นขั้นตอนการสำคัญของการทดสอบระบบโดยเป็นกระบวนการตรวจสอบข้อมูลขั้นพื้นฐานในการทดสอบในแต่ละกรณี รวมไปถึงการออกแบบข้อกำหนดการทดสอบ และ ความเสี่ยงต่าง ๆ ที่จะมีโอกาสเกิดขึ้นในระบบ

2) การวางแผนการทดสอบระบบ (Test Plan Creation) การวางแผนการทดสอบระบบเป็นขั้นตอนการออกแบบแนวทางรวมถึงวัตถุประสงค์ของการทดสอบ โดยอาศัยการวิเคราะห์ความต้องการของผู้ใช้ระบบ ในการเขียนกรณีทดสอบรวมถึงการกำหนด

สภาพแวดล้อมการทดสอบ และระบุเกณฑ์ความสำเร็จว่าต้องอยู่ที่ระดับใดในกรณีที่ทำกรทดสอบเสร็จสมบูรณ์

3) การออกแบบเอกสารกรณีทดสอบระบบ (Test Case Creation) เป็นการพัฒนาและจัดลำดับความสำคัญในแต่ละกรณีทดสอบ โดยใช้การสร้างข้อมูลการทดสอบในการทดสอบตามกรณีทดสอบต่าง ๆ

4) การทดสอบระบบตามเอกสารกรณีการทดสอบระบบ (Test Case Execution) คือการดำเนินทดสอบระบบตามกรณีทดสอบที่พัฒนาไว้ โดยใช้ผู้ทดสอบระบบเป็นผู้ที่ระบุและเก็บผลการทดสอบว่าแต่ละกรณีเป็นไปตามความต้องการของผู้ใช้ระบบหรือไม่ ซึ่งการทำงานแบบบอจี้เป็นการทำงานที่ต้องอาศัยความรวดเร็วและถูกต้องแม่นยำ ผู้จัดทำวิจัยจึงได้นำการทดสอบแบบ โดยใช้โรบอทเฟรมเวิร์ค มาใช้ในการทดสอบ เพื่อเป็นการลดเวลาและความผิดพลาดที่จะเกิดขึ้นจากมนุษย์ โดยทดสอบแบบรวมหน่วย (SIT : System Integration Test) และเพื่อเป็นการส่งมอบซอฟต์แวร์ให้กับผู้ใช้ (End User) อย่างมีคุณภาพและรวดเร็ว โดยลดข้อผิดพลาดในกระบวนการทดสอบการใช้งานโดยผู้ใช้งานจริง (UAT : User Acceptance Test)

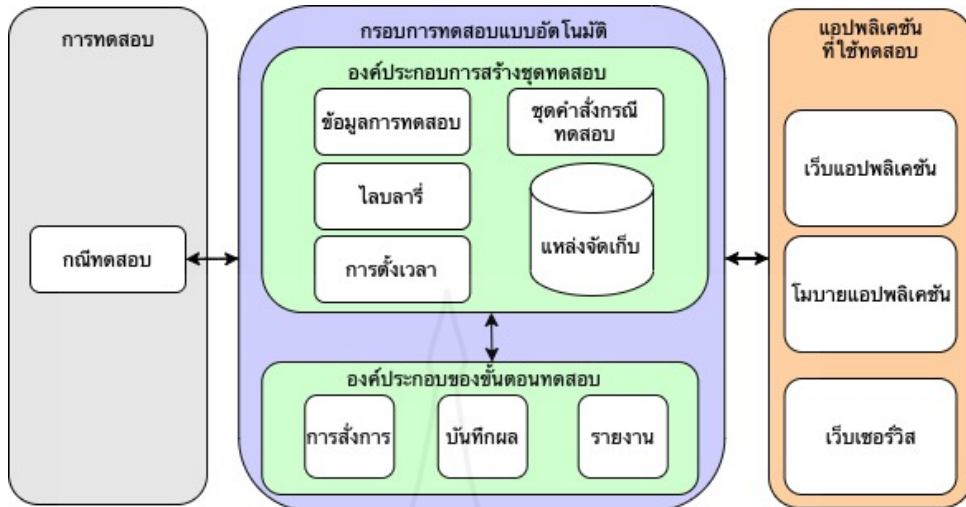
5) การรายงานข้อผิดพลาดของระบบ (Defect Logging) คือการบันทึกข้อผิดพลาดของระบบที่ผู้ทดสอบระบบพบข้อบกพร่องระหว่างการทดสอบ โดยการรายงานข้อผิดพลาดนี้จะต้องส่งไปให้ผู้ทดสอบระบบเป็นผู้แก้ไขให้เป็นไปตามความต้องการของผู้ใช้ระบบ

6) การทดสอบระบบซ้ำหลังจากเจอข้อผิดพลาดของระบบ (Re-Verification) คือการทดสอบกรณีทดสอบที่ผิดพลาดซ้ำหลังจากที่ ผู้พัฒนาระบบได้ทำการแก้ไขข้อผิดพลาดจากการรายงานข้อผิดพลาด เพื่อพิสูจน์ว่ากรณีทดสอบนั้นถูกต้องตามความต้องการของผู้ใช้ระบบ

2.1.2 การทดสอบระบบแบบอัตโนมัติ (Automation Testing)

คือการสร้างชุดกรณีทดสอบโดยสั่งงานผ่านซอฟต์แวร์ โดยนิยมนำมาใช้ในการทดสอบเชิงถดถอย เพื่อลดเวลาและข้อผิดพลาดจากการทดสอบโดยคน ในการทดสอบในกรณีที่ระบบมีขนาดใหญ่ และสามารถนำชุดทดสอบมาใช้ซ้ำได้ตามต้องการเนื่องจากเป็นโปรแกรมที่ทำงานด้วยคอมพิวเตอร์ การทำงานแต่ละครั้งจึงไม่แตกต่างกัน (หาญวรงค์, 2013)

2.1.2.1 กรอบการทดสอบระบบแบบอัตโนมัติ (Test Automation Framework) เป็นโครงสร้างที่ใช้ในการกำหนดแนวทางในการพัฒนาและบำรุงรักษากรณีทดสอบแบบอัตโนมัติ โดยกรอบการทดสอบระบบแบบอัตโนมัติ คือเครื่องมือการทำงานเครื่องมือหนึ่งที่ใช้ในการสร้างหรือพัฒนากรณีทดสอบแบบอัตโนมัติ โดยภาพแบบการใช้ขึ้นอยู่กับการใช้งานตามลักษณะของ แอปพลิเคชัน การทำวิจัยครั้งนี้ได้นำโรบอทเฟรมเวิร์คมาทำการวิจัยซึ่งโรบอทเฟรมเวิร์คเป็น กรอบการทดสอบที่สามารถทำงานได้บน เว็บแอปพลิเคชัน โมบายแอปพลิเคชัน และ เว็บเซอร์วิส ดังภาพที่



ภาพที่ 2.3 โครงสร้างของ Automation framework

การทดสอบ (Test Management) เป็นกระบวนการสร้างเอกสารกรณีทดสอบต่าง ๆ ตามเอกสารความต้องการของลูกค้า เพื่อทดสอบว่าผลลัพธ์เป็นไปตามที่คาดหวัง

กรอบการทดสอบแบบอัตโนมัติ (Test Generating Layer) เป็นกระบวนการที่จะสร้างกรณีทดสอบแบบอัตโนมัติขึ้นมาเพื่อใช้ในการทดสอบ ซึ่งการสร้างกรณีทดสอบแบบอัตโนมัตินั้นต้องประกอบไปด้วย ข้อมูลการทดสอบ (Test Data) เป็นข้อมูลที่ถูกสร้างขึ้นเพื่อเป็นข้อมูลนำเข้าสำหรับการทดสอบระบบ ชุดคำสั่งกรณีทดสอบ (Test Script) คือการพัฒนากรณีทดสอบให้อยู่ในภาพแบบของการเขียนโปรแกรม เพื่อให้สามารถทำการทดสอบได้ตามภาษาโปรแกรมนั้น ๆ รองรับ และมีเทสต์ไลบรารี (Test Library) เป็นตัวเพิ่มความสามารถของกรณีทดสอบแบบอัตโนมัติ ซึ่งการทำกรณีทดสอบแบบอัตโนมัติในแต่ละครั้งนั้นควรจะต้องมีแหล่งจัดเก็บ (Object Repository) เพื่อใช้ในการเก็บกรณีทดสอบแบบอัตโนมัติ เมื่อกรณีทดสอบแบบอัตโนมัติที่ครอบคลุมทุกกรณีทดสอบแล้วสิ่งต่อมาคือการ ทดสอบแบบอัตโนมัติ (Test Execution Layer) ซึ่งการรันทดสอบแบบอัตโนมัตินั้นไม่ว่าจะรันทดสอบด้วยสภาพแวดล้อม (Environment) หรือ ระบบใด และสามารถเก็บผลการทดสอบและสามารถออกรายงานผลการทดสอบได้

2.1.2.2 วัตถุประสงค์ในการทำการทดสอบแบบอัตโนมัติในทางธุรกิจ วัตถุประสงค์หรือเป้าหมายทางธุรกิจในการทำการทดสอบแบบอัตโนมัตินั้น เพื่อลดเวลาในการทำการทดสอบแบบถดถอย (Regression Test) ที่ใช้เวลาในมากในการทดสอบและช่วยลดอัตราความผิดพลาดของผู้พัฒนาระบบ โดยเฉพาะซอฟต์แวร์ที่เป็นระบบใหญ่ และช่วยประหยัดค่าใช้จ่ายให้กับองค์กรจากการจ้างคนเพิ่ม

2.1.3. ประเภทของการทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์แบ่งออกเป็น 6 ประเภท ได้แก่

2.1.3.1 การทดสอบแบบโมดูลหรือทดสอบหน่วย (Unit Test) เป็นระยะแรกของการทดสอบระบบหลังจากที่ผู้พัฒนาระบบได้เขียนคำสั่งโปรแกรมตามที่ได้รับมอบหมายตามเอกสารความต้องการของลูกค้าเสร็จ ระยะนี้เป็นการค้นหาข้อผิดพลาดของหน่วยโปรแกรม เช่น การทดสอบภาพแบบภาษาหรือไวยากรณ์ การทดสอบสูตรคำนวณ เป็นต้น

2.1.3.2 การทดสอบแบบรวม (Integration Test) เป็นการทดสอบรวมการทำงานของโมดูลโปรแกรมทั้งหมด โดยการนำโมดูลโปรแกรมทั้งหมดมารวมกันแล้วใช้วิธีการเรียงลำดับการทดสอบเป็นลำดับขั้นเพื่อให้สอดคล้องกับการออกแบบเป็นโมดูล

2.1.3.3 การทดสอบฟังก์ชัน (Function Test) เป็นการทดสอบที่มุ่งไปที่การทำงานตามเอกสารระบุความต้องการของลูกค้า โดยยึดการตรวจสอบจากเอกสารระบุความต้องการของลูกค้าเทียบกับระบบที่พัฒนาขึ้น ในระยะนี้ทีมทดสอบต้องมีความรู้ถึงหน้าที่และกิจกรรมต่าง ๆ ที่ต้องการให้ระบบสามารถกระทำได้

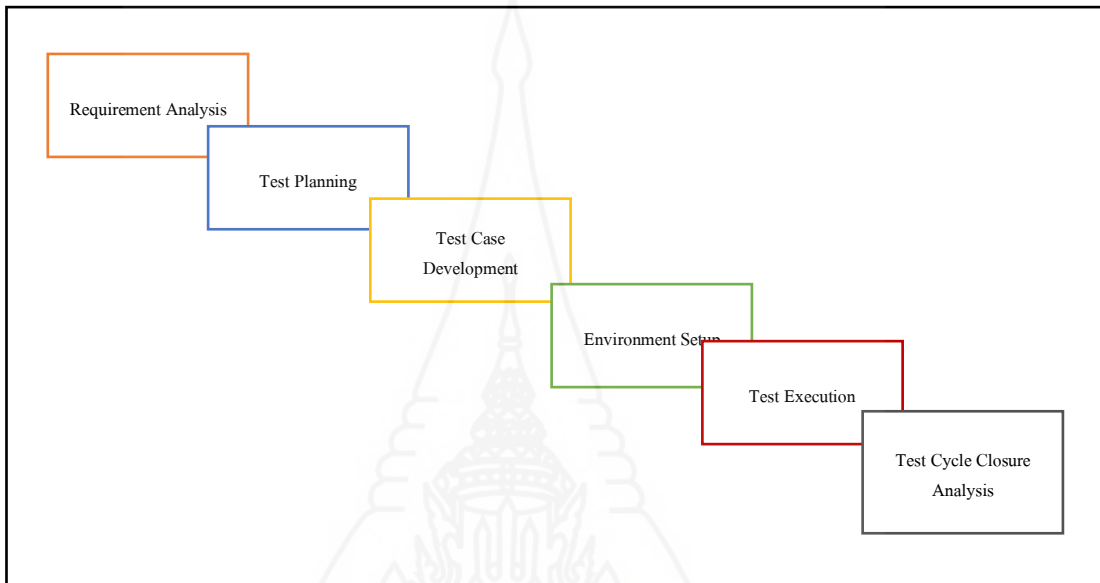
2.1.3.4 การทดสอบประสิทธิภาพ (Performance Test) เป็นการเปรียบเทียบโมดูลที่ผ่านการทดสอบรวม โดยการทดสอบจากสภาพแวดล้อมของผู้ใช้งานจริง เรียกว่า ระบบความตรวจสอบความสมเหตุสมผล (Validated System) หรือทดสอบในสภาพแวดล้อมจำลอง เรียกว่า ระบบทวนสอบ (Verified System) อ้างอิงจากวัตถุประสงค์ของลูกค้าบนพื้นฐานของความต้องการ ได้แก่ การทดสอบความตึงเครียด (Stress Test) เป็นการทดสอบความสามารถของระบบเมื่อเกิดความตึงเครียดในช่วงเวลาสั้น ๆ การทดสอบปริมาณ (Volume Test) เป็นการทดสอบปริมาณของข้อมูลที่มากที่สุดที่ระบบจะสามารถจัดการได้ การทดสอบแบบถดถอย (Regression Test) เป็นการทดสอบที่จำเป็นสำหรับการนำระบบใหม่แทนที่ระบบงานที่กำลังปฏิบัติการอยู่ เพื่อรับประกันว่าระบบใหม่มีประสิทธิภาพที่ดีกว่าและไม่มีผลกระทบเสียหายต่อระบบที่กำลังปฏิบัติการอยู่ การทดสอบความปลอดภัย (Security Test) เป็นการทดสอบความปลอดภัยของระบบ โดยตรวจสอบการเข้าถึงฟังก์ชันการทำงานการเข้าถึงข้อมูลของผู้ใช้ระดับต่าง ๆ

2.1.3.5 การทดสอบการยอมรับ (Acceptance Test) การทดสอบระยะนี้เป็นการทดสอบเปรียบเทียบระหว่างโปรแกรมกับเอกสารระบุความต้องการของลูกค้า เพื่อให้แน่ใจว่าระบบสามารถทำงานได้ตรงตามความต้องการที่กำหนดไว้ โดยระยะทดสอบนี้ จะมีผู้ทดสอบระบบก็คือผู้ที่ให้เอกสารระบุความต้องการมาทำการทดสอบ เพื่อวัดประสิทธิภาพและความถูกต้องตามความต้องการ ก่อนที่จะนำไปสู่การทดสอบการติดตั้งระบบบนสภาพแวดล้อมจริง

2.1.3.6 การทดสอบติดตั้ง (Installation Test) เป็นการทดสอบเพื่อให้แน่ใจว่าระบบสามารถทำงานได้จริงบนสภาพแวดล้อมของผู้ใช้งานจริง เป็นขั้นตอนสุดท้ายก่อนที่จะให้ผู้ใช้งานจริงหรือลูกค้าได้ใช้งานระบบ (Bundit, 2015)

2.1.4 วงจรชีวิตการทดสอบซอฟต์แวร์ (STLC : Software Testing Life Cycle)

เป็นกระบวนการที่ใช้ในการพัฒนาซอฟต์แวร์ โดยมีข้อกำหนดต่าง ๆ เพื่อสามารถพัฒนาและส่งมอบซอฟต์แวร์ได้อย่างมีประสิทธิภาพ โดยการทดสอบระบบจะแตกต่างกันในแต่ละโครงการ ซึ่งกระบวนการที่ใช้ในการพัฒนาซอฟต์แวร์ หรือ วงจรชีวิตการทดสอบซอฟต์แวร์ (STLC : Software Testing Life Cycle) ประกอบไปด้วยขั้นตอนหลัก ๆ ดังต่อไปนี้ (ศิริรังษี, 2017)



ภาพที่ 2.4 วงจรชีวิตการทดสอบซอฟต์แวร์ (STLC : Software Testing Life Cycle)

2.1.4.1 การวิเคราะห์ความต้องการของระบบ (Requirement Analysis) เป็นขั้นตอนแรกในวงจรชีวิตการทดสอบ โดยเน้นความเข้าใจของความต้องการของระบบจากมุมมองของนักทดสอบ โดยขั้นตอนนี้จะเป็นการประสานงานติดต่อกันกับนักวิเคราะห์ระบบและทีมพัฒนาระบบ เพื่อให้เกิดความเข้าใจในความต้องการของระบบได้อย่างสมบูรณ์ที่สุด ทั้งในส่วนของความต้องการทางด้านการทำงานของฟังก์ชันที่ต้องการและไม่ต้องการของระบบ ในขั้นตอนนี้นักทดสอบระบบจะต้องวิเคราะห์ความต้องการของระบบเพื่อให้แน่ใจว่าความต้องการของระบบสามารถทำการทดสอบได้

2.1.4.2 การวางแผนการทดสอบ (Test Planning) เป็นขั้นตอนการสร้างเอกสารที่เรียกว่า แผนการทดสอบ เพื่อใช้ในการแสดงรายละเอียดของการทดสอบระบบ ทุก ๆ การทำงานที่เกี่ยวข้อง เช่น คุณสมบัติที่ต้องการทดสอบ วิธีการทดสอบที่เลือกใช้ สภาพแวดล้อมของการทดสอบ เทคนิคและวิธีการทดสอบ และการกำหนดปัจจัยเสี่ยงต่อการทดสอบ

2.1.4.3 การสร้างกรณีทดสอบ (Test Case Development) เป็นขั้นตอนการสร้างและตรวจสอบความถูกต้องของกรณีทดสอบและสคริปต์สำหรับการทดสอบ รวมไปถึงข้อมูลที่ใช้ในการทดสอบ เทคนิคการทดสอบที่ถูกเลือกใช้จะถูกนำมาใช้ร่วมกันกับการออกแบบกรณีทดสอบ

โดยนักทดสอบระบบจะต้องเขียนกรณีทดสอบที่สามารถอ้างอิงจากเทคนิคดังกล่าว เช่น การทดสอบแบบอัตโนมัติ นักทดสอบจำเป็นจะต้องสร้างสคริปต์สำหรับการทดสอบ ซึ่งสคริปต์สำหรับการทดสอบจะแตกต่างกันไปตามเครื่องมือที่ใช้ เช่น การใช้โรบอทเฟรมเวิร์คในการเขียนสคริปต์การทดสอบ ใช้สคริปต์ที่เขียนมาจากภาษาไพธอน (Python) เป็นต้น

2.1.4.4 การกำหนดสภาพแวดล้อมของการทดสอบ (Test Environment Setting) สภาพแวดล้อมของการทดสอบระบบเป็นการติดตั้งซอฟต์แวร์และฮาร์ดแวร์ที่ต้องการใช้ร่วมกับโปรแกรมภายใต้การทดสอบ ซึ่งประกอบไปด้วยการติดตั้งระบบปฏิบัติการทั้งไคลเอนท์และเซิร์ฟเวอร์ (Server) รวมไปถึงระบบฐานข้อมูล ตลอดจนสิ่งจำเป็นอื่น ๆ ที่จำเป็นสำหรับการรันโปรแกรมทดสอบ

2.1.4.5 การประมวลผลการทดสอบ (Test Execution) หลังจากขั้นตอนการทดสอบแบบทีละโมดูลที่ถูกดำเนินการโดยนักพัฒนาเสร็จเรียบร้อยแล้ว กรณีทดสอบซึ่งถูกจัดเตรียมไว้ในขั้นตอนก่อนหน้าจะถูกนำมาประมวลผลตามเทคนิคการทดสอบต่าง ๆ ที่ถูกกำหนดไว้ ไม่ว่าจะเป็นการทดสอบด้วยมือหรือการทดสอบอัตโนมัติ ในกรณีผลการทดสอบไม่ผ่านตามข้อกำหนดจะถูกบันทึกไว้ในรายงานผลการทดสอบ เพื่อส่งกลับไปให้ทีมพัฒนาทำการแก้ไข หลังจากนั้นจะถูกนำส่งมาให้ทดสอบอีกครั้ง เพื่อให้แน่ใจว่าข้อผิดพลาดหรือข้อบกพร่องได้ถูกแก้ไขแล้ว

2.1.4.6 การสิ้นสุดวงจรทดสอบ (Test Cycle Closure) เมื่อนักทดสอบแน่ใจว่าการรายงานผลการทดสอบได้ถูกแก้ไขเรียบร้อยแล้วตามความต้องการของระบบที่กำหนดไว้ วงจรชีวิตของการทดสอบระบบจะเข้าสู่ขั้นตอนการปิดการทดสอบ ซึ่งในขั้นตอนนี้เป็นการประเมินผลการทดสอบโดยจัดการประชุมเพื่อระดมความเห็นและวิเคราะห์ผลลัพธ์จากการทดสอบที่ผ่านมา เพื่อให้ให้นักทดสอบระบบทำการทดสอบระบบได้มีประสิทธิภาพมากขึ้นในการทดสอบระบบโครงการต่อไปในอนาคต

2.2 การทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์

อไจล์ คือ กรอบแนวคิด และวิธีการทำงานอย่างคล่องแคล่วรวดเร็ว เพื่อปรับเปลี่ยนให้สามารถตอบสนองความต้องการของลูกค้าได้อย่างรวดเร็วในสถานะที่มีความไม่แน่นอน โดยมีกำหนดประกาศหรือ Manifesto ที่สำคัญ 4 ข้อ และมีหลักการ (Principle) 12 ข้อ ดังนี้ (สุขสว่าง, 2020)

2.2.1 คำประกาศ หรือ Manifesto ที่สำคัญ 4 ข้อ ของ Agile

3.1.1 การให้ความสำคัญกับตัวผู้ทำงานและการมีปฏิสัมพันธ์ระหว่างผู้ทำงาน มากกว่าขั้นตอนวิธีการหรือเครื่องมือ

3.1.2 การสร้างซอฟต์แวร์ (หรือผลิตภัณฑ์/บริการ/กระบวนการทำงาน) ที่สามารถนำไปใช้งานได้จริง มากกว่าเอกสาร คู่มือ หรือการใช้เวลาในการวางแผนเอกสารงาน

3.1.3 การทำงานร่วมกับลูกค้า มากกว่าการต่อรองสัญญาับลูกค้า

3.1.4 การยอมรับปรับเปลี่ยนซอฟต์แวร์ (หรือผลิตภัณฑ์/บริการ/กระบวนการทำงาน) ตามความเปลี่ยนแปลงของความต้องการของลูกค้า มากกว่าการทำตามแผนการเพียงอย่างเดียว

2.2.2 หลักการ 12 ข้อ

3.2.1 ความสำคัญสูงสุดคือความพึงพอใจของลูกค้าผ่านการส่งมอบซอฟต์แวร์

3.2.2 ยินดีปรับเปลี่ยนเพื่อแก้ไขงานตามความต้องการของลูกค้า แม้ว่าจะอยู่ในขั้นตอนท้ายๆของการพัฒนาแล้วก็ตาม กระบวนการเปลี่ยนแปลงนี้เพื่อให้ลูกค้าสามารถแข่งขันได้

3.2.3 การส่งมอบซอฟต์แวร์ (หรือผลิตภัณฑ์/บริการ/กระบวนการทำงาน) อย่างสม่ำเสมอ ด้วยการตั้งค่าเพื่อลดระยะเวลาการทำงานให้สั้นลง

3.2.4 ลูกค้าหรือผู้ใช้งานต้องทำงานร่วมกันอย่างสม่ำเสมอ เพื่อให้สามารถพัฒนาซอฟต์แวร์ (หรือผลิตภัณฑ์/บริการ/กระบวนการทำงาน) ที่ตรงใจลูกค้าได้

3.2.5 การพัฒนาโปรเจกต์โดยรวมคนทำงานที่มีแรงจูงใจ และกระตือรือร้นในการทำงาน โดยสร้างสภาพแวดล้อมและการสนับสนุนที่พวกเขาต้องการ และให้ความไว้วางใจทีมงานว่าจะสามารถพัฒนางานที่ได้ออกมาได้

3.2.6 วิธีการที่มีประสิทธิภาพและประสิทธิผลนั้น ให้มีการแลกเปลี่ยนข้อมูล ระหว่างทีมพัฒนาด้วยกันเอง หรือกับลูกค้า ด้วยการสื่อสารแบบตัวต่อตัวเห็นหน้า เจอตัวกันมากกว่า การคุยหรือสื่อสารผ่านเครื่องมือ

3.2.7 การสร้างหรือพัฒนาซอฟต์แวร์ (หรือผลิตภัณฑ์/บริการ/กระบวนการทำงาน) ที่มีคุณค่า เป็นตัววัดความก้าวหน้าของการทำงาน

3.2.8 กระบวนการที่คล่องตัว หรือ อไจล์ จะเป็นการส่งเสริมการพัฒนาที่ยั่งยืน ดังนั้นผู้สนับสนุน นักพัฒนา และกลุ่มผู้ใช้ จะต้องทำงานด้วยระดับความเร็วที่สม่ำเสมอ ไม่ช้าเกินไป หรือ ไปเร่งงานช่วงท้ายของการพัฒนา

3.2.9 การพัฒนาความรู้เชิงเทคนิคให้ดีเยี่ยมอย่างต่อเนื่อง และการออกแบบที่ดี เมื่อมีทั้งสองสิ่งนี้จะทำให้เกิดการคล่องตัวในการพัฒนาผลิตภัณฑ์ได้อย่างรวดเร็ว

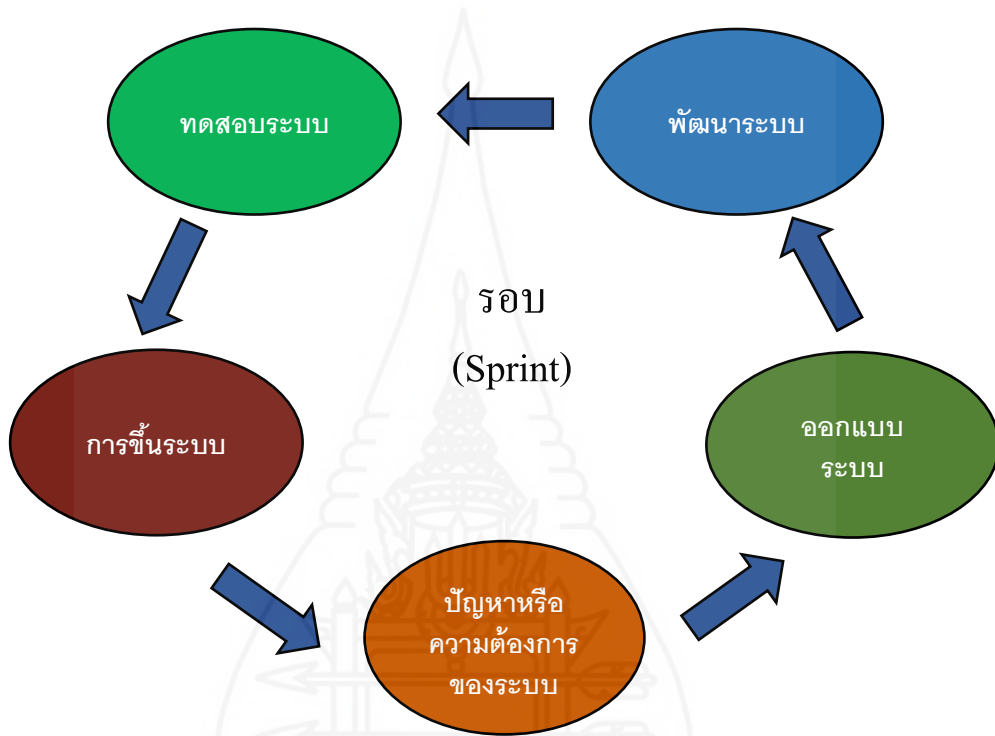
3.2.10 ความเรียบง่าย เป็นศิลปะในการทำงานที่มีความพิเศษ ไม่ใช่จำนวนสูงสุดของจำนวนงานที่ทำ ซึ่งในการทำงานนั้น พยายามทำงาน หรือพัฒนาผลิตภัณฑ์ให้เรียบง่าย ไม่ซับซ้อน แต่ยังมีประสิทธิภาพการใช้งานได้สูงสุดดีกว่า

3.2.11 ภาพแบบของงาน ความต้องการของลูกค้า และการออกแบบที่ดี จะมาจากที่คนในทีมมีการรับผิดชอบในงานของตัวเองอย่างดีที่สุด สามารถคิดและหาวิธีการทำงานที่มีประสิทธิภาพ และแก้ปัญหาได้โดยไม่ต้องรอให้ใครมาบริหารจัดการ นอกจากจัดการตัวเองให้พร้อม และพัฒนาตัวเองอยู่เสมอ

3.2.12 ในช่วงเวลาปกติ ทีมต้องมีการแลกเปลี่ยน หรือให้ข้อมูลป้อนกลับ ทั้งให้กับตัวเอง และทีมงาน ซึ่งจะผ่านการให้ feedback กันในทีมก็ได้ เพื่อปรับเปลี่ยนพฤติกรรม แนวทางการทำงาน เพื่อให้เกิดการพัฒนาขึ้นอย่างต่อเนื่องและสม่ำเสมอ

2.2.3 วงจรชีวิตการทำงานแบบอไจล์ (Agile SDLC Model)

มีขั้นตอนและกระบวนการดังภาพที่ 2.5



ภาพที่ 2.5 วงจรชีวิตการทำงานแบบอไจล์

1) ปัญหาหรือความต้องการของระบบ (Requirement) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการรับปัญหาหรือความต้องการของระบบ คือ ผู้ใช้งาน สรุปได้ว่า ปัญหาหรือความต้องการของระบบได้มาจากคนที่ผู้ใช้งาน นั้นมีความต้องการหรือพบปัญหาเกี่ยวกับระบบที่ใช้งานอยู่และต้องการให้มีการแก้ไขหรือเพิ่มฟีเจอร์ (Feature) ให้กับระบบ ผู้ใช้งานจะให้เอกสารปัญหาหรือความต้องการมาในภาพแบบยูสเซอร์ สตอรี (User Story) ซึ่งทีมงานอไจล์ จะต้องรับปัญหาหรือความต้องการไปสร้างเป็นงานตามลักษณะงานของตัวเอง

2) ออกแบบระบบ (Design) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการออกแบบระบบ คือ ผู้วิเคราะห์และออกแบบระบบ เป็นผู้รับปัญหาหรือความต้องการจากผู้ใช้งานและมาทำการวิเคราะห์และออกแบบแนวทางการพัฒนาซอฟต์แวร์เพื่อให้ทีมพัฒนาระบบนำไปพัฒนาให้ตรงกับความต้องการของผู้ใช้งาน

3) พัฒนาระบบ (Development) จากการที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการพัฒนาระบบคือ ผู้พัฒนาระบบเป็นผู้ที่ทำหน้าที่พัฒนาซอฟต์แวร์

ตามที่คุณวิเคราะห์และออกแบบระบบได้ออกแบบไว้โดยสอดคล้องกับปัญหาหรือความต้องการของระบบของผู้ใช้งานให้สำเร็จถึงจะนำซอฟต์แวร์ที่เสร็จสมบูรณ์ส่งให้ทีมทดสอบระบบทำการตรวจสอบ

4) ทดสอบระบบ (Software Testing) เป็นการตรวจสอบความถูกต้องของซอฟต์แวร์หลังจากที่ผู้พัฒนาระบบทำการพัฒนาเสร็จสมบูรณ์ ซึ่งการทดสอบระบบนั้นต้องใช้เวลาและความละเอียดรอบคอบ ซึ่งการทำงานแบบอไจล์ เป็นการทำงานที่ต้องอาศัยความรวดเร็วและถูกต้องแม่นยำ

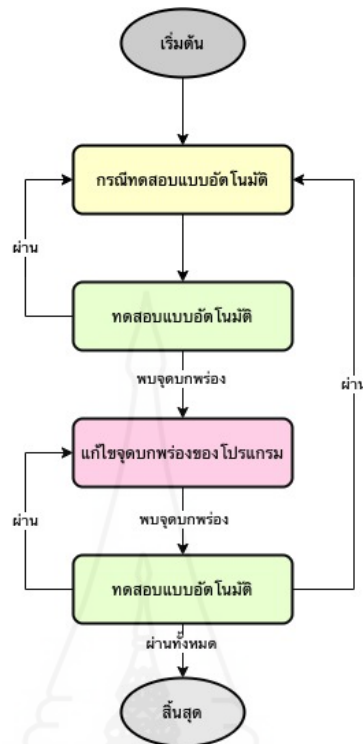
5) การขึ้นระบบ (Deployment) จากการทำที่ผู้วิจัยได้ทำการสัมภาษณ์บุคลากรที่ทำหน้าที่เกี่ยวกับการการขึ้นระบบ คือ ผู้พัฒนาระบบ ทำหน้าที่นำซอฟต์แวร์ที่ผ่านกระบวนการทดสอบเรียบร้อยแล้วขึ้นไปให้ผู้ใช้งานจริงได้ใช้งาน

2.2.3 การทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์

Rijwan Khan (2016) ได้กล่าวว่า การทดสอบระบบที่พัฒนาขึ้นโดยวิธีการทำงานแบบ Agile นั้นไม่ใช้การแยกการทำงานระหว่างทีมพัฒนาและทีมทดสอบ แต่เป็นการร่วมมือกันทำงานเพื่อสร้างคุณภาพของระบบ โดยทีมอไจล์ จะขาดทีมทดสอบระบบไปไม่ได้ เพราะทีมทดสอบระบบเป็นทีมที่มีความเชี่ยวชาญในการสร้างกรณีทดสอบต่าง ๆ ที่ตรงตามพฤติกรรมของลูกค้านั้น ทีมทดสอบระบบสามารถทำให้ระบบที่พัฒนามีคุณค่าและความน่าเชื่อถือมากขึ้นโดยกระบวนการทดสอบระบบในการทำงานแบบ Agile นั้น คือขั้นตอนการหาข้อผิดพลาดของระบบและทำการทดสอบยืนยันว่าระบบไม่มีข้อผิดพลาดหรือมีข้อบกพร่องน้อยที่สุด

Sandeep Sivanandan (2018) ได้กล่าวว่า การทดสอบแบบอัตโนมัติเป็นวิธีที่มีประสิทธิภาพในการทดสอบระบบเมื่อนำไปใช้กับเฟรมเวิร์กจะมีประสิทธิภาพสูงสุด และเป็นวิธีการทดสอบที่นิยมมากที่สุดในการทำงานแบบอไจล์ ซึ่งสามารถทดสอบได้ตรงตามพฤติกรรมของลูกค้านั้นแบบที่ละขั้นตอนอย่างละเอียดโดยแต่ละโครงการอาจใช้กระบวนการพัฒนาซอฟต์แวร์ที่ใช้ในการทำงานแบบอไจล์ แตกต่างกันไปคือ

1) การพัฒนาที่ขับเคลื่อนด้วยการทดสอบ (Test-Driven Development :TDD) คือ กระบวนการพัฒนาซอฟต์แวร์ ที่อาศัยหลักการทำซ้ำในวงจรพัฒนาที่สั้น ๆ โดยนักทดสอบระบบจะทำการเขียนกรณีทดสอบแบบอัตโนมัติเพื่อกำหนดคุณสมบัติของฟังก์ชัน (Function) ใหม่ที่ต้องการ และจากนั้นใช้วิธีการเขียนโค้ดให้น้อยที่สุดที่จะทำการทดสอบให้ผ่านแล้วจึงค่อยปรับปรุงโค้ดให้ได้มาตรฐาน เพื่อเป็นการหาช่องโหว่ของระบบและนำมาแก้ไขให้ถูกต้องมากที่สุด ดังภาพที่ 2.6



ภาพที่ 2.6 กระบวนการพัฒนาที่ขับเคลื่อนด้วยการทดสอบ

2) การพัฒนาที่ขับเคลื่อนด้วยพฤติกรรม (Behavioral-Driven Development) (BDD) คือการทดสอบระบบแบบหน่วยโดยคำนึงถึงผลลัพธ์เป็นไปตามความต้องการของลูกค้า โดยใช้วิธี Given-When-Then โดยมีตัวอย่างเช่น

Given – ผู้ใช้งานป้อนข้อมูลเข้าระบบได้ถูกต้อง

When – ผู้ใช้งานกดปุ่มเข้าสู่ระบบ

Then – เข้าสู่ระบบสำเร็จ

จะเห็นได้ว่าตัวอย่างข้างต้นของการพัฒนาที่ขับเคลื่อนด้วยพฤติกรรมนั้นใช้ภาษาที่เข้าใจง่ายซึ่งช่วยให้ทุกคนในทีมเข้าใจในการที่จะพัฒนาระบบนั้น ๆ

3) การพัฒนาที่ขับเคลื่อนด้วยการทดสอบการยอมรับ (Acceptance Test Driven Development : ATDD) เป็นอีกหนึ่งของการทำงานแบบอไจล์ที่ให้ความสำคัญของความต้องการของลูกค้า และกรณีทดสอบ เพื่อสร้างความเข้าใจที่ตรงกันระหว่างลูกค้ากับทีมทดสอบ โดยเป็นการทำงานแบบวนซ้ำ ซึ่งประกอบไปด้วย 3 ขั้นตอน คือ

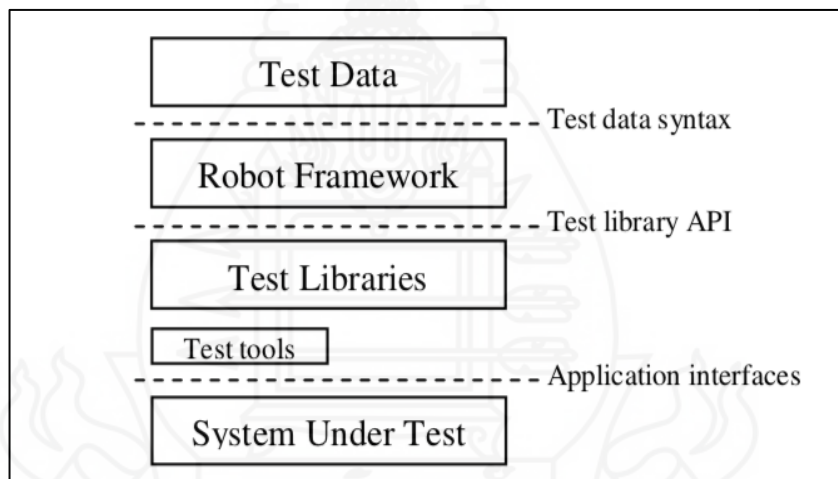
การอธิบาย (Discuss) เป็นขั้นตอนการปรึกษาปัญหาที่นำมาจาก Product Backlog เพื่อหา Solution โดยทีมทั้งหมดจะทำงานร่วมกันเพื่อหาวิธีแก้ปัญหาที่ดีที่สุด ช่วยให้ทีมเข้าใจความต้องการของลูกค้าไปในทางเดียวกัน รวมถึงมีการกำหนดตัวอย่างที่ชัดเจนเพื่อให้ทีมทดสอบระบบทำการสร้างกรณีทดสอบเพื่อใช้ในการทดสอบต่อไป

การพัฒนาซอฟต์แวร์ (Develop) เป็นขั้นตอนของการพัฒนากรณีทดสอบแบบอัตโนมัติ โดยนำข้อมูลมาจากการถกเถียงหารือ โดยทีมจำเป็นต้องร่วมกันพัฒนาทดสอบแบบอัตโนมัติขึ้นเพื่อนำไปทำการทดสอบ และเมื่อการทดสอบสำเร็จโดยไม่มีข้อผิดพลาดครบทุกกรณีแล้วนั้นจะถือว่าขั้นตอนนี้เสร็จสมบูรณ์

การส่งมอบ (Deliver) เป็นการส่งมอบงานที่ทำผ่านการทดสอบแล้วให้กับลูกค้า โดยมีรายงานผลการทดสอบเพื่อเป็นสิ่งที่ช่วยยืนยันว่า ระบบผ่านการทดสอบมาแล้วและตรงกับความต้องการของลูกค้าที่ได้ตกลงกันในช่วงการถกเถียงหารือ

2.3 ไรบอทเฟรมเวิร์ค (Robot Framework)

เป็นเฟรมเวิร์คสำหรับพัฒนากรณีทดสอบแบบอัตโนมัติที่ไม่มีค่าใช้จ่าย ใช้กระบวนการพัฒนาโดยภาษาที่เข้าใจง่ายไม่ซับซ้อนโดยใช้พื้นฐานภาษาไพธอน (Python) หรือ จาว่า (Java) สามารถพัฒนาไลบรารี เพื่อนำมาประยุกต์ใช้ได้ในแต่ละโครงการ โดยโครงสร้างของไรบอทเฟรมเวิร์ค สามารถแบ่งได้ดังภาพที่ 2.7



ภาพที่ 2.7 โครงสร้างของไรบอทเฟรมเวิร์ค (robotframework, 2021)

2.3.1 ข้อมูลทดสอบ (Test Data) หรือข้อมูลการทดสอบเป็นข้อมูลสำหรับให้ ไรบอทเฟรมเวิร์ค (Robot Framework) มาเรียกใช้ตามกรณีทดสอบและทำการบันทึกรายงาน โดยสามารถแบ่งกลุ่มของข้อมูลได้เป็น 3 ภาพแบบคือ

Scalar = \$ {}

เป็นภาพแบบการเก็บข้อมูลโดยกำหนดภาพแบบข้อมูลเป็นตัวอักษร เช่น

$\${data}$ = TestString

List = @{}

เป็นภาพแบบการเก็บข้อมูลแบบ List หรือ Array เช่น

@{list} = a,b,c,d,e,f

Dictionary = &{}

เป็นภาพแบบการเก็บข้อมูลแบบ Key-Value เช่น

@{dictionary} key1=value1 key2=value1

2.3.2 โรบอทเฟรมเวิร์ค (Robot Framework) คือเครื่องมือในการสร้างกรณีทดสอบแบบอัตโนมัติ โดยแบ่งโครงสร้างได้ดังนี้

1) กรณีทดสอบ (Test Case) เป็นไฟล์สำหรับกำหนดกรณีทดสอบโดยอ้างอิงจากเอกสารความต้องการของลูกค้า โดยมีโครงสร้างในการออกแบบกรณีทดสอบคือ

การตั้งค่า (Setting) ในส่วนของการตั้งค่า ใช้ในการ นำเข้าไฟล์ (Include File), การกำหนดคำอธิบายของไฟล์ (Document Description), การทำสืบทอด (Suite Setup), การทำทီးรด์าวน์ (Teardown) และ การนำเข้าไลบรารี (Import Library) ซึ่งการ การนำเข้าไลบรารีสามารถแบ่งไป 3 ส่วน นั่นคือ ไลบรารีมาตรฐานของโปรแกรม (Standard), ไลบรารีที่ติดตั้งจากภายนอก (External), ไลบรารีที่ผู้ใช้งานพัฒนาขึ้นเอง (Other) (robotframework, 2019)

วาเรียเบิล (Variables) คือการประกาศตัวแปรเก็บข้อมูลการทดสอบต่าง ๆ สำหรับการนำไปเขียนเทสต์สคริปต์

2) คีย์เวิร์ด (Keyword) คือสิ่งที่นำมาใช้แทนขอบเขตของเนื้อหาที่ผู้ใช้ต้องการ โดยคีย์เวิร์ดจะมีหน้าที่ในการทำงานคล้ายกับฟังก์ชัน (Function) หรือ คลาส (Class) ในหลาย ๆ ภาษา แต่สามารถเขียนขึ้นได้ง่ายกว่าโดยในคีย์เวิร์ดอาจจะมีตัวรับส่งค่าข้อมูลในโรบอทเฟรมเวิร์คที่เรียกว่า อาร์กิวเมนต์ (Arguments) ซึ่งจะรับส่งค่าข้อมูลอยู่ในภาพแบบของตัวแปร $\${Scalar}$, $@{List}$, $\&{Dictionary}$

3) รายงานและบันทึกผลการทดสอบ (Report and Log) คือรายงานผลการทดสอบโดยแสดงรายละเอียดของการทดสอบในครั้งนั้น ๆ ประกอบไปด้วย

การรายงานข้อมูลทดสอบรวม (Summary Information) เป็นการสรุปภาพรวมของการทดสอบว่า สถานะของการทดสอบทั้งหมดในครั้งนั้นมีผลเป็นอย่างไร เริ่มและสิ้นสุดรันการทดสอบตอนไหน และใช้เวลาในการรันการทดสอบเท่าไร

สถิติของการทดสอบ (Test Statistics) เป็นการรายงานจำนวนกรณีทดสอบที่ใช้รันทดสอบในรอบนั้น ๆ โดยจะระบุเป็นตัวเลขของจำนวนกรณีทดสอบ และรายงานผลการทดสอบเป็นระดับข้อ ระดับกลุ่มการทดสอบและระดับไฟล์การทดสอบ

การบันทึกผล (Log) เป็นการรายงานผลการทดสอบระดับข้อโดยมีการระบุรายละเอียดผลการทำงานของแต่ละคีย์เวิร์ดในข้อนั้น ๆ ซึ่งช่วยให้ผู้ทดสอบระบบสามารถหาสาเหตุได้ง่ายในกรณีที่ผลการทดสอบไม่ผ่าน

2.3.3 ไลบรารีทดสอบ (Test Library) เป็นปัจจัยสำคัญที่ทำให้ทำให้เกิดกรณีทดสอบอัตโนมัติขึ้นได้ ไลบรารีทดสอบเป็นสิ่งที่เพิ่มความสามารถในการทำงานของโรบอทเฟรมเวิร์ค ซึ่งทำหน้าที่เก็บรวบรวมคีย์เวิร์ดไว้โดยแยกตามจุดประสงค์ของการทดสอบในไฟล์นั้น ๆ โดย ไลบรารีทดสอบ มีทั้งที่เป็นของโรบอทเฟรมเวิร์คที่ทำการพัฒนาไว้ให้ผู้ใช้งานได้ใช้ไลบรารีทดสอบที่มาจากแหล่งอื่น ๆ รวมทั้งผู้ใช้เองสามารถพัฒนาไลบรารีทดสอบขึ้นมาเอง (Custom Test Library) ตามความต้องการ

2.3.4 ระบบที่ใช้ทดสอบ (System Under Test / Target System) เป็นระบบที่จะนำการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คโดยโรบอทเฟรมเวิร์คสามารถใช้ในการทดสอบได้กับหลายระบบ เช่น การทดสอบหน้าจอ (UI Testing) การทดสอบแอปพลิเคชันระบบปฏิบัติการ โทรศัพท์มือถือ (Mobile Testing) การทดสอบฐานข้อมูล (Database Testing) การทดสอบส่วนต่อประสานโปรแกรม (API Testing) โดยแต่ละระบบการทดสอบจำเป็นที่จะต้องใช้ไลบรารีทดสอบที่เหมาะสมกับระบบนั้น ๆ เพื่อให้เกิดประสิทธิภาพสูงสุดในการทดสอบ

2.4 เครื่องมือทางสถิติ

2.4.1 ค่ามาตรฐาน (Standard Value)

ค่ามาตรฐานเป็นค่าที่ใช้เปรียบเทียบค่าของข้อมูล ตั้งแต่ 2 ข้อมูลขึ้นไปที่มาจากข้อมูลคนละชุด เพื่อแสดงว่าข้อมูลใดมีคุณภาพดีกว่ากัน (รองรวม, 2557) ดังสมการที่ 2.1

$$Z_i = \frac{X_i - \bar{X}}{S} \quad (2.1)$$

เมื่อ Z_i แทนค่ามาตรฐานของข้อมูลตัวที่ i
 X_i แทนค่าของข้อมูลตัวที่ i
 \bar{X} แทนค่าเฉลี่ยเลขคณิตของข้อมูลชุดนั้น
 S แทนค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูลชุดนั้น

สมบัติของค่ามาตรฐาน

1. ค่ามาตรฐานไม่มีหน่วย
2. ผลรวมของค่ามาตรฐานชุดใดชุดหนึ่งจะมีค่าเท่ากับ 0 เสมอ
3. ค่าเฉลี่ยเลขคณิตของค่ามาตรฐานเท่ากับ 0
4. ส่วนเบี่ยงเบนมาตรฐานของค่ามาตรฐานเท่ากับ 1
5. ค่ามาตรฐานของข้อมูลอาจเป็นค่าบวกหรือลบก็ได้
6. ค่ามาตรฐานต้องมีค่าอยู่ระหว่าง +3 ถึง -3

7. ผลรวมของกำลังสองของค่ามาตรฐานเท่ากับจำนวนข้อมูล

2.4.2 เทคนิคการสุ่มซ้ำด้วยวิธีการบูทสเตรป (Bootstrapping) คือกระบวนการสุ่มตัวอย่างแบบซ้ำช่วยให้การประมาณค่าทางสถิติมีความแม่นยำขึ้น ซึ่งวิธีนี้จะถือว่ากลุ่มตัวอย่างที่เก็บมาเป็นประชากรเสมือน และทำการสุ่มตัวอย่างซ้ำ ๆ กันหลาย ๆ ครั้ง และใช้การประมาณค่าที่ได้มาสร้าง ความเชื่อมั่น โดยใช้โปรแกรมสำเร็จรูปในการวิเคราะห์ เมื่อกำหนดให้ θ เป็นพารามิเตอร์ของประชากรที่มีการแจกแจงแบบใด ๆ โดยสุ่มตัวอย่างขนาด n จากประชากรที่มีการแจกแจงแบบใด ๆ แทนด้วย $x_1^*, x_2^*, \dots, x_n^*$ และนำมาหาค่าเฉลี่ยความแปรปรวนจะได้ค่าความแปรปรวนของค่าประมาณแบบจุดของพารามิเตอร์ โดยการทำบูทสเตรป สามารถทำได้ 2 วิธีคือ การสุ่มตัวอย่างแบบไม่คืนที่ (Sampling without replacement) และ การสุ่มตัวอย่างแบบคืนที่ (Sampling with replacement)

2.4.3 การทดสอบหาค่าความเชื่อมั่น (Reliability)

เป็นคุณสมบัติว่าเมื่อใช้เครื่องมืออื่น ๆ ไปวัดผลในสิ่งเดียวกันจะได้ผลแน่นอน สม่ำเสมอ คงเส้นคงวา เป็นที่มั่นใจ หรือเชื่อถือในผลนั้นได้จริง ถึงแม้จะมีการวัดซ้ำอีก ผลที่ได้ก็จะไม่เปลี่ยนแปลงไปจากเดิม โดยการหาความเชื่อมั่น โดยใช้สูตรสัมประสิทธิ์อัลฟาของ ครอนบัก (Coefficient Alpha or Cronbach's Alpha) ดังสมการที่ 2.2

$$a = \left[\frac{k}{k-1} \right] \left[1 - \frac{\sum S_i^2}{S_t^2} \right] \quad (2.2)$$

เมื่อ a เป็นสัมประสิทธิ์แอลฟา

k เป็นจำนวนข้อคำถาม

S_i^2 เป็นความแปรปรวนของคะแนนข้อที่ i

S_t^2 เป็นความแปรปรวนของคะแนนข้อที่ t

2.4.4 การวิเคราะห์ปัจจัย (Factor Analysis)

การวิเคราะห์ปัจจัย เป็นเทคนิควิธีทางสถิติเพื่อใช้ในการจับกลุ่ม รวมกลุ่ม หรือรวมตัวแปรที่มีความสัมพันธ์กันไว้ในกลุ่มเดียวกัน โดยเป็นไปได้ทั้งทางบวกและลบ โดยตัวแปรในองค์ประกอบเดียวกันจะมีความสัมพันธ์กันสูง ส่วนตัวแปรต่างองค์ประกอบกันจะมีความสัมพันธ์กันน้อยหรือไม่มีความสัมพันธ์กัน โดยจะพิจารณาจากค่าของตัวแปรประจักษ์ในการจัดกลุ่มตัวแปรที่มีความสัมพันธ์กัน ดังนั้นตัวแปรประจักษ์ต่าง ๆ ควรมีการทำให้เป็นค่ามาตรฐานเสียก่อน เพื่อไม่ให้เกิดความผิดพลาดจากขนาดและหน่วยวัดของแต่ละตัวแปรประจักษ์ โดยขั้นตอนการวิจัยมีดังนี้

1) การสร้างเมทริกซ์ความสัมพันธ์ระหว่างคู่ของตัวแปร (Correlation Matrix) เป็นขั้นแรกตอนของการวิเคราะห์ปัจจัยที่ทำการหาความสัมพันธ์ในภาพแบบของเส้นตรงโดยวิธีของ

Pearson Correlation ระหว่างตัวแปรทุกคู่ที่ต้องการมาจัดกลุ่ม Correlation Matrix เป็นการวิเคราะห์ที่ว่าตัวแปรสองตัวหรือชุดตัวแปรมีความผันแปรเกี่ยวเนื่องกันมากน้อยเพียงใด เรียกค่าที่แสดงขนาดความสัมพันธ์ว่าค่าสัมประสิทธิ์สหสัมพันธ์ ซึ่งมีค่าอยู่ระหว่าง 0 ถึง ± 1 ค่าใกล้ 0 หมายถึงสัมพันธ์น้อย ค่าใกล้ 1 หมายถึงสัมพันธ์มาก ดังสมการที่ 2.3 (สุภมาศ อังสุโชติ, 2560)

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (2.3)$$

เมื่อ x_i, y_i คือค่าใด ๆ ของตัวแปรที่เป็นคู่กัน
 \bar{x}, \bar{y} คือค่าเฉลี่ยของตัวแปร

2) การสกัดปัจจัย (Factor Extraction) เป็นขั้นตอนการหาจำนวนองค์ประกอบที่สามารถใช้แทนตัวแปรทั้งหมดทุกตัวได้ วิธีการสกัดปัจจัยมีหลายวิธีโดยวิธีที่นิยมใช้ก็คือ วิธีองค์ประกอบหลัก (Principal Component Analysis : PCA) เป็นเทคนิคที่อาศัยหลักความสัมพันธ์เชิงเส้นระหว่างตัวแปรที่ใช้เป็นข้อมูลองค์ประกอบหลัก ด้วยการสร้างการรวมกันของตัวแปรเชิงเส้น (Line Combination) โดยองค์ประกอบที่ 1 จะเป็น Line Combination แรก และอธิบายความผันแปรได้มากที่สุดหรือมีความแปรปรวนสูงสุด องค์ประกอบที่ 2 จะเป็น Line Combination ของตัวแปรและอธิบายความผันแปรได้อันดับที่สอง โดยองค์ประกอบที่สองต้องตั้งฉาก (Orthogonal) กับองค์ประกอบที่ 1 หรือกล่าวได้ว่าองค์ประกอบทั้งสองไม่มีความสัมพันธ์กัน ดังสมการที่ 2.4 (ธีระดาภิญญา, 2561)

$$F_i = \sum_{j=1}^k W_{ij} \cdot X_j \quad (2.4)$$

โดยที่ F_i เป็นค่าประมาณการของปัจจัยที่ i ได้จากการผสมสมการเชิงเส้นของตัวแปรเดิม X_j ($j = 1, 2, \dots, k$)

W_{ij} เป็นค่าถ่วงน้ำหนักที่ปัจจัยหนึ่งเกิดจากตัวแปรประจักษ์ต่าง ๆ เรียกว่า Factor Weight

X_j เป็นค่าของตัวแปรเดิมที่ j ซึ่งเป็นสมาชิกของปัจจัย F_i โดยค่า X_j ควรเป็นค่าคะแนนมาตรฐาน

3) หมุนแกนแบบมุมฉาก (Varimax Rotation) การที่ค่าน้ำหนักปัจจัย (Factor Loading) ไม่ชัดเจน จึงไม่สามารถจัดตัวแปรได้ว่าควรจัดอยู่ในกลุ่มใดได้นั้น จะต้องอาศัยการหมุน

แกน ดังนั้นวัตถุประสงค์ของการหมุนแกนปัจจัยคือ เพื่อให้ค่าน้ำหนักปัจจัยของตัวแปร มีค่ามากขึ้นหรือลดลงจนกระทั่งทำให้ทราบได้ว่าตัวแปรนั้นควรอยู่ในกลุ่มใด หรือไม่ควรอยู่ในกลุ่มใดโดยในวิจัยครั้งนี้ผู้วิจัยได้ทำการหมุนแกนแบบมูมาจก (Varimax Rotation) เพื่อให้ได้องค์ประกอบที่เป็นอิสระต่อกัน เนื่องจากการหมุนแกนแบบจกเป็นการเปลี่ยนตำแหน่งของตัวแปรให้แกนตัวประกอบตั้งฉากซึ่งกันและกัน การที่องค์ประกอบทำมุม 90 องศาซึ่งกันและกัน แสดงความเป็นอิสระทางสถิติขององค์ประกอบเหล่านั้นและการหมุนแกนแบบจกทำให้เกิดการหมุนแกนองค์ประกอบที่ละแกนครบทุกคู่ โดยใช้เกณฑ์ตัดสินใจการคัดเลือกตัวแปรในแต่ละองค์ประกอบว่าแต่ละองค์ประกอบต้องมีตัวแปรอย่างน้อย 3 ตัวแปรขึ้นไป และมีค่าน้ำหนักองค์ประกอบตั้งแต่ 0.30 ขึ้นไป

4) การหาความสัมพันธ์ระหว่างตัวแปรประจักษ์ ในลักษณะของเมทริกซ์สหสัมพันธ์ โดยใช้ค่าสถิติต่าง ๆ ดังนี้

KMO and Bartlett's test of sphericity เป็นการคำนวณค่าสถิติ 2 ตัว คือ KMO และ Bartlett's test of sphericity โดยค่า KMO เป็นการตรวจสอบหาความเหมาะสมของกลุ่มตัวอย่าง โดยค่าของ KMO ควรจะมากกว่า 0.6 ถ้าขนาดของกลุ่มตัวอย่างเหมาะสม สำหรับ Bartlett's test of sphericity เป็นการตรวจสอบเมทริกซ์สหสัมพันธ์ของประชากรว่าเป็นเมทริกซ์เอกลักษณ์หรือไม่ (Identity matrix : คือเมทริกซ์ที่แนวทแยงมีค่าเป็น 1 และเหนือและต่ำกว่าแนวทแยงมีค่าเป็น 0) ซึ่งถ้าเป็นเมทริกซ์เอกลักษณ์แล้วหมายความว่าตัวแปรแต่ละตัวแปรมีความเป็นอิสระจากกันอย่างสมบูรณ์ ดังนั้น ค่า Bartlett's test of sphericity ควรจะมีนัยยะสำคัญทางสถิติ (Sig) น้อยกว่า 0.05 (ปรารธนา เข้มผกา & ประวิทย์ เขมะสุนันท์, 2553)

$$KMO = \frac{\sum r_i^2}{\sum r_i^2 + \sum (\text{partial correlation})^2} \quad (2.5)$$

โดย r = ค่าสัมประสิทธิ์สหสัมพันธ์ซึ่งทำให้ค่า $0 < KMO < 1$

2.4.5 แบบจำลองสมการโครงสร้าง (Structural Equation Modeling)

โมเดลสมการโครงสร้าง (Structural Equation Modeling) หรือ SEM เป็นเทคนิควิธีการวิเคราะห์ทางสถิติโดยมุ่งเน้นโมเดลความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝงเชิงทฤษฎี ที่มีความสัมพันธ์ต่อกันหลาย ๆ ตัวแปร หรือใช้วิเคราะห์สำหรับโมเดลความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝง (Latent Variables) กับตัวแปรสังเกตได้ (Observed Variable) โดยทำการประมาณค่าพารามิเตอร์ของโมเดลพร้อมกันทั้งหมดด้วยระบบสมการโมเดลโครงสร้าง หรือ SEM เพื่อเป็นเทคนิคทางสถิติที่ใช้ในการยืนยัน (Confirmatory) ว่าผลงานวิจัยมีความถูกต้องมากน้อย

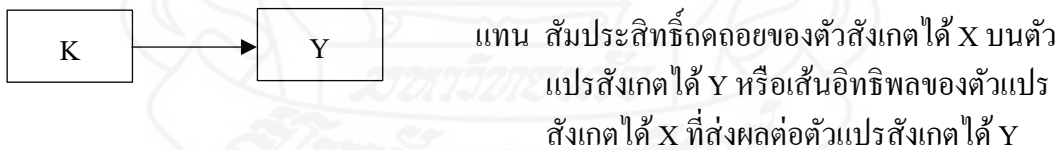
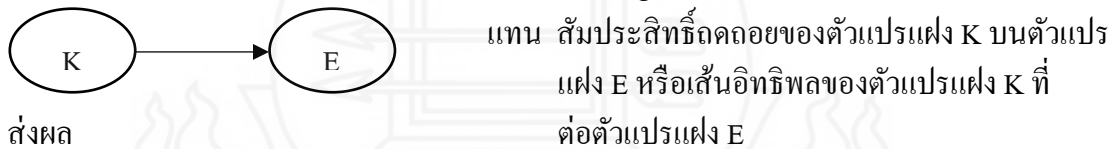
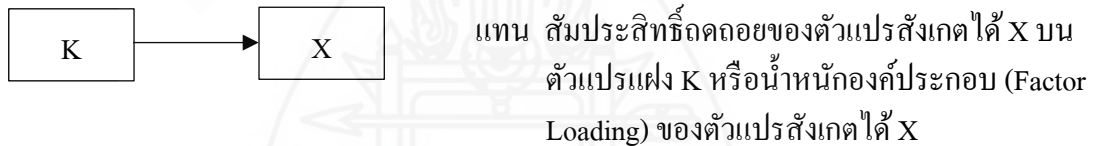
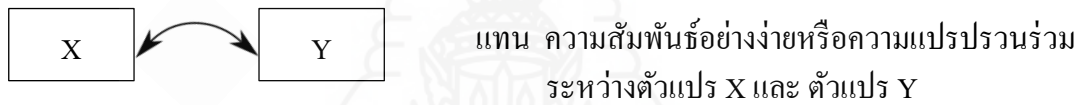
เพียงใดเมื่อเปรียบเทียบกับข้อมูลจริงในปรากฏการณ์หรือข้อมูลเชิงประจักษ์ (มหาวิทยาลัยนเรศวร, 2558)

สัญลักษณ์ที่ใช้ในโมเดลสมการโครงสร้าง แบ่งเป็น 3 ลักษณะ ได้แก่ สัญลักษณ์ที่ใช้แทนตัวแปร สัญลักษณ์ที่ใช้แทนความสัมพันธ์ และสัญลักษณ์แทนความคลาดเคลื่อน มีลักษณะดังนี้

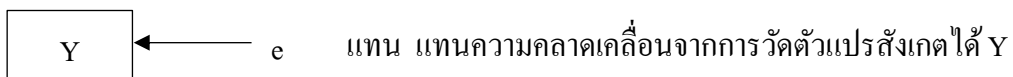
1) สัญลักษณ์ที่ใช้แทนตัวแปร ประกอบไปด้วยตัวแปร 2 ลักษณะ ได้แก่ตัวแปรสังเกตได้ (observed variable) และตัวแปรแฝง (Latent variable) ใช้สัญลักษณ์ดังนี้



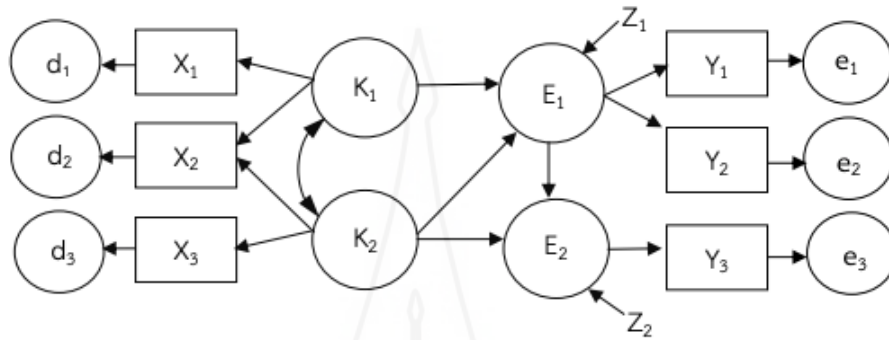
2) สัญลักษณ์ที่ใช้แทนความสัมพันธ์ มี 2 ลักษณะ ได้แก่ ความสัมพันธ์อย่างง่าย (Simple Correlation) หรือ ความแปรปรวนร่วม (Covariance) และความสัมพันธ์เชิงเหตุผล หรืออิทธิพล (Effect) มีสัญลักษณ์ ดังนี้



3) สัญลักษณ์ที่ใช้แทนความคลาดเคลื่อน มี 2 ลักษณะ ได้แก่ ความคลาดเคลื่อนจากการวัด (Measurement Error) และความคลาดเคลื่อนจากการพยากรณ์ มีสัญลักษณ์ดังนี้



โมเดลหลักของสมการโครงสร้างประกอบไปด้วยโมเดลสำคัญ 2 ส่วน คือ โมเดลการวัด (Measurement Model) และ โมเดลโครงสร้าง (Structural Model)



ภาพที่ 2.8 ตัวอย่างโมเดลหลักของโมเดลสมการโครงสร้าง

1) โมเดลการวัด (Measurement Model) คือโมเดลที่ระบุความสัมพันธ์เชิงเส้นของตัวแปรแฝงกับตัวแปรสังเกตได้ เป็นส่วนหนึ่งของการวิเคราะห์องค์ประกอบเชิงยืนยัน จากภาพที่ 2.9 เป็นโมเดลที่ระบุความสัมพันธ์เชิงเส้นระหว่าง K ซึ่งเป็นตัวแปรแฝง กับ X เป็นชุดตัวแปรสังเกตได้ และ d เป็นชุดของตัวแปรความคลาดเคลื่อน ซึ่ง K ทำหน้าที่เป็นตัวแปรแฝงภายนอก (Exogenous Latent Variable) วัดได้จาก X ซึ่งเป็นตัวแปรสังเกตได้ภายนอก (Exogenous Observed Variable) และ d เป็นตัวแปรความคลาดเคลื่อนจากการวัดตัวแปรสังเกตได้ภายนอก X นอกจากนั้น จากภาพที่ 2.9 โมเดลประกอบด้วยการวัดโมเดลระบุความสัมพันธ์เชิงเส้นระหว่าง E กับชุดตัวแปรสังเกตได้ Y และชุดของตัวแปรความคลาดเคลื่อน e ซึ่งใช้ภาพแบบโมเดลเหมือนกัน

2) โมเดลโครงสร้าง (Structural Model) คือโมเดลระบุความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝง จากภาพที่ 2.9 เป็นโมเดลมีโครงสร้างระบุความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝง K_1, K_2 กับตัวแปรแฝง E_1 และความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝง K_2 กับ E_2 รวมถึงความสัมพันธ์เชิงสาเหตุระหว่างตัวแปรแฝง E_1, E_2 อีกด้วย

ขั้นตอนการพัฒนาโมเดลสมการโครงสร้าง มีขั้นตอนสำคัญ ดังนี้

- 1) การกำหนดโครงสร้างของโมเดล (Model Specification)
- 2) การระบุลักษณะเฉพาะของโมเดล (Model Identification)
- 3) การประมาณค่าพารามิเตอร์ของโมเดล
- 4) การตรวจสอบความสอดคล้องของโมเดลกับข้อมูลเชิงประจักษ์
- 5) การปรับโมเดล

2.4.6 การประมาณค่าความแม่นยำ (Accuracy Evaluation)

การประมาณความแม่นยำ จากวิธีการสร้างแบบจำลองสมการเชิงโครงสร้าง ซึ่งต้องมีความแม่นยำเข้ากันได้ (Model Best Fit) โดยจะถูกนำไปตรวจสอบกับชุดข้อมูลที่รับค่าจริง (Actual Data) กับผลลัพธ์จากการประมาณค่าของชุดข้อมูลใหม่ (Predicted Data) ด้วยการนำไปหาค่าความคลาดเคลื่อนสัมพัทธ์ (Magnitude of Relative Error : MRE) ดังสมการที่ 2.4

$$MRE = \frac{\text{Actual Cost}(i) - \text{Predicted}(i)}{\text{Actual Cost}(i)} \quad (2.4)$$

โดย MRE = ค่าความคลาดเคลื่อนสัมพัทธ์

Actual Cost(i) = ค่าปริมาณแท้จริงของตัวอย่างที่ใช้ในการทดสอบลำดับที่ i

Predicted(i) = ค่าปริมาณจากการประมาณการตัวอย่างที่ใช้ทดสอบลำดับที่ i

หากชุดข้อมูลมีปริมาณมากต้องทำการหาค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (Mean-MMRE) หากค่า MMRE มีค่ามากแสดงว่าค่าที่ได้จากการประมาณค่าสูญหายมีความแม่นยำน้อย แต่ถ้าค่า MMRE มีค่าเข้าใกล้ 0 แสดงว่าค่าที่ได้มีความแม่นยำ (Accuracy) สูง ดังสมการที่ 2.5

$$MMRE = \frac{1}{n} \sum_{i=1}^n \frac{|\text{Actual}(i) - \text{Predicted}(i)|}{\text{Actual}(i)} \times 100 \quad (2.5)$$

$$\text{Accuracy} = 100 - MMRE$$

2.5 งานวิจัยอื่น ๆ ที่เกี่ยวข้อง (Related Research)

จากการศึกษาค้นคว้างานวิจัย วรรณกรรมที่มุ่งเน้นศึกษาในส่วนของงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ ตั้งแต่อดีตถึงปัจจุบัน ได้พบงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

ณัฐรัตน์ หาญวรวงศ์ (2556) ได้ทำการวิจัยเรื่อง การออกแบบและพัฒนาระบบการสร้างกรณีทดสอบสำหรับการทดสอบซอฟต์แวร์แบบอัตโนมัติโดยใช้โครงสร้างยูไอ ผลการวิจัยพบว่า นักทดสอบซอฟต์แวร์จะทำการทดสอบซอฟต์แวร์เวอร์ชันใหม่โดยใช้กรณีทดสอบที่มีอยู่แล้ว เพื่อทดสอบให้มั่นใจว่าการเปลี่ยนแปลงซอฟต์แวร์ไม่ได้ส่งผลให้การทำงานต่าง ๆ ที่เคยมีอยู่ผิดพลาด โดยการใช้เครื่องมือต่าง ๆ สำหรับสร้างกรณีทดสอบแบบอัตโนมัติโดยใช้โครงสร้างยูไอ ซึ่งช่วยให้การพัฒนาระบบทดสอบสำหรับการทดสอบซอฟต์แวร์แบบอัตโนมัติมีความสะดวกและรวดเร็ว

นันท์นันทน์ ช่วยชู (2558) ได้ทำการวิจัยเรื่อง การสร้างกรณีทดสอบโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม ผลการวิจัยพบว่า การทดสอบซอฟต์แวร์เป็นหนึ่งในขั้นตอนที่สำคัญในกระบวนการผลิตซอฟต์แวร์ การทดสอบซอฟต์แวร์ช่วยให้ซอฟต์แวร์ที่พัฒนามีความน่าเชื่อถือ และมีความถูกต้องเพิ่มมากขึ้น รวมถึงการสร้างกรณีทดสอบก็เป็นอีกองค์ประกอบที่สำคัญในกระบวนการทดสอบซอฟต์แวร์ คุณภาพของการทดสอบจะขึ้นอยู่กับคุณภาพของกรณีทดสอบ

เศรษฐพงษ์ อิ่มสุวรรณ (2560) ได้ทำการวิจัยเรื่อง การใช้ระบบอัตโนมัติทดสอบซอฟต์แวร์เว็บแอปพลิเคชัน ผลการวิจัยพบว่า การทดสอบแบบอัตโนมัติที่นำมาประยุกต์ใช้กับการทดสอบซอฟต์แวร์แอปพลิเคชันทำให้ลดระยะเวลาในการทดสอบซอฟต์แวร์แอปพลิเคชัน และสามารถช่วยลดแรงงาน ค่าใช้จ่าย ความผิดพลาดในการบันทึกข้อมูล มากกว่าการทดสอบซอฟต์แวร์แอปพลิเคชันด้วยมือ

สกรณ์ บุษบง (2556) ได้ทำการวิจัยเรื่อง การสร้างกรณีทดสอบสำหรับการทดสอบระดับรวมหน่วยแบบเพิ่มทีละหน่วยโดยอัตโนมัติจากกรณีทดสอบระดับหน่วย ผลการวิจัยพบว่า วิธีการสร้างกรณีทดสอบสำหรับการทดสอบระดับหน่วยสามารถช่วยลดความซ้ำซ้อนและเวลาในการสร้างกรณีทดสอบ โดยใช้เครื่องมือที่ใช้ในการสร้างกรณีทดสอบระดับรวมหน่วยอัตโนมัติ (Unit Test) ในการลดขั้นตอนที่มีความซับซ้อน ในการทดสอบระดับรวมหน่วย (Integration Test)

ศิวมล เย็นสว (2558) ได้ทำการวิจัยเรื่อง แนวทางการแก้ไขปัญหาการทดสอบซอฟต์แวร์ด้วยวิธี ATTD และการบริหารผลปฏิบัติงาน ผลการวิจัยพบว่า การใช้แนวทางการแก้ไขปัญหาการทดสอบซอฟต์แวร์ด้วยวิธี Acceptance Test Driven Development : ATTD ช่วยในด้านของการออกแบบกรณีทดสอบก่อนเริ่มการทดสอบ โดยการใช้เครื่องมือในการทดสอบแบบอัตโนมัติช่วยค้นพบจุดบกพร่องในการทดสอบและสามารถทำให้แก้ปัญหาได้อย่างรวดเร็ว

Sandeep Sivanandan และ Yogeisha C. B (2557) ได้ทำการวิจัยเรื่อง Agile Development Cycle: Approach to Design an Effective Model Based Testing with Behaviour Driven Automation Framework ผลการวิจัยพบว่า การทดสอบระบบแบบอัตโนมัติมีผลต่อกระบวนการในกรณีทดสอบที่นำไปใช้กับกรอบการทำงานและพฤติกรรมการพัฒนาภาพแบบการทดสอบแบบอัตโนมัติโดยอาศัยภาพแบบการทำงานแบบ Agile เพื่อช่วยให้การทำงานในแต่ละขั้นตอนสามารถนำกลับมาทดสอบซ้ำได้อย่างมีประสิทธิภาพโดยอาศัยเทคนิค Model Base Testing (MBT) ที่ทำงานร่วมกันระหว่าง Graph Walker Robot Framework และ Selenium Web Driver เพื่อทำกระบวนการสร้างและสั่งการทดสอบด้วยสคริปต์ ซึ่งส่งผลให้การทำงานระหว่าง Model Base Testing (MBT) ร่วมกับกระบวนการพัฒนาแบบ Agile เกิดความยืดหยุ่นในการใช้งาน สามารถบำรุงรักษาและดูแลได้ง่าย รวมถึง Robot Framework ยังเป็นเครื่องมือที่ใช้ในการสั่งงานแบบอัตโนมัติ และปรับปรุงขั้นตอนการพัฒนาให้ยืดหยุ่นตามบริบทของงานได้

Rijwan Khan , Akhilesh Kumar Srivastava and Dilleshwar Pandey (2559) ได้ทำการวิจัยเรื่อง Agile Approach for Software Testing Process ผลการวิจัยพบว่า กระบวนการทดสอบระบบนั้นเป็นส่วนหนึ่งของกระบวนการพัฒนาซอฟต์แวร์ ในการทำงานในภาพแบบของ Agile จำเป็นจะต้องมีนักทดสอบระบบเป็นผู้กำหนดกรณีทดสอบตามพฤติกรรมของลูกค้า โดยการทำงานภาพแบบ Agile ออกแบบขึ้นมาเพื่อให้กระบวนการทางธุรกิจและความต้องการทางเทคนิคเกิดการประสานร่วมกันในการนำไปใช้เพื่อให้ได้บรรลุวัตถุประสงค์ในการทำงาน ส่วนกระบวนการทดสอบระบบเป็นส่วนที่ช่วยในการตรวจสอบ ยืนยันผลกระทบ ข้อปัญหา รวมถึงข้อบกพร่องต่าง ๆ ดังนั้น Agile Testing เป็นกระบวนการทำงานที่ผสมผสานกันระหว่างกระบวนการทำงานแบบ Agile และ

กระบวนการทดสอบระบบ โดยทีมพัฒนาและลูกค้ามีส่วนร่วมในการทำงานด้วยกัน ทำให้ซอฟต์แวร์เกิดประสิทธิภาพสูงสุด

Harsha T และ B A Sujatha Kumari (2560) ได้ทำการวิจัยเรื่อง Software Test Automation with Robot Framework ผลการวิจัยพบว่า Robot Framework เป็นเครื่องมือคู่ขนานสำหรับแอปพลิเคชันและเทคนิคเชิงลึก ที่ช่วยให้ทำงานร่วมกันเมื่อมีการปรับให้เป็นเครื่องมือการทดสอบแบบอัตโนมัติโดยอาศัยข้อมูลการทดสอบ กรอบการทำงานแบบอัตโนมัติ (Robot Framework) Test Library และซอฟต์แวร์ภายใต้การทดสอบ เข้าสู่กระบวนการพัฒนาด้วยภาษาคอมพิวเตอร์ให้ขับเคลื่อนเป็นกระบวนการทดสอบและยอมรับระหว่างผู้ใช้งาน นักพัฒนาระบบ และผู้ทดสอบระบบ เพื่อปรับความเข้าใจด้านความต้องการระบบของลูกค้า โดยอาศัยการสร้างกรณีทดสอบ การใช้งาน Keyword การวัดผลและรายงานผล และคำสั่งในการสั่งงานกรณีทดสอบ ซึ่งส่งผลให้คุณภาพของซอฟต์แวร์ที่เกิดขึ้นด้วยกระบวนการทดสอบแบบอัตโนมัติ มีมาตรฐานในการบริหารจัดการมากขึ้น ไม่ใช่เพียงแค่การพัฒนาขึ้นเพียงหนึ่งครั้ง แต่สามารถนำกลับมาใช้งานร่วมกับการทดสอบซ้ำและปรับเปลี่ยนกระบวนการทดสอบได้ในอนาคต เพื่อเพิ่มประสิทธิภาพการทดสอบและคุณภาพของซอฟต์แวร์ อันถือเป็นวัตถุประสงค์ของการทดสอบระบบ

Stanislav Stresnjak และ Zeljko Hocenski (2554) ได้ทำการวิจัยเรื่อง Usage of Robot Framework

in Automation of Functional Test Regression ผลการวิจัยพบว่า การทดสอบระบบแบบมือเป็นกระบวนการที่ใช้เวลานานในการทำการทดสอบแบบถดถอย (Regression Testing) การทดสอบระบบแบบอัตโนมัติจึงเป็นการบวนการทดสอบที่ตอบโต้มากกว่า โดยโรบอทเฟรมเวิร์ค (Robot Framework) เป็นเครื่องมือที่นิยมใช้ทำการทดสอบแบบอัตโนมัติ และสามารถทดสอบได้หลายสภาพแวดล้อมไม่ว่าจะเป็น การทดสอบหน้าประสาณผู้ใช้งาน (User Interface) ส่วนต่อประสานแอปพลิเคชัน (API) โดยทำการทดสอบได้อย่างมีประสิทธิภาพ รวดเร็ว ง่ายต่อการทดสอบแบบถดถอยหรือทดสอบแบบรวมหน่วย และประหยัดค่าใช้จ่าย ให้กับโครงการ

Evvari Koppel (2555) ได้ทำการวิจัยเรื่อง Software Test Management Tool Evaluation Framework ผลการวิจัยพบว่า ผลการวิจัยพบว่าการใช้เครื่องมือบริหารจัดการทดสอบซอฟต์แวร์ที่สร้างขึ้น ช่วยให้องค์กรบรรลุเป้าหมายและวัตถุประสงค์ และเครื่องมือบริหารจัดการทดสอบซอฟต์แวร์มีความสามารถในการทดสอบซอฟต์แวร์ได้ในอนาคต โดยความต้องการของเครื่องมือบริหารจัดการทดสอบซอฟต์แวร์ จะเป็นพื้นฐานในการนำไปสู่กระบวนการทดสอบยุคใหม่ที่สามารถทำให้เกิดความสนใจในกระบวนการทดสอบระบบมากขึ้นผ่านเครื่องมือบริหารจัดการทดสอบซอฟต์แวร์ในการวัดผล

จากการศึกษางานวิจัยที่เกี่ยวข้องทั้งหมดได้ทำการศึกษาเครื่องมือและกระบวนการที่ใช้ในการพัฒนาและวัดผลการวิจัยโดยนำโรบอทเฟรมเวิร์ค (Robot Framework) ซึ่งจากงานวิจัยที่เกี่ยวข้องพบว่าโรบอทเฟรมเวิร์ค (Robot Framework) เป็นเครื่องมือที่ใช้ในการสนับสนุนการพัฒนากระบวนการการทดสอบแบบอัตโนมัติ ซึ่งมีความยืดหยุ่นในการปรับกระบวนการการทำงานให้เหมาะสมกับภาพแบบวิธีการทำงานแบบ Agile Methodology ควบคู่ไปกับ

กระบวนการพัฒนากรณีทดสอบแบบอัตโนมัติ (Automate Test Case) และแบบมือ (Manual Test Case) อันสามารถนำไปสู่กระบวนการประเมินประสิทธิภาพและวัดผลการใช้งานการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบไฮลด์ที่สามารถวัดผลด้วยความแม่นยำในการทดสอบ คุณภาพและระยะเวลาในการทดสอบ เพื่อสนับสนุนการตัดสินใจในการเลือกใช้กระบวนการทดสอบแบบอัตโนมัติที่พัฒนาขึ้นด้วยกระบวนการยอมรับการพัฒนาการทดสอบ Acceptance Tests Driven Development (ATTD) ซึ่งจากงานวิจัยที่เกี่ยวข้องโดยส่วนใหญ่ได้เลือกมาใช้ในการศึกษาและวิเคราะห์กระบวนการวิจัย



ตารางที่ 2.1 เอกสารและงานวิจัยที่เกี่ยวข้อง

ลำดับ	ชื่องานวิจัย/ผู้วิจัย	ระบบ/ชิ้นงาน (เช่น Recommendation System, Question & Answering, DSS)	วิธีการวิเคราะห์ ออกแบบ พัฒนา (เช่น UML, DFD, SDLC, Agile)	ภาษา/เครื่องมือ/ฐานข้อมูล ในการพัฒนา (เช่น Ionic Framework, PHP, Python, MySQL, VS Code)	อัลกอริทึม/เทคนิค (เช่น Clustering, Content-Based, Rule-Based, API, Chatbot, RPA)
1	การออกแบบและพัฒนาระบบการทดสอบสำหรับทดสอบซอฟต์แวร์แบบอัตโนมัติโดยใช้โครงสร้างยูไอ : หาญวรวงศ์, ณัฐรัตน์	DSS	ATDD	Visual Studio Test	-
2	Software Test Management Tool Evaluation Framework : Evari Koppel	Recommendation System	SDLC, Agile	Evaluation Framework	-
3	การสร้างกรณีทดสอบโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม : นันทินี ช่วยชู	DSS	DFD	Visual Studio .NET	Genetic Algorithm
4	การใช้ระบบอัตโนมัติทดสอบซอฟต์แวร์เว็บแอปพลิเคชัน : เศรษฐพงษ์ อิ่มสุวรรณ	DSS	BDD	Robot Framework	-

ลำดับ	ชื่องานวิจัย/ผู้วิจัย	ระบบ/ชิ้นงาน (เช่น Recommendation System, Question & Answering, DSS)	วิธีการวิเคราะห์ ออกแบบ พัฒนา (เช่น UML, DFD, SDLC, Agile)	ภาษา/เครื่องมือ/ฐานข้อมูล ในการพัฒนา (เช่น Ionic Framework, PHP, Python, MySQL, VS Code)	อัลกอริทึม/เทคนิค (เช่น Clustering, Content-Based, Rule-Based, API, Chatbot, RPA)
5	Analysis and Design of Selenium WebDriver Automation Testing Framework : Satish Gojare	Recommendation System	SDLC	Automation Testing Framework	API
6	การนำแนวคิด Agile มาใช้กับการบริหารโครงการ : รัชชากานต์ เดวิส	Question & Answering	Agile	-	-
7	ปัจจัยที่ส่งผลต่อความตั้งใจในการใช้งานแนวคิด การพัฒนาซอฟต์แวร์รูปแบบ DevOps : สนิธพรณ จิตตั้งสมบูรณ์	Question & Answering	Agile, DevOps	-	-
8	Agile Development Cycle Approach to Design an Effective Model Based Testing with Behaviour Driven Automation Framework : Sandeep Sivanandan	DSS	ATDD	Behaviour Driven Automation Framework	Test Generation Algorithm

ลำดับ	ชื่องานวิจัย/ผู้วิจัย	ระบบ/ชิ้นงาน (เช่น Recommendation System, Question & Answering, DSS)	วิธีการวิเคราะห์ ออกแบบ พัฒนา (เช่น UML, DFD, SDLC, Agile)	ภาษา/เครื่องมือ/ฐานข้อมูล ในการพัฒนา (เช่น Ionic Framework, PHP, Python, MySQL, VS Code)	อัลกอริทึม/เทคนิค (เช่น Clustering, Content-Based, Rule-Based, API, Chatbot, RPA)
9	Usage of Robot Framework in Automation of Functional Test Regression : Stanislav Stresnjak	DSS	ATDD	Python, JAVA	Keyword Driven Testing Approach
10	การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วย โรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงาน แบบไฮลิ : อภิสิทธิ์ เชนนาม	DSS	Agile	Robot Framework	SEM MMRE

บทที่ 3

วิธีดำเนินการวิจัย

การวิจัยครั้งนี้ผู้วิจัยได้ดำเนินการวิจัยเพื่อให้สอดคล้องกับวัตถุประสงค์ของการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบบอใจล์ ซึ่งมีรายละเอียดวิธีการดำเนินการวิจัยดังต่อไปนี้

- 3.1 พัฒนาการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอใจล์
- 3.2 ทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอใจล์
- 3.3 ประเมินประสิทธิภาพการทดสอบภาพซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบบอใจล์

3.1 พัฒนาการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและกรณีทดสอบแบบมือภายใต้รูปแบบการทำงานแบบบอใจล์

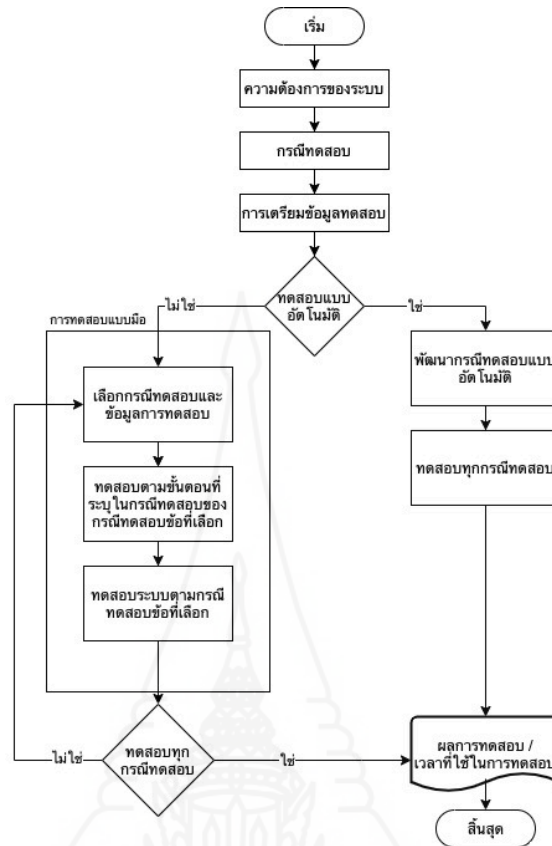
3.1.1 ประชากร

ประชากรที่ใช้ในการวิจัยและจัดทำวิทยานิพนธ์ ได้แก่ บุคลากรที่ประกอบอาชีพด้านเทคโนโลยีและปริมาณงานที่มีส่วนเกี่ยวข้องกับการทดสอบระบบ ในแผนก QA Digital Platform ของบริษัท A จำนวน 48 ทีม คิดเป็นจำนวนบุคลากร 151 คน

3.1.2 กลุ่มตัวอย่าง

กลุ่มตัวอย่างที่ใช้ในการวิจัยในครั้งนี้ ได้แก่บุคลากรที่ประกอบอาชีพด้านเทคโนโลยีสารสนเทศ ที่มีส่วนเกี่ยวข้องกับการทดสอบระบบ ภายใน บริษัท A ในปีพุทธศักราช 2563 โดยวิธีการสุ่มอย่างง่าย (Simple Random Sampling) จากการเปิดตารางเครจซี่และมอร์แกน (Krejcie & Morgan) ให้เกิดเป็นกลุ่มตัวอย่างจากปริมาณงานแทนปริมาณจำนวนบุคลากร โดยมีลักษณะการสุ่มตามปริมาณงานที่ใช้ในการทำกรณีทดสอบแบบอัตโนมัติร่วมกับการทดสอบแบบมือรวมทั้งสิ้น 108 ตัวอย่าง

ผู้วิจัยได้ดำเนินการวิจัยโดยการศึกษา พัฒนา กรณีทดสอบแบบมือและกรณีทดสอบแบบอัตโนมัติ โดยใช้โรบอทเฟรมเวิร์ค เพื่อนำมาทดสอบและวิเคราะห์ผล ดังภาพที่ 3.1



ภาพที่ 3.1 แผนภาพแสดงขั้นตอนการพัฒนากรณีทดสอบแบบมือและกรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค

3.1.3 กรณีทดสอบแบบอัตโนมัติ

ผู้วิจัยได้พัฒนากรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คในการพัฒนากรณีทดสอบในการทดสอบระบบแบบอัตโนมัติ ซึ่งมีขั้นตอนในการพัฒนา ดังนี้

1) ขั้นตอนการวิเคราะห์เอกสารความต้องการ (Requirement / User Story) เป็นขั้นตอนที่นำข้อมูลจากเอกสารความต้องการมาวิเคราะห์และสร้างเป็นสถานการณ์การทดสอบ (Test Scenario) สามารถแบ่งออกเป็นสถานการณ์ทดสอบแบบปกติ (Positive Scenario) และสถานการณ์ทดสอบแบบไม่ปกติ (Negative Scenario) จนครอบคลุมทุกฟังก์ชันของระบบ เพื่อให้มองเห็นทุกมุมมองของระบบได้ง่ายและชัดเจน

2) ขั้นตอนการสร้างกรณีทดสอบ เป็นขั้นตอนการนำสถานการณ์การทดสอบ (Test Scenario) ที่ออกแบบครอบคลุมทั้งระบบแล้ว มาทำการแยกเงื่อนไขแต่ละเงื่อนไขให้อยู่ในกรณีทดสอบในแต่ละข้อ ตัวอย่างเช่น ระบบมีสถานการณ์การทดสอบ 1 ชุด คือ สถานการณ์การทดสอบของหน้า Login ซึ่งประกอบไปด้วยการตรวจสอบเงื่อนไขการทดสอบแบบปกติและไม่ปกติของหน้า Login โดยจากสถานการณ์การทดสอบของหน้า Login แบบปกตินั้น สามารถแยกออกเป็นกรณีทดสอบโดยแยกเป็นเงื่อนไขเดียวได้ เช่น กรณีกรอกรหัสผู้ใช้งานถูกต้องสามารถทำ

การ Login ได้ หรือ กรณีกรอกรหัสผ่านถูกต้องสามารถทำการ Login ได้ เป็นต้น เช่นเดียวกันกับ สถานการณ์การทดสอบของหน้า Login แบบไม่ปกติ สามารถแยกออกเป็นกรณีทดสอบโดยแยก เป็นเงื่อนไขเดียวได้ เช่น กรณีกรอกรหัสผู้ใช้งานไม่ถูกต้อง ต้องไม่สามารถทำการ Login ได้ หรือ กรณีกรอกรหัสผ่านไม่ถูกต้อง ต้องไม่สามารถทำการ Login ได้ เป็นต้น โดยการสร้างกรณีทดสอบ มีส่วนประกอบ ดังตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างกรณีทดสอบ

No.	Test Case Name	Test Step	Test Data	Expected Result	Actual Result	Test Result
1	Validate login invalid email	1. Enter Email. 2. Input invalid email. 3. Input valid password. 4. click login.	Email : testinvalid Password : 123456	Cannot login email.		
2	Validate login valid email	1. Enter Email. 2. Input valid email. 3. Input valid password. 4. click login.	Email : testvalid@mail.com Password : 123456	login email success.		

โดยรายละเอียดกรณีทดสอบ ดังนี้

No.	คือ สิ่งที่แสดงถึงลำดับของแต่ละข้อของกรณีทดสอบ
Test Case Name	คือ สิ่งที่อธิบายชื่อของกรณีทดสอบที่ต้องการตรวจสอบของกรณี
Test Step	ทดสอบในข้อนั้น ๆ คือ สิ่งที่บ่งบอกถึงขั้นตอนการทดสอบเพื่อไปให้ถึงจุดที่ต้องการตรวจสอบของกรณีทดสอบข้อนั้น ๆ
Test Data	คือ ข้อมูลที่จำเป็นในการทำการทดสอบของกรณีทดสอบของข้อนั้น ๆ
Expected Result	คือ ผลของการทดสอบตามสมมุติฐานที่ควรจะเป็น โดยยึดจากเอกสารความต้องการของระบบ
Actual Result	คือ ผลของการทดสอบที่เกิดขึ้นจริงหลังจากได้ทำการทดสอบกรณีทดสอบนั้น ๆ
Test Result	คือ เป็นผลของการทดสอบของกรณีทดสอบข้อนั้น ๆ โดยถ้าหากผลการทดสอบจริง (Actual Result) เป็นไปตามสมมุติฐานการทดสอบ (Expected Result) จะให้ผลเป็น ผ่าน (Pass) ถ้าหากผลการทดสอบไม่ตรงตามสมมุติฐานการทดสอบ (Expected Result) จะให้ผลเป็น ไม่ผ่าน (Fail)

3) ขั้นตอนการเตรียมข้อมูลการทดสอบ (Prepare Test Data) เป็นขั้นตอนการเตรียมข้อมูลเพื่อใช้ในการทดสอบ โดยเก็บไว้ในรูปแบบของตัวแปรต่าง ๆ แยกออกมาเป็นไฟล์เพื่อให้โรบอทเฟรมเวิร์กเรียกใช้ร่วมกับชุดกรณีทดสอบ โดยภาพแบบการเก็บข้อมูลการทดสอบให้อยู่ในรูปแบบของตัวแปร มีดังนี้

- `Scalar` เป็นภาพแบบการเก็บข้อมูลโดยกำหนดภาพแบบข้อมูลเป็นตัวอักษร เช่น `{data} = TestString`
- `List` เป็นภาพแบบการเก็บข้อมูลแบบ List หรือ Array เช่น `@{list} = a,b,c,d,e,f`
- `Dictionary` เป็นภาพแบบการเก็บข้อมูลแบบ Key-Value เช่น `@{dictionary} key1=value1 key2=value1`

4) การพัฒนากรณีทดสอบแบบอัตโนมัติ (Develop Automation Test Script) เมื่อได้ข้อมูลการทดสอบ และกรณีทดสอบที่ได้จากสถานการณ์ทดสอบแล้ว ผู้วิจัยได้ทำการพัฒนากรณีทดสอบแบบอัตโนมัติ ในภาพแบบสคริปต์การทดสอบ (Test Script) โดยมีขั้นตอนดังนี้

```
*** Settings ***
Resource    ../../Resources/Import.robot
```

ภาพที่ 3.2 ตัวอย่าง Setting ของโรบอทเฟรมเวิร์ค

ขั้นตอนนำเข้าไฟล์ หรือ Test Library ที่ใช้ในการทดสอบของชุดทดสอบ รวมถึงการใส่ข้อมูลอธิบายวัตถุประสงค์ของชุดทดสอบ

```
*** Variables ***
${url}      https://testproject.io/
${invalid_email}  testinvalid
${valid_email}   testvalid@mail.com
${password}    123456
```

ภาพที่ 3.3 ตัวอย่าง Variable ของโรบอทเฟรมเวิร์ค

ขั้นตอนนำเข้าข้อมูลการทดสอบที่จัดเตรียมไว้ในภาพแบบของตัวแปรต่าง ๆ เพื่อนำมาเรียกใช้ในการรันชุดการทดสอบ

```

*** Keywords ***
open web test
  [Arguments]  ${url}
  open browser  ${url}  gc

close advertise
  click web element  xpath=//img[@class="popup-close-img"]

login invalid email
  [Arguments]  ${email}
  input web text  xpath=//input[@name="your-email"]  ${email}
  click web element  xpath=//input[@type="submit"]

```

ภาพที่ 3.4 ตัวอย่าง Keywords ของโรบอทเฟรมเวิร์ค

ขั้นตอนการรับค่าพารามิเตอร์ (Parameter) หรือ ตัวแปร (Variable) ซึ่งทำหน้าที่เป็นขั้นตอนการทดสอบ (Test Step) ของกรณีทดสอบข้อนั้น ๆ

```

*** Test Cases ***
1_Test_Invalid_Email
  [Documentation]  test invalid email login
  open web test  ${url}
  login invalid email  ${invalid_email}
  validate invalid email
  [Teardown]  Close browser

2_Test_Valid_Email
  [Documentation]  test valid email login
  open web test  ${url}
  login invalid email  ${valid_email}
  validate valid email
  [Teardown]  Close browser

```

ภาพที่ 3.5 ตัวอย่าง กรณีทดสอบ ของโรบอทเฟรมเวิร์ค

ขั้นตอนการนำกรณีทดสอบที่สร้างไว้มาแปลงเป็นกรณีทดสอบแบบอัตโนมัติ เพื่อใช้ในการรันทดสอบ โดยสามารถตั้งชื่อของกรณีทดสอบแบบอัตโนมัติให้ตรงตาม

กรณีทดสอบที่ออกแบบไว้ ใช้ฟังก์ชัน เอกสาร (Documentation) เป็นตัวอธิบายวัตถุประสงค์ของการทดสอบ ใช้คำสำคัญ (Keyword) เป็นขั้นตอนการทดสอบ และใช้ ตัวแปร เป็นตัวจัดการข้อมูลทดสอบ

5) การดำเนินการทดสอบ (Test Execution) เป็นขั้นตอนการรันกรณีทดสอบแบบอัตโนมัติโดยใช้ชุดคำสั่ง (Command Line) โดยทำการ Change Directory ไปที่ไฟล์ที่ต้องการทำการทดสอบ จากนั้นใช้คำสั่ง ดังนี้

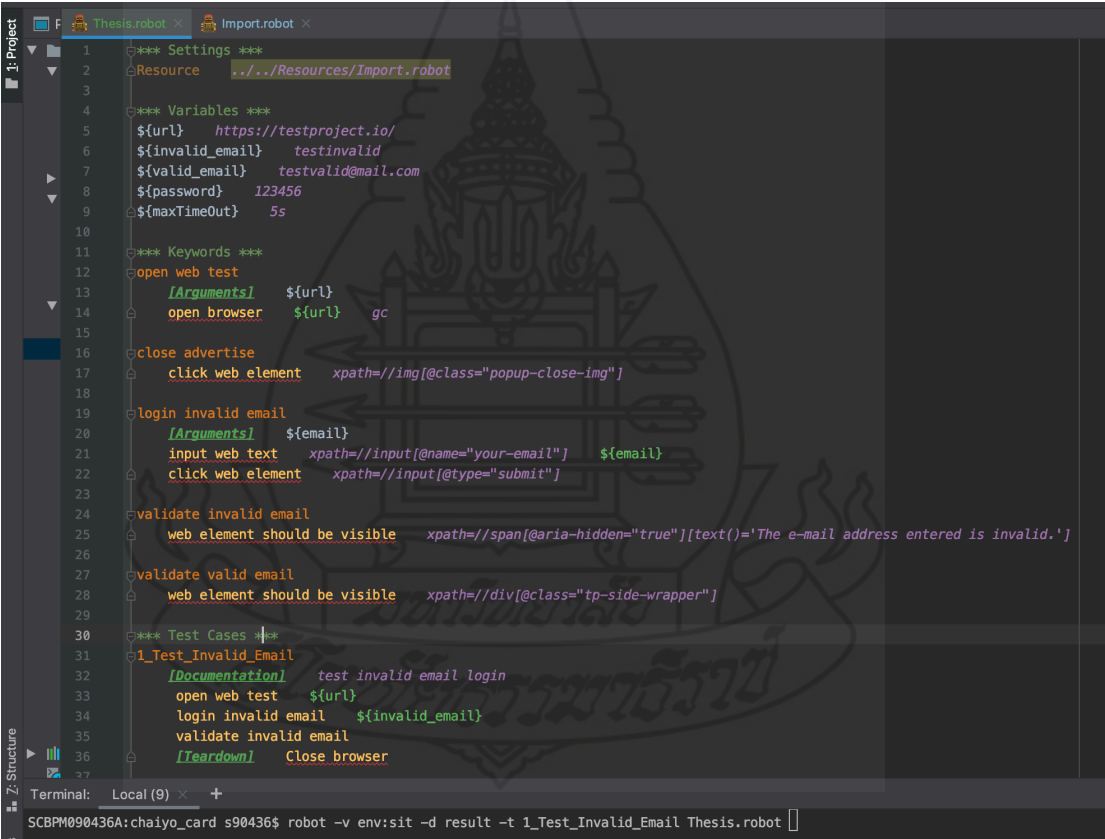
robot เป็นชุดคำสั่งของการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค

-d เพื่อกำหนดที่จัดเก็บผลการทดสอบ

-t เพื่อระบุชื่อกรณีทดสอบที่ต้องการทดสอบ

-i เพื่อระบุกลุ่มการทดสอบที่ต้องการทดสอบ

TestSuit.robot เพื่อกำหนดชื่อไฟล์ที่เก็บข้อมูลกรณีทดสอบไว้



```

1  *** Settings ***
2  Resource  ../../Resources/Import.robot
3
4  *** Variables ***
5  ${url}    https://testproject.io/
6  ${invalid_email}    testinvalid
7  ${valid_email}    testvalid@mail.com
8  ${password}    123456
9  ${maxTimeOut}    5s
10
11 *** Keywords ***
12 open web test
13   [Arguments]  ${url}
14   open browser  ${url}    gc
15
16 close advertise
17   click web element  xpath=//img[@class="popup-close-img"]
18
19 login invalid email
20   [Arguments]  ${email}
21   input web text  xpath=//input[@name="your-email"]    ${email}
22   click web element  xpath=//input[@type="submit"]
23
24 validate invalid email
25   web element should be visible  xpath=//span[@aria-hidden="true"][text()='The e-mail address entered is invalid.']
26
27 validate valid email
28   web element should be visible  xpath=//div[@class="tp-side-wrapper"]
29
30 *** Test Cases ***
31 1_Test_Invalid_Email
32   [Documentation]  test invalid email login
33   open web test  ${url}
34   login invalid email  ${invalid_email}
35   validate invalid email
36   [Teardown]  Close browser
37
Terminal: Local (9) x +
SCBPM090436A:chaiyo_card s90436$ robot -v env:sit -d result -t 1_Test_Invalid_Email Thesis.robot

```

ภาพที่ 3.6 ตัวอย่างชุดทดสอบกรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค

6) การรายงานผลการทดสอบ (Test Summary Report) เมื่อทำการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คจนครบทุกกรณีทดสอบแล้วนั้น ผลการทดสอบจะถูกจัดเก็บเป็น

ภาพแบบ .html ไฟล์โดยจะเก็บไว้ที่ Directory (-d) ที่ทำการกำหนดไว้ที่ชุดคำสั่งการทดสอบแบบอัตโนมัติ โดยการรายงานผลการทดสอบจะประกอบไปด้วยข้อมูลของการทดสอบต่าง ๆ รวมถึงเวลาที่ใช้ในการทดสอบทั้งหมด ดังภาพที่ 3.7

Validate						Generated																																				
						20210824 09:20:51 UTC+07:00																																				
						7 days 12 hours ago																																				
Test Statistics																																										
Total Statistics				Total	Pass	Fail	Elapsed	Pass / Fail																																		
Critical Tests				17	17	0	00:00:05	<div style="width: 100%; height: 10px; background-color: green;"></div>																																		
All Tests				17	17	0	00:00:05	<div style="width: 100%; height: 10px; background-color: green;"></div>																																		
Statistics by Tag				Total	Pass	Fail	Elapsed	Pass / Fail																																		
regression				17	17	0	00:00:05	<div style="width: 100%; height: 10px; background-color: green;"></div>																																		
Statistics by Suite				Total	Pass	Fail	Elapsed	Pass / Fail																																		
Validate Dopa				17	17	0	00:00:08	<div style="width: 100%; height: 10px; background-color: green;"></div>																																		
Test Execution Log																																										
<div style="display: flex; justify-content: space-between;"> Validate Dopa 00:00:07.733 </div> <p>Full Name: Validate Dopa</p> <p>Source: </p> <p>Start / End / Elapsed: 20210824 09:20:43.229 / 20210824 09:20:50.962 / 00:00:07.733</p> <p>Status: 17 critical test, 17 passed, 0 failed 17 test total, 17 passed, 0 failed</p> <table border="1"> <thead> <tr> <th>TEST</th> <th>Elapsed</th> </tr> </thead> <tbody> <tr><td>INTC-7193</td><td>00:00:00.269</td></tr> <tr><td>INTC-7194</td><td>00:00:00.288</td></tr> <tr><td>INTC-7195</td><td>00:00:00.265</td></tr> <tr><td>INTC-7196</td><td>00:00:00.257</td></tr> <tr><td>INTC-7197</td><td>00:00:00.277</td></tr> <tr><td>INTC-7198</td><td>00:00:00.284</td></tr> <tr><td>INTC-7199</td><td>00:00:00.301</td></tr> <tr><td>INTC-7200</td><td>00:00:00.300</td></tr> <tr><td>INTC-7201</td><td>00:00:00.296</td></tr> <tr><td>INTC-7202</td><td>00:00:00.276</td></tr> <tr><td>INTC-7203</td><td>00:00:00.266</td></tr> <tr><td>INTC-7204</td><td>00:00:00.301</td></tr> <tr><td>INTC-7205</td><td>00:00:00.267</td></tr> <tr><td>INTC-7206</td><td>00:00:00.323</td></tr> <tr><td>INTC-7207</td><td>00:00:00.283</td></tr> <tr><td>INTC-7208</td><td>00:00:00.310</td></tr> <tr><td>INTC-7209</td><td>00:00:00.279</td></tr> </tbody> </table>							TEST	Elapsed	INTC-7193	00:00:00.269	INTC-7194	00:00:00.288	INTC-7195	00:00:00.265	INTC-7196	00:00:00.257	INTC-7197	00:00:00.277	INTC-7198	00:00:00.284	INTC-7199	00:00:00.301	INTC-7200	00:00:00.300	INTC-7201	00:00:00.296	INTC-7202	00:00:00.276	INTC-7203	00:00:00.266	INTC-7204	00:00:00.301	INTC-7205	00:00:00.267	INTC-7206	00:00:00.323	INTC-7207	00:00:00.283	INTC-7208	00:00:00.310	INTC-7209	00:00:00.279
TEST	Elapsed																																									
INTC-7193	00:00:00.269																																									
INTC-7194	00:00:00.288																																									
INTC-7195	00:00:00.265																																									
INTC-7196	00:00:00.257																																									
INTC-7197	00:00:00.277																																									
INTC-7198	00:00:00.284																																									
INTC-7199	00:00:00.301																																									
INTC-7200	00:00:00.300																																									
INTC-7201	00:00:00.296																																									
INTC-7202	00:00:00.276																																									
INTC-7203	00:00:00.266																																									
INTC-7204	00:00:00.301																																									
INTC-7205	00:00:00.267																																									
INTC-7206	00:00:00.323																																									
INTC-7207	00:00:00.283																																									
INTC-7208	00:00:00.310																																									
INTC-7209	00:00:00.279																																									

ภาพที่ 3.7 การรายงานผลการทดสอบด้วยโรบอทเฟรมเวิร์ค

3.2.4 กรณีทดสอบแบบมือ

ผู้วิจัยได้พัฒนากรณีทดสอบแบบมือ ซึ่งมีขั้นตอนในการพัฒนาในการทดสอบดังนี้

1) ขั้นตอนการวิเคราะห์เอกสารความต้องการ (Requirement / User Story) เป็นขั้นตอนที่นำข้อมูลจากเอกสารความต้องการมาวิเคราะห์และสร้างเป็นสถานการณ์การทดสอบ (Test Scenario) สามารถแบ่งออกเป็นสถานการณ์ทดสอบแบบปกติ (Positive Scenario) และสถานการณ์ทดสอบแบบไม่ปกติ (Negative Scenario) จนครอบคลุมทุกฟังก์ชันของระบบ เพื่อให้มองเห็นทุกมุมมองของระบบได้ง่ายและชัดเจน

2) ขั้นตอนการสร้างกรณีทดสอบ เป็นขั้นตอนการนำสถานการณ์การทดสอบ (Test Scenario) ที่ออกแบบครอบคลุมทั้งระบบแล้ว มาทำการแยกเงื่อนไขแต่ละเงื่อนไขให้อยู่ในกรณีทดสอบในแต่ละข้อ ตัวอย่างเช่น ระบบมีสถานการณ์ทดสอบ 1 ชุด คือ สถานการณ์การทดสอบของหน้า Login ซึ่งประกอบไปด้วยการตรวจสอบเงื่อนไขการทดสอบแบบปกติและไม่ปกติของหน้า Login โดยจากสถานการณ์การทดสอบของหน้า Login แบบปกตินั้น สามารถแยกออกเป็นกรณีทดสอบโดยแยกเป็นเงื่อนไขเดียวได้ เช่น กรณีกรอกรหัสผู้ใช้งานถูกต้องสามารถทำการ Login ได้ หรือ กรณีกรอกรหัสผ่านถูกต้องสามารถทำการ Login ได้ เป็นต้น เช่นเดียวกันกับ สถานการณ์การทดสอบของหน้า Login แบบไม่ปกติ สามารถแยกออกเป็นกรณีทดสอบโดยแยกเป็นเงื่อนไขเดียวได้ เช่น กรณีกรอกรหัสผู้ใช้งานไม่ถูกต้อง ต้องไม่สามารถทำการ Login ได้ หรือ กรณีกรอกรหัสผ่านไม่ถูกต้อง ต้องไม่สามารถทำการ Login ได้ เป็นต้น

3) ขั้นตอนการเตรียมข้อมูลการทดสอบ (Prepare Test Data) เป็นขั้นตอนการเตรียมข้อมูลเพื่อใช้ในการทดสอบ โดยเก็บไว้ในไฟล์กรณีทดสอบของกรณีทดสอบแบบมือข้อนั้น

4) ขั้นตอนการทดสอบแบบมือ (Manual Testing) เป็นการทดสอบโดยใช้คนทำการทดสอบ โดยที่ต้องทำการเลือกกรณีทดสอบที่ต้องการทดสอบ จากนั้นนำข้อมูลการทดสอบมาทำการทดสอบตามขั้นตอนการทดสอบ และทำการเทียบผลการทดสอบด้วยตา และทำการทดสอบทีละข้อไปเรื่อย ๆ จนกว่าจะครบจำนวนกรณีทดสอบที่สร้างไว้

5) การรายงานผลการทดสอบ (Test Summary Report) เมื่อทำการทดสอบระบบครบทุกกรณีทดสอบแล้ว ผู้ทดสอบจะต้องบันทึกผลการทดสอบทั้งหมดลงในเอกสารเก็บผลการทดสอบ ประกอบไปด้วยผลการทดสอบ และ เวลาที่ใช้ในการทดสอบเพื่อทำเป็นรายงานผลการทดสอบ

3.2 การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอใจล์

ผู้วิจัยนำทำการทดสอบระบบแบบอัตโนมัติโดยใช้โรบอทเฟรมเวิร์กตามภาพแบบการทำงานแบบอใจล์ทั้งระบบ โดยทำการบันทึกผลการทดสอบจาก รายงานผลการทดสอบรวม (Test Summary Report) โดยทดสอบด้วยคำสั่งการทดสอบแบบอัตโนมัติโดยใช้โปรแกรมโรบอทเฟรมเวิร์ค หลังจากนั้นได้ทำการรวบรวมผลการทดสอบโดยคิดจากเวลาในการทำการทดสอบระบบของทั้งการทดสอบแบบอัตโนมัติและการทดสอบแบบมือ นำไปเปรียบเทียบกัน เพื่อคำนวณหาค่าความคลาดเคลื่อนสัมพัทธ์ของการทดสอบทั้ง 2 รูปแบบจนครบ 108 การทดสอบโดยกลุ่มตัวอย่างซึ่งกลุ่มตัวอย่าง 1 คนสามารถทำการทดสอบได้มากกว่า 1 ครั้ง จากนั้นทำการหาค่าเฉลี่ยค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) ได้เป็นค่าร้อยละความคลาดเคลื่อนสัมพัทธ์ โดยใช้สมการที่ 2.4 ซึ่งค่าที่ได้จะแสดงในบทถัดไป

หากชุดข้อมูลมีปริมาณมากต้องทำการหาค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (Mean-MMRE) หากค่า MMRE มีค่ามากแสดงว่าค่าที่ได้จากการประมาณค่าสูญหายมีความแม่นยำน้อย แต่ถ้าวัดค่า MMRE มีค่าเข้าใกล้ 0 แสดงว่าค่าที่ได้มีความแม่นยำ (Accuracy) สูง ดังสมการที่ 2.5

3.3 ประเมินประสิทธิภาพการทดสอบภาพซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบบอใจล์

ผู้วิจัยได้ศึกษา พัฒนา และจัดทำแบบประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบบอใจล์ สำหรับให้กลุ่มตัวอย่างทำการประเมินหลังการทดสอบ โดยแบ่งเป็น 3 ตอน ดังนี้

1) ตอนที่ 1 ข้อมูลทั่วไปสำหรับผู้ตอบแบบประเมินการใช้งาน ซึ่งจะเป็คำถามปลายปิดเกี่ยวกับการสอบถามของผู้ตอบแบบประเมินการใช้งาน โดยต้องทำการตอบคำถามทุกข้อ

2) ตอนที่ 2 การแสดงความคิดเห็นของผู้ตอบแบบสอบถามเกี่ยวกับการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบบอใจล์ ทั้งนี้การตอบคำถามจะแบ่งระดับคะแนนออกเป็น 5 ระดับ ดังนี้

คะแนน 5	หมายถึง มากที่สุด
คะแนน 4	หมายถึง มาก
คะแนน 3	หมายถึง ปานกลาง
คะแนน 2	หมายถึง น้อย
คะแนน 1	หมายถึง น้อยที่สุด

3) ตอนที่ 3 ข้อเสนอแนะ หรือ ความคิดเห็น เป็นคำถามปลายเปิด เพื่อให้กลุ่มตัวอย่างสามารถเสนอความคิดเห็นและข้อเสนอแนะต่าง ๆ ที่เกี่ยวข้องกับการใช้งาน กรณีทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอใจล์ ซึ่งกลุ่มตัวอย่างจะเสนอแนะความคิดเห็นหรือไม่ก็ได้

3.4.1 การหาค่าความเที่ยงตรง (Validation)

โดยนำแบบประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบบอใจล์ ที่ผู้วิจัยสร้างขึ้นไปตรวจสอบความเที่ยงตรงตามเนื้อหาจากผู้เชี่ยวชาญ จำนวน 5 คน เพื่อนำมาหาค่าดัชนีความสอดคล้องโดยใช้สูตร (IOC) (3.1)

$$IOC = \frac{\sum R}{N} \quad (3.1)$$

IOC คือ ดัชนีความสอดคล้องของเครื่องมือ

R คือ คะแนนพิจารณาของผู้เชี่ยวชาญ
 $\sum R$ คือ ผลรวมของคะแนนพิจารณาของผู้เชี่ยวชาญ
 N คือ จำนวนผู้เชี่ยวชาญ

ผู้วิจัยได้รวบรวมค่าของคำถาม IOC ที่มีค่ามากกว่า 0.5 มาใช้เป็นข้อคำถาม ซึ่งได้ตรวจสอบแบบสอบถามและเห็นว่าแบบสอบถามทุกข้อที่ผู้วิจัยสร้างขึ้นในแต่ละข้อมีความเที่ยงตรงและครอบคลุมในแต่ละด้าน และครอบคลุมวัตถุประสงค์ของการวิจัย โดยผู้วิจัยนำผลรวมของคะแนนในแต่ละข้อที่ผู้วิจัยได้ทำการประเมินมาทำการคำนวณหาค่า Validation โดยใช้ IOC ซึ่งในแต่ละข้อจะต้องไม่ต่ำกว่า 0.5 ซึ่งผลการวัดค่าดัชนีความสอดคล้อง (IOC) ของข้อคำถามมีค่าอยู่ระหว่าง 0.60 ถึง 1.00 และได้ค่า IOC เท่ากับ 0.80 ซึ่งมีความเที่ยงตรงสามารถใช้ได้

จากนั้นผู้วิจัยได้ทำการนำแบบสอบถามที่สร้างขึ้นไปทดสอบความเชื่อมั่น โดยนำแบบสอบถามไปทดสอบกลุ่มคนที่ทำงานด้านทดสอบระบบและไอทีที่ไม่ใช่กลุ่มตัวอย่างจำนวน 30 ท่าน โดยใช้วิธีการคำนวณหาค่าความเชื่อมั่น (Reliability) ของคอนบราค ซึ่งเรียกว่า ค่าสัมประสิทธิ์อัลฟา (Cronbach's Alpha Coefficient) ซึ่งได้ค่าความเชื่อมั่นเท่ากับ 0.796 ซึ่งอยู่ในระดับดีมากหมายถึงแบบสอบถามมีความน่าเชื่อถือและสามารถนำไปศึกษากับกลุ่มตัวอย่างจริงได้

ตารางที่ 3.2 ค่า Cronbach's Alpha ของกลุ่มตัวอย่าง 30 ราย

Case Processing Summary			
		N	%
Cases	Valid	30	100.00
	Excluded ^a	0	0
	Total	30	100.00

a. Listwise deletion based on all variable in the procedure

Reliability Statistics	
Cronbach's Alpha	N of Item
.796	24

3.4.2 การวิเคราะห์ข้อมูล

การดำเนินการวิเคราะห์ข้อมูล ผู้วิจัยได้ดำเนินการตามขั้นตอนดังต่อไปนี้

1) การกำหนดจำนวนตัวอย่าง ซึ่งผู้วิจัยได้ข้อมูลทั้งหมด 88 ตัวอย่าง

หลังจากที่ผู้วิจัยได้เก็บรวบรวมข้อเรียบร้อยแล้ว ได้ทำการทดสอบความน่าเชื่อถือ (Reliability Test) ด้วยเทคนิค Cronbach's Alpha เพื่อตรวจสอบว่าแบบสอบถามมีความเหมาะสมที่จะนำไปใช้ในการวิเคราะห์ประสิทธิภาพมากน้อยเพียงใดผ่านโปรแกรมสำเร็จภาพเพื่อการวิเคราะห์ทางสถิติ

ตารางที่ 3.3 ค่า Cronbach's Alpha ของกลุ่มตัวอย่าง 88 ราย

		N	%
Cases	Valid	88	100.00
	Excluded ^a	0	0
	Total	88	100.00

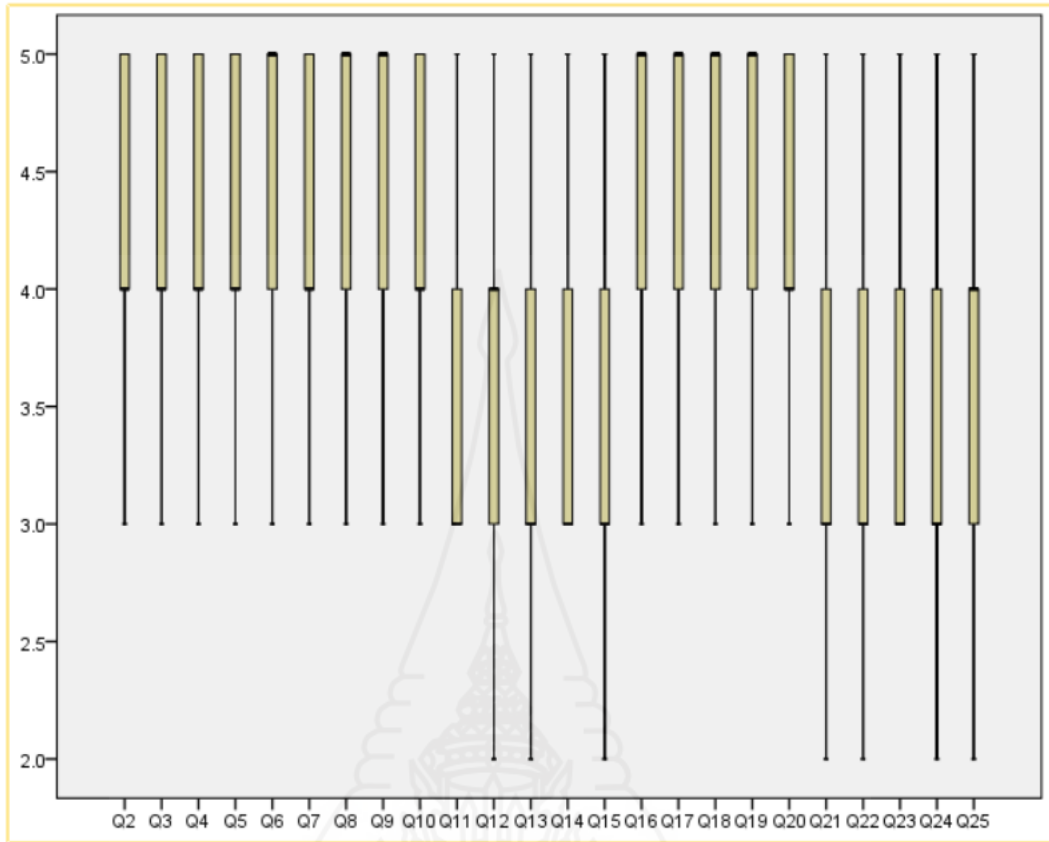
a. Listwise deletion based on all variable in the procedure

Cronbach's Alpha	N of Item
.811	24

จากตารางที่ 3.3 อธิบายค่าได้คือ ค่า N เท่ากับ 88 หมายถึง จำนวนข้อมูลทั้งหมด 88 ข้อมูลที่นำมาใช้ทดสอบ, ค่า เปอร์เซนต์ (%) เท่ากับ 100 หมายถึง กลุ่มตัวอย่างทำแบบสอบถามครบถ้วน โดยไม่ถูกตัดออกเนื่องจากข้อมูลหายไป, ค่า N of Items เท่ากับ .811 ซึ่งมีค่าใกล้ 1 หมายความว่าแบบสอบถามที่สร้างขึ้นมีความน่าเชื่อถือสูงสามารถนำไปวิเคราะห์ประสิทธิภาพได้

2) ค่าสังเกตที่ผิดปกติ (Outlier)

ผู้วิจัยได้ทำการหาค่าสังเกตที่ผิดปกติของตัวแปรที่มีค่าสูงเกินไป หรือต่ำเกินไป ผิดปกติไปจากกลุ่มตัวอย่างอื่น ๆ ทุกตัว ซึ่งผลลัพธ์ค่าสังเกตที่ผิดปกติพบว่าข้อมูลตัวชี้วัดมีค่าสังเกตผิดปกติ จึงทำการแก้ไขโดยแทนที่ค่าข้อมูลนั้น ๆ ด้วยข้อมูลที่เก็บเกี่ยวมา เนื่องจากข้อมูลที่รวบรวมมามีจำนวนที่เพียงพอ



ภาพที่ 3.8 ค่าสังเกตที่ผิดปกติของทุกตัวชี้วัดหลังจากปรับปรุงข้อมูลที่เป็น Outlier

จากภาพที่ 3.8 แสดงการวิเคราะห์หองศ์ประกอบของการทดสอบแบบอัตโนมัติภายใต้การทำงานแบบบอใจล์ ไม่มีค่าผิดปกติ หมายความว่า ไม่มีข้อมูลใด ๆ ที่มีค่าสูงเกินไป หรือต่ำมากผิดปกติไปจากกลุ่มตัวอย่างอื่น ๆ

3) การปรับปรุงข้อมูลให้เป็นค่ามาตรฐาน

เนื่องจากแบบสอบถามที่ผู้วิจัยสร้างขึ้นมานั้นมีบางคำถามที่ใช้หน่วยชี้วัดที่ต่างกัน เช่น บางตัวแปรใช้ข้อมูลที่เป็นเชิงปริมาณ บางตัวแปรเป็น 0-5 (Rating Scale) ซึ่งอาจก่อให้เกิดความผิดพลาดในการสร้างแบบจำลองได้ ดังนั้นผู้วิจัยจึงทำการปรับปรุงข้อมูลให้เป็นหน่วยเดียวกันทั้งหมดให้เป็นค่ามาตรฐาน (Standardization) เช่น ตัวชี้วัด TC_EFF จะถูกแปลงค่าเป็นค่ามาตรฐานคือ ZTC_EFF, ZTC_TIME, ZTC_ERR เป็นต้น

ZTC_EFF	ZQ2	ZQ3	ZQ4	ZQ5	ZQ6	ZQ7
-43422	1.15791	1.16849	1.14746	1.11663	.87950	98581
-43422	-38597	-30047	-33749	-30748	.87950	98581
-43422	1.15791	-1.76943	-1.82244	-1.73158	-60888	-28994
-43422	-38597	-30047	-33749	-1.73158	.87950	98581
-43422	-38597	-30047	-33749	-30748	.87950	98581
-43422	-38597	-1.76943	-33749	-1.73158	-60888	98581
-43422	-38597	-30047	1.14746	1.11663	-60888	-28994
-43422	-1.92985	-1.76943	-1.82244	-1.73158	.87950	98581
44421	-38597	-30047	-33749	-30748	-60888	-1.56570
1.32264	1.15791	1.16849	-33749	-30748	.87950	98581
-43422	1.15791	1.16849	-33749	1.11663	-60888	-28994
-87344	-38597	-30047	-33749	-30748	.87950	98581
-43422	-38597	-1.76943	-1.82244	-1.73158	-60888	-28994
44421	-38597	-30047	-33749	-30748	.87950	-1.56570
1.32264	-38597	-30047	1.14746	-30748	-2.09726	-1.56570
-43422	1.15791	1.16849	1.14746	1.11663	.87950	-28994
44421	-38597	-30047	-33749	1.11663	.87950	98581
-43422	1.15791	1.16849	-33749	1.11663	-60888	-28994
-43422	-1.92985	-1.76943	-1.82244	-1.73158	-60888	98581
-43422	1.15791	1.16849	1.14746	1.11663	.87950	98581
-43422	1.15791	1.16849	1.14746	1.11663	-60888	-28994
-43422	1.15791	1.16849	1.14746	1.11663	-60888	-28994
-43422	1.15791	1.16849	1.14746	1.11663	-60888	-28994
-43422	1.15791	1.16849	-33749	1.11663	-60888	-1.56570
44421	1.15791	1.16849	1.14746	1.11663	-2.09726	-1.56570
1.32264	-38597	-30047	-33749	-30748	-60888	-28994
-43422	1.15791	1.16849	1.14746	1.11663	.87950	98581

ภาพที่ 3.9 ค่ามาตรฐานของทุกตัวแปร (บางส่วน)

จากภาพที่ 3.3 แสดงข้อมูลหน่วยที่เป็นมาตรฐานของทุกตัวแปร เพื่อนำไปใช้ในการวิเคราะห์ข้อมูล ดังนั้นเมื่อตัวแปรใดมีการเปลี่ยนแปลงเป็นค่ามาตรฐานแล้ว โปรแกรมจะทำการกำหนดชื่อตัวแปรให้ใหม่เช่น Q1 เป็น ZQ1 และ Q2 เป็น ZQ2 เป็นต้น

4) การกำหนดค่าเบ้ (Skewness) และทำการกำหนดค่า Transform Compute Variable โดยทั่วไปแล้วข้อมูลจะพบว่าเป็นค่าข้อมูลเส้นโค้งคือเส้นโค้งชนิดสมมาตรกันหรือไม่สมมาตรกัน ซึ่งเส้นโค้งปกติหมายถึงเส้นโค้งที่ด้านซ้าย ด้านขวาและด้านกึ่งกลางเท่ากันทุกประการคือ ค่าเฉลี่ย ค่ามัธยฐาน ค่าฐานนิยม ซึ่งมีค่าเท่ากันและทับเส้นกันสนิท แต่ถ้าข้อมูลที่มีลักษณะเส้นโค้งชนิดไม่สมมาตรจะมีลักษณะเบ้ไปข้างใดข้างหนึ่ง

ถ้าค่าการแจกแจงของข้อมูลดิบมีการแจกแจงความเบ้เล็กน้อย ค่าสัมประสิทธิ์ค่าเบ้จะมีค่าอยู่ในช่วง 1 ถ้าค่าเบ้มีค่ามากกว่า 1 หรือน้อยกว่า -1 แสดงว่าค่าเบ้มีค่ามากกว่าปกติ ในกรณีที่ค่าเบ้มีความผิดปกติมากจะทำการตัดข้อนี้ทิ้งออกจากแบบสอบถามเนื่องจากไม่สามารถนำไปวิเคราะห์ได้

ตารางที่ 3.4 ผลการตรวจสอบค่าเบ้ (Skewness)

	N	Minimum	Maximum	Mean	Std. Deviation	Skewness	Std. Error
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	
Q2	88	3	5	4.25	.648	-.292	.257
Q3	88	3	5	4.20	.681	-.279	.257
Q4	88	3	5	4.23	.673	-.306	.257
Q5	88	3	5	4.22	.702	-.329	.257
Q6	88	3	5	4.41	.672	-.707	.257
Q7	88	3	5	4.23	.784	-.427	.257
Q8	88	3	5	4.36	.746	-.709	.257
Q9	88	3	5	4.40	.704	-.741	.257
Q10	88	3	5	4.36	.681	-.605	.257
Q11	88	3	5	3.60	.751	.808	.257
Q12	88	2	5	3.73	.813	.280	.257
Q13	88	2	5	3.45	.921	.227	.257
Q14	88	3	5	3.51	.678	.977	.257
Q15	88	2	5	3.32	.635	.444	.257
Q16	88	3	5	4.51	.625	-.911	.257
Q17	88	3	5	4.57	.563	-.867	.257
Q18	88	3	5	4.56	.564	-.817	.257
Q19	88	3	5	4.56	.564	-.817	.257
Q20	88	3	5	4.39	.596	-.381	.257
Q21	88	2	5	3.58	.813	.395	.257
Q22	88	2	5	3.60	.751	.641	.257
Q23	88	3	5	3.61	.734	.755	.257
Q24	88	2	5	3.60	.766	.666	.257
Q25	88	2	5	3.76	.816	.340	.257
Valid N (listwise)	88						

จากตารางที่ 3.4 ผลการตรวจสอบค่าเบ้ (Skewness) ของตัวแปรต่าง ๆ มีค่าเบ้อยู่ระหว่าง ± 1 ซึ่งหมายความว่าไม่ต้องทำการแก้ไขค่าใด ๆ ทั้งสิ้น

3.4.3 การวิเคราะห์ปัจจัย (Factor Analyze)

1) การสร้างเมทริกซ์สหสัมพันธ์ (Correlation Matrix) มีวัตถุประสงค์เพื่อใช้ในการตรวจสอบความสัมพันธ์กันระหว่างตัวแปรต่าง ๆ เพื่อใช้ในการพิจารณาเลือกตัวแปรเข้ามารวมในปัจจัยเดียวกัน การทำเมทริกซ์สหสัมพันธ์เริ่มจากการระบุตัวแปรที่ต้องการวิเคราะห์ เมื่อวิเคราะห์แล้วจะได้ค่าของ Correlation Matrix (KMO and Bertlett's test of sphericity) หลักจากการวิเคราะห์แล้วผู้วิจัยได้ทำการจัดกลุ่มตัวแปรต่าง ๆ ที่มีความสัมพันธ์กันสูง โดยพิจารณาจากค่าสัมประสิทธิ์สหสัมพันธ์ในตารางที่ 3.5

ตารางที่ 3.5 ค่าสัมประสิทธิ์สหสัมพันธ์ของตัวแปรต่าง ๆ (บางส่วน)

		Zscore(Q2)	Zscore(Q3)	Zscore(Q4)	Zscore(Q5)	Zscore(Q6)	Zscore(Q7)
Correlation	Zscore(Q2)	1.000	.925	.817	.866	.132	-.045
	Zscore(Q3)	.925	1.000	.875	.941	.167	-.045
	Zscore(Q4)	.817	.875	1.000	.867	.097	-.077
	Zscore(Q5)	.866	.941	.867	1.000	.152	-.048
	Zscore(Q6)	.132	.167	.097	.152	1.000	.629
	Zscore(Q7)	-.045	-.045	-.077	-.048	.629	1.000
	Zscore(Q8)	.143	.169	.085	.134	.732	.526
	Zscore(Q9)	.334	.356	.219	.336	.576	.272
	Zscore(Q10)	.078	.110	-.007	.098	.525	.339
	Zscore(Q11)	.301	.318	.317	.317	-.061	-.099
	Zscore(Q12)	.393	.414	.409	.427	-.067	-.082
	Zscore(Q13)	.231	.253	.239	.255	-.137	-.240
	Zscore(Q14)	.177	.194	.145	.200	.116	-.005
	Zscore(Q15)	.000	.007	.071	.050	-.012	-.008
	Zscore(Q16)	-.121	-.114	-.061	-.176	.153	.276
	Zscore(Q17)	-.142	-.127	-.041	-.139	.077	.303
	Zscore(Q18)	-.228	-.210	-.155	-.220	.180	.282
	Zscore(Q19)	-.165	-.150	-.125	-.162	.059	.230
	Zscore(Q20)	-.045	-.055	-.049	-.092	.060	.327
	Zscore(Q21)	.071	.074	.051	.100	.003	.007
	Zscore(Q22)	.089	.093	-.001	.099	.007	-.040
	Zscore(Q23)	.085	.091	.063	.119	-.025	.015
	Zscore(Q24)	.087	.092	.066	.119	-.038	-.001
	Zscore(Q25)	.201	.213	.183	.151	-.030	.068

จากตารางที่ 3.5 สามารถจัดกลุ่มตัวแปรต่าง ๆ ได้ โดยเลือกตัวแปรที่มีค่าสัมประสิทธิ์สหสัมพันธ์กันสูง ๆ มาอยู่ในปัจจัยเดียวกัน เช่น ตัวแปร Zscore(Q2) มีค่าสัมประสิทธิ์

สหสัมพันธ์กับตัวแปร Zscore(Q3) และ Zscore(Q4) เท่ากับ 0.925 และ 0.817 ตามลำดับซึ่งมีค่าสูงเมื่อนำมาสกัดปัจจัยควรอยู่ในปัจจัยเดียวกัน

2) การสกัดปัจจัย (Factor Extraction) เป็นการสกัดปัจจัยว่าตัวแปรต่าง ๆ ที่มีความสัมพันธ์กันสูงสามารถทำให้อยู่ปัจจัยเดียวกันได้กี่ปัจจัย หลังจากนั้นผู้วิจัยได้ตรวจสอบค่า KMO (Kaiser-Meyer-Olkin) ซึ่งเป็นค่าที่แสดงว่าข้อมูลที่ผู้วิจัยได้เก็บรวบรวมมานั้น มีความเหมาะสมในการนำไปสกัดปัจจัยหรือไม่ ซึ่งค่า KMO ที่สมควรมีความมากกว่า 0.6

ตารางที่ 3.6 ค่า KMO

KMO and Bartlett's test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.	0.738
Bartlett's Test of Sphericity	Approx. Chi-Square
	1850.145
	df
	276
	Sig.
	0

จาก ตารางที่ 3.6 ค่า KMO เท่ากับ 0.738 หมายความว่า ข้อมูลที่เก็บรวบรวมมาสามารถนำไปสกัดปัจจัยได้อย่างเหมาะสม มีจำนวนตัวอย่างมากพอต่อการวิเคราะห์ปัจจัย และค่า Bartlett's test มีนัยสำคัญทางสถิติ แสดงว่าตัวแปรมีความสัมพันธ์กัน สามารถนำมาสกัดปัจจัยได้ดี โดยผลการสกัดปัจจัยแบบไม่หมุนแกนแสดงในตารางที่ 3.7

ตารางที่ 3.7 ผลการสกัดปัจจัยแบบไม่หมุนแกน (Component Matrix)

	Component				
	1	2	3	4	5
Zscore(Q2)	.777	.060	-.364	.059	-.386
Zscore(Q3)	.809	.086	-.381	.070	-.388
Zscore(Q4)	.734	.085	-.348	-.039	-.451
Zscore(Q5)	.809	.055	-.360	.057	-.357
Zscore(Q6)	.090	.416	-.310	.683	.273
Zscore(Q7)	-.127	.503	-.092	.545	.137
Zscore(Q8)	.191	.603	-.233	.362	.417
Zscore(Q9)	.289	.164	-.374	.591	.156
Zscore(Q10)	.033	.217	-.277	.610	.219
Zscore(Q11)	.577	.289	-.028	-.456	.356
Zscore(Q12)	.630	.335	-.104	-.507	.300
Zscore(Q13)	.437	.221	-.069	-.595	.265
Zscore(Q14)	.427	.246	-.146	-.257	.565
Zscore(Q15)	.255	.258	.281	-.277	.246
Zscore(Q16)	-.275	.816	.074	-.159	-.184
Zscore(Q17)	-.321	.755	.168	-.093	-.380
Zscore(Q18)	-.402	.798	.045	-.110	-.191
Zscore(Q19)	-.258	.754	.153	-.123	-.151
Zscore(Q20)	-.219	.746	.027	-.134	-.198
Zscore(Q21)	.424	.000	.738	.316	-.008
Zscore(Q22)	.366	.200	.560	.036	.082
Zscore(Q23)	.461	.007	.773	.259	.004
Zscore(Q24)	.475	.004	.797	.261	.002
Zscore(Q25)	.453	.125	.670	.177	-.177

Extraction Method: Principal Component Analysis.
a. 5 components extracted.

ผลการสกัดปัจจัยแบบไม่หมุนแกนในตารางที่ 3.7 เป็นผลของการสกัดปัจจัยโดยใช้วิธี PCA พบว่า มีปัจจัยรวม 5 ปัจจัย การพิจารณาว่าตัวแปรหนึ่ง ๆ อยู่ที่ปัจจัยใด สามารถดูได้จากค่า

Factor Coefficient เช่น Zscore(Q2) มี Factor Coefficient กับปัจจัยที่ 1 ปัจจัยที่ 2 ปัจจัยที่ 3 ปัจจัยที่ 4 และ ปัจจัยที่ 5 เท่ากับ 0.777, 0.06, -0.364, 0.059 และ -0.386 ดังนั้น Zscore(Q2) มีค่า Factor Coefficient สูงสุดอยู่ที่ปัจจัยที่ 1 แสดงว่าตัวแปร Zscore(Q2) ควรอยู่ในปัจจัยที่ 1 การสกัดปัจจัยสามารถอธิบายความผันแปรของทุกตัวแปรสะสมได้ดังตารางที่ 3.8



ตารางที่ 3.8 ค่าความผันแปรของทุกตัวแปรสะสมที่อธิบายได้

Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	5.176	21.568	21.568	5.176	21.568	21.568	4.048	16.868	16.868
2	4.308	17.952	39.520	4.308	17.952	39.520	3.857	16.069	32.937
3	3.621	15.087	54.608	3.621	15.087	54.608	3.815	15.896	48.832
4	2.936	12.235	66.843	2.936	12.235	66.843	3.207	13.364	62.197
5	1.952	8.134	74.977	1.952	8.134	74.977	3.067	12.780	74.977
6	.881	3.672	78.649						
7	.782	3.258	81.906						
8	.694	2.892	84.798						
9	.534	2.226	87.024						
10	.524	2.182	89.206						
11	.419	1.747	90.953						
12	.375	1.563	92.516						

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
13	.302	1.258	93.774						
14	.286	1.194	94.967						
15	.264	1.100	96.067						
16	.189	.786	96.853						
17	.163	.679	97.532						
18	.143	.595	98.127						
19	.127	.530	98.657						
20	.119	.494	99.151						
21	.088	.365	99.516						
22	.063	.264	99.780						
23	.034	.140	99.920						
24	.019	.080	100.000						

Extraction Method: Principal Component Analysis.

จากตารางที่ 3.8 แสดงค่าความผันแปรแต่ละปัจจัยที่มีอยู่ 24 ตัวแปร (24 ตัวชี้วัด) จากค่า Initial Eigenvalues เป็นค่าที่แสดงค่าความผันแปรของตัวแปรเดิมซึ่งสามารถอธิบายได้ว่าปัจจัยที่ 1 มีความผันแปรมากที่สุด เท่ากับ 1.576 ปัจจัยที่ 2 ความผันแปร เท่ากับ 4.308 ปัจจัยที่ 3 ความผันแปร เท่ากับ 3.621 ปัจจัยที่ 4 ความผันแปร เท่ากับ 2.936 ปัจจัยที่ 5 ความผันแปร เท่ากับ 1.952 ซึ่งจะเห็นว่าค่า Initial Eigenvalues ที่มากกว่า 1 มีอยู่ 5 ค่า ดังนั้นการสกัดปัจจัยใหม่ได้ทั้งหมด 5 ปัจจัย และ ความผันแปรของตัวแปรสะสมเท่ากับ 74.977% หมายความว่า การสกัดปัจจัยนี้มีความน่าเชื่อถือถึง 74.977%

3) การหมุนแกน (Rotation) การจัดตัวแปรโดยไม่หมุนแกนนั้น อาจทำได้ไม่ชัดเจนเนื่องจากค่า Factor Coefficient ของตัวแปรอาจมีค่าไม่ชัดเจนทำให้การจัดตัวแปรในแต่ละปัจจัยได้ลำบาก จึงจะต้องมีการหมุนแกนเพื่อให้ค่า Factor Coefficient ของตัวแปรบางตัวมีค่ามากขึ้นหรือน้อยลงทำให้การจัดตัวแปรในแต่ละปัจจัยชัดเจนและง่ายขึ้น ซึ่งผู้วิจัยได้ทำการหมุนแกนแบบ Varimax เป็นวิธีการหาค่า Factor Coefficient ของตัวแปรเดิมที่มีค่าสูงขึ้นในจำนวนตัวแปรที่น้อยที่สุด การหมุนแกนแบบ Varimax ซึ่งแสดงค่า Factor Coefficient ในตารางที่ 3.9

ตารางที่ 3.9 ผลการหมุนแกน (Rotation)

Rotated Component Matrix^a

	Component				
	1	2	3	4	5
Zscore(Q2)	-.101	.059	.922	.137	.098
Zscore(Q3)	-.091	.064	.954	.153	.124
Zscore(Q4)	-.025	.031	.923	.140	-.002
Zscore(Q5)	-.124	.074	.926	.168	.108
Zscore(Q6)	.085	-.012	.075	-.033	.896
Zscore(Q7)	.322	.053	-.083	-.123	.682
Zscore(Q8)	.228	.015	.033	.337	.770
Zscore(Q9)	-.143	-.022	.288	-.033	.721
Zscore(Q10)	-.034	-.043	.037	-.114	.726
Zscore(Q11)	.021	.104	.212	.834	-.025
Zscore(Q12)	.064	.053	.314	.871	-.040
Zscore(Q13)	.055	-.032	.175	.772	-.195

	Component				
	1	2	3	4	5
Zscore(Q14)	-.091	-.015	.034	.770	.217
Zscore(Q15)	.137	.282	-.068	.489	-.073
Zscore(Q16)	.887	-.031	-.065	.098	.072
Zscore(Q17)	.917	.050	-.022	-.102	-.015
Zscore(Q18)	.900	-.098	-.138	.003	.097
Zscore(Q19)	.820	.047	-.106	.083	.066
Zscore(Q20)	.804	-.045	-.002	.083	.079
Zscore(Q21)	-.093	.901	.020	-.022	.044
Zscore(Q22)	.091	.657	.001	.234	-.002
Zscore(Q23)	-.088	.932	.024	.036	.002
Zscore(Q24)	-.092	.959	.025	.038	-.005
Zscore(Q25)	.081	.833	.171	.010	-.057

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.^a

a. Rotation converged in 6 iterations.

จากตารางที่ 3.9 ผลของการหมุนแกน พบว่าการหมุนแกนใช้จำนวนการหมุนทั้งสิ้น 6 รอบ ได้ 5 ปัจจัย โดยแต่ละปัจจัยสามารถกำหนดตัวแปรต่าง ๆ ได้ผลการจัดตัวแปรให้กับปัจจัยได้ดังนี้

ปัจจัยที่ 1 (F1) มีตัวแปร คือ ZQ16, ZQ17, ZQ18, ZQ19, ZQ20

ปัจจัยที่ 2 (F2) มีตัวแปร คือ ZQ21, ZQ22, ZQ23, ZQ24, ZQ25

ปัจจัยที่ 3 (F3) มีตัวแปร คือ ZQ2, ZQ3, ZQ4, ZQ5

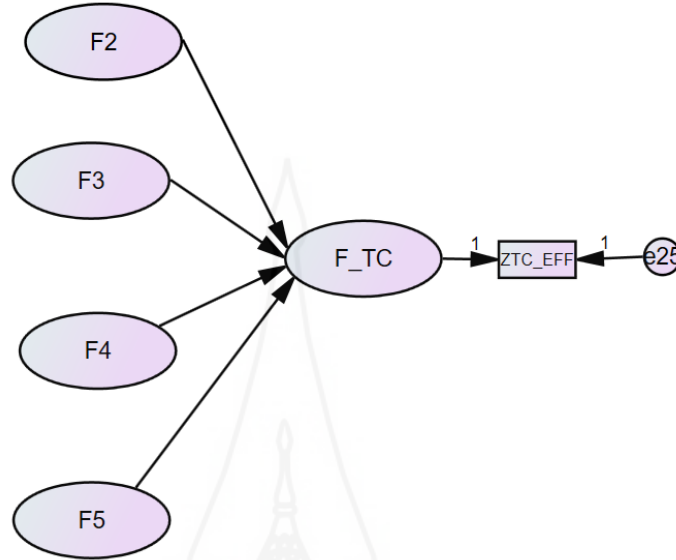
ปัจจัยที่ 4 (F4) มีตัวแปร คือ ZQ11, ZQ12, ZQ13, ZQ14, ZQ15

ปัจจัยที่ 5 (F5) มีตัวแปร คือ ZQ6, ZQ7, ZQ8, ZQ9, ZQ10

3.4.4 การพัฒนาแบบจำลองสมการโครงสร้าง (Structural Equation Model) หลังจากผู้วิจัยได้วิเคราะห์ข้อมูลการสัปดาห์ปัจจัย ได้ทำการจัดกลุ่มของตัวแปรต่าง ๆ ออกเป็นปัจจัยหลักจำนวน 5 ปัจจัย นำมาสร้างเป็นแบบจำลองสมการโครงสร้าง โดยมีการดำเนินการดังนี้

1) สมมุติฐานเกี่ยวกับความสัมพันธ์ของแบบจำลอง (Hypothesis Model) ในการวิจัยครั้งนี้ได้นำแบบสอบถามไปสัปดาห์ปัจจัย ได้ปัจจัยมาจำนวนหนึ่ง โดยเทคนิคการหมุนแกน เมื่อได้วัตถุประสงค์ของงานวิจัยที่ตั้งไว้ คือทุกตัวชี้วัดเป็นอิสระที่ชี้แนะตัวแปรตามการทดสอบ

ซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโใจล์ ดังนั้น สมมุติฐานในแบบจำลองที่นำเสนอจึงมีตัวอย่างตามภาพที่ 3.10



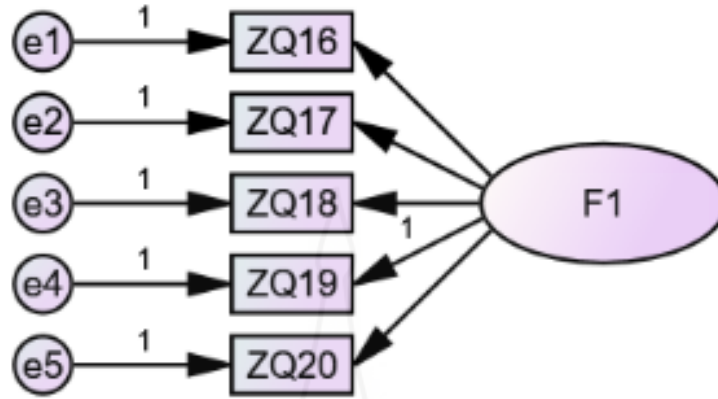
ภาพที่ 3.10 สมมุติฐานแบบจำลองวัดองค์ประกอบการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโใจล์

2) คุณลักษณะแบบจำลอง (Model Specification) ผลจากการสกัดปัจจัยได้ตัวชี้วัดในแต่ละปัจจัย ดังตารางที่ 3.10

ตารางที่ 3.10 ตัวชี้วัดในแต่ละปัจจัย

ปัจจัยหลัก	ชื่อปัจจัย	ตัวชี้วัด
F1	การพัฒนาาระบบ	ZQ16, ZQ17, ZQ18, ZQ19, ZQ20
F2	การขึ้นระบบจริง	ZQ21, ZQ22, ZQ23, ZQ24, ZQ25
F3	การทดสอบระบบแบบอัตโนมัติ	ZQ2, ZQ3, ZQ4, ZQ5
F4	การวิเคราะห์ระบบ	ZQ11, ZQ12, ZQ13, ZQ14, ZQ15
F5	การรับความต้องการของระบบ	ZQ6, ZQ7, ZQ8, ZQ9, ZQ10

การระบุคุณลักษณะแบบจำลองของแต่ละปัจจัย (ปัจจัยแฝง) สามารถกำหนดได้ดังนี้



ภาพที่ 3.11 ตัวแปรที่อยู่ในปัจจัย F1

$$\xi_1 = F1 = 1 \times ZQ16 + \lambda_{ZQ17,1}ZQ17 + \lambda_{ZQ18,1}ZQ18 + \lambda_{ZQ19,1}ZQ19 + \lambda_{ZQ20,1}ZQ20 \quad (3.2)$$

$$ZQ16 = \xi_1 + \delta_{ZQ16}(e_1) \quad (3.3)$$

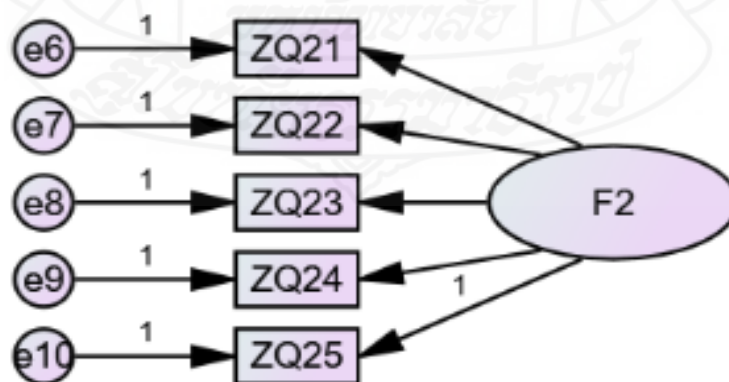
$$ZQ17 = \lambda_{ZQ17,1} \xi_1 + \delta_{ZQ17}(e_2) \quad (3.4)$$

$$ZQ18 = \lambda_{ZQ18,1} \xi_1 + \delta_{ZQ18}(e_3) \quad (3.5)$$

$$ZQ19 = \lambda_{ZQ19,1} \xi_1 + \delta_{ZQ19}(e_4) \quad (3.6)$$

$$ZQ20 = \lambda_{ZQ20,1} \xi_1 + \delta_{ZQ20}(e_5) \quad (3.7)$$

โดยที่ λ_{ZQ16} $\lambda_{1,1}$ กำหนดให้เป็น 1 เป็นการให้ค่าคงที่ตัวแปร ZQ16 ให้เป็นหน่วยหนึ่งของ ξ_1



ภาพที่ 3.12 ตัวแปรที่อยู่ในปัจจัย F2

$$\xi_2 = F2 = 1 \times ZQ21 + \lambda_{ZQ22,1} ZQ22 + \lambda_{ZQ23,1} ZQ23 + \lambda_{ZQ24,1} ZQ24 + \lambda_{ZQ25,1} ZQ25 \quad (3.8)$$

$$ZQ21 = \xi_2 + \delta_{ZQ21}(e_6) \quad (3.9)$$

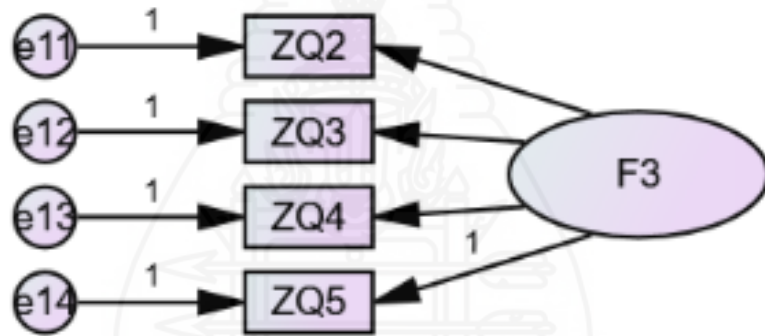
$$ZQ22 = \lambda_{ZQ22,1} \xi_1 + \delta_{ZQ22}(e_7) \quad (3.10)$$

$$ZQ23 = \lambda_{ZQ23,1} \xi_1 + \delta_{ZQ23}(e_8) \quad (3.11)$$

$$ZQ24 = \lambda_{ZQ24,1} \xi_1 + \delta_{ZQ24}(e_9) \quad (3.12)$$

$$ZQ25 = \lambda_{ZQ25,1} \xi_1 + \delta_{ZQ25}(e_{10}) \quad (3.13)$$

โดยที่ λ_{ZQ21} λ_{1_2} กำหนดให้เป็น 1 เป็นการให้ค่าคงที่ตัวแปร ZQ21 ให้เป็นหน่วยหนึ่งของ ξ_2



ภาพที่ 3.13 ตัวแปรที่อยู่ในปัจจัย F3

$$\xi_3 = F3 = 1 \times ZQ2 + \lambda_{ZQ3,1} ZQ3 + \lambda_{ZQ4,1} ZQ4 + \lambda_{ZQ5,1} ZQ5 \quad (3.14)$$

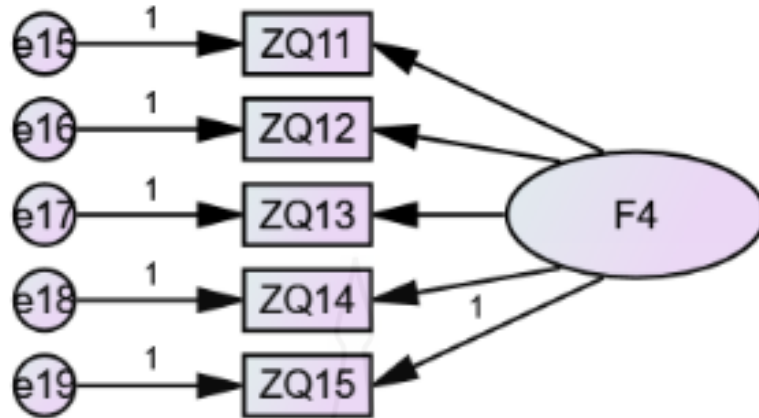
$$ZQ2 = \xi_3 + \delta_{ZQ2}(e_{11}) \quad (3.15)$$

$$ZQ3 = \lambda_{ZQ3,1} \xi_1 + \delta_{ZQ3}(e_{12}) \quad (3.16)$$

$$ZQ4 = \lambda_{ZQ4,1} \xi_1 + \delta_{ZQ4}(e_{13}) \quad (3.17)$$

$$ZQ5 = \lambda_{ZQ5,1} \xi_1 + \delta_{ZQ5}(e_{14}) \quad (3.18)$$

โดยที่ λ_{ZQ2} λ_{1_3} กำหนดให้เป็น 1 เป็นการให้ค่าคงที่ตัวแปร ZQ2 ให้เป็นหน่วยหนึ่งของ ξ_3



ภาพที่ 3.14 ตัวแปรที่อยู่ในปัจจัย F4

$$\xi_4 = F4 = 1 \times ZQ11 + \lambda_{ZQ12,1} ZQ12 + \lambda_{ZQ13,1} ZQ13 + \lambda_{ZQ14,1} ZQ14 + \lambda_{ZQ15,1} ZQ15 \quad (3.19)$$

$$ZQ11 = \xi_4 + \delta_{ZQ11}(e_{15}) \quad (3.20)$$

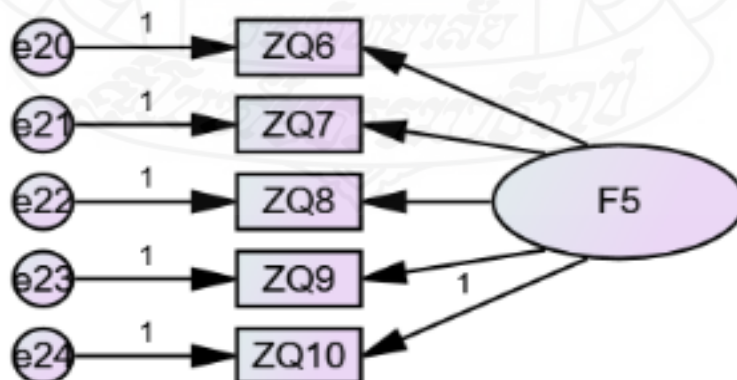
$$ZQ12 = \lambda_{ZQ12,1} \xi_4 + \delta_{ZQ12}(e_{16}) \quad (3.21)$$

$$ZQ13 = \lambda_{ZQ13,1} \xi_4 + \delta_{ZQ13}(e_{17}) \quad (3.22)$$

$$ZQ14 = \lambda_{ZQ14,1} \xi_4 + \delta_{ZQ14}(e_{18}) \quad (3.23)$$

$$ZQ15 = \lambda_{ZQ15,1} \xi_4 + \delta_{ZQ15}(e_{19}) \quad (3.24)$$

โดยที่ λ_{ZQ11} λ_{1_4} กำหนดให้เป็น 1 เป็นการให้ค่าคงที่ตัวแปร ZQ11 ให้เป็นหน่วยหนึ่งของ ξ_4



ภาพที่ 3.15 ตัวแปรที่อยู่ในปัจจัย F5

$$\xi_5 = F_5 = 1 \times ZQ6 + \lambda_{ZQ7,1}ZQ7 + \lambda_{ZQ8,1}ZQ8 + \lambda_{ZQ9,1}ZQ9 + \lambda_{ZQ10,1}ZQ10 \quad (3.25)$$

$$ZQ6 = \xi_4 + \delta_{ZQ6}(e_{20}) \quad (3.26)$$

$$ZQ7 = \lambda_{ZQ7,1} + \xi_1 + \delta_{ZQ7}(e_{21}) \quad (3.27)$$

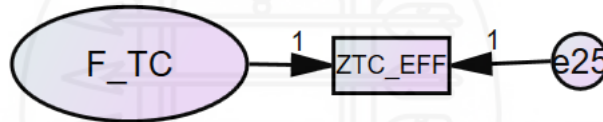
$$ZQ8 = \lambda_{ZQ8,1} + \xi_1 + \delta_{ZQ8}(e_{22}) \quad (3.28)$$

$$ZQ9 = \lambda_{ZQ9,1} + \xi_1 + \delta_{ZQ9}(e_{23}) \quad (3.29)$$

$$ZQ10 = \lambda_{ZQ10,1} + \xi_1 + \delta_{ZQ10}(e_{24}) \quad (3.30)$$

โดยที่ λ_{ZQ6} λ_{1_5} กำหนดให้เป็น 1 เป็นการให้ค่าคงที่ตัวแปร ZQ6 ให้เป็นหน่วยหนึ่งของ ξ_5

สำหรับปัจจัยที่ผู้วิจัยได้กำหนดไว้ในขั้นตอนการเก็บรวบรวมข้อมูลการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโงิล คือ ขนาดของซอฟต์แวร์ โดยวัดจากจำนวนกรณีทดสอบ ได้ถูกแทนค่าด้วยตัวแปรแฝง ZTC_EFF ผลการผิดพลาดจากการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ก ได้ถูกแทนค่าด้วยตัวแปรแฝง ZTC_ERR และ เวลาที่ใช้ในการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ก ได้ถูกแทนค่าด้วยตัวแปรแฝง ZTC_TIME ซึ่งมี Model Specification ตามลำดับดังนี้

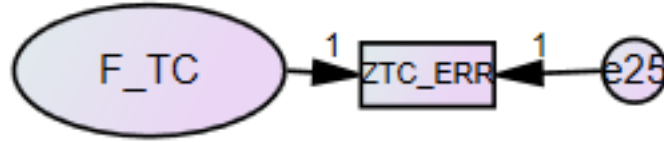


ภาพที่ 3.16 Model Specification ของ ZTC_Eff

แทนด้วยสัญลักษณ์ตามที่กำหนด สามารถแทนด้วยสมการ

$$n_1 = \lambda_{TC_Eff, F_TC} TC_Eff + \varepsilon_{e25} (e_{25}) \quad (3.31)$$

$n_1 = \lambda_{ZscoreF_TC, Score} ZscoreF_TC + \varepsilon_{e25}$ และตัวแปรแฝง F_TC มีตัวแปรประจักษ์เพียง 1 ตัว คือ ZTC_Eff นั่นคือ λ_{TC_Eff, F_TC} จะถูกกำหนดให้มีค่าเป็น 1

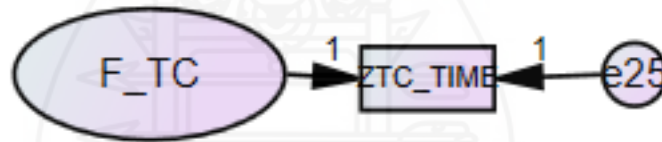


ภาพที่ 3.17 Model Specification ของ ZTC_ERR

แทนด้วยสัญลักษณ์ตามที่กำหนด สามารถแทนด้วยสมการ

$$n_1 = \lambda_{TC_ERR, F_TC} TC_ERR + \varepsilon_{e25} (e_{25}) \quad (3.32)$$

$n_1 = \lambda_{ZScoreF_TC, ScoreZscoreF_TC} + \varepsilon_{e25}$ และตัวแปรแฝง F_TC มีตัวแปรประจักษ์เพียง 1 ตัว คือ ZTC_ERR นั่นคือ λ_{TC_ERR, F_TC} จะถูกกำหนดให้มีค่าเป็น 1

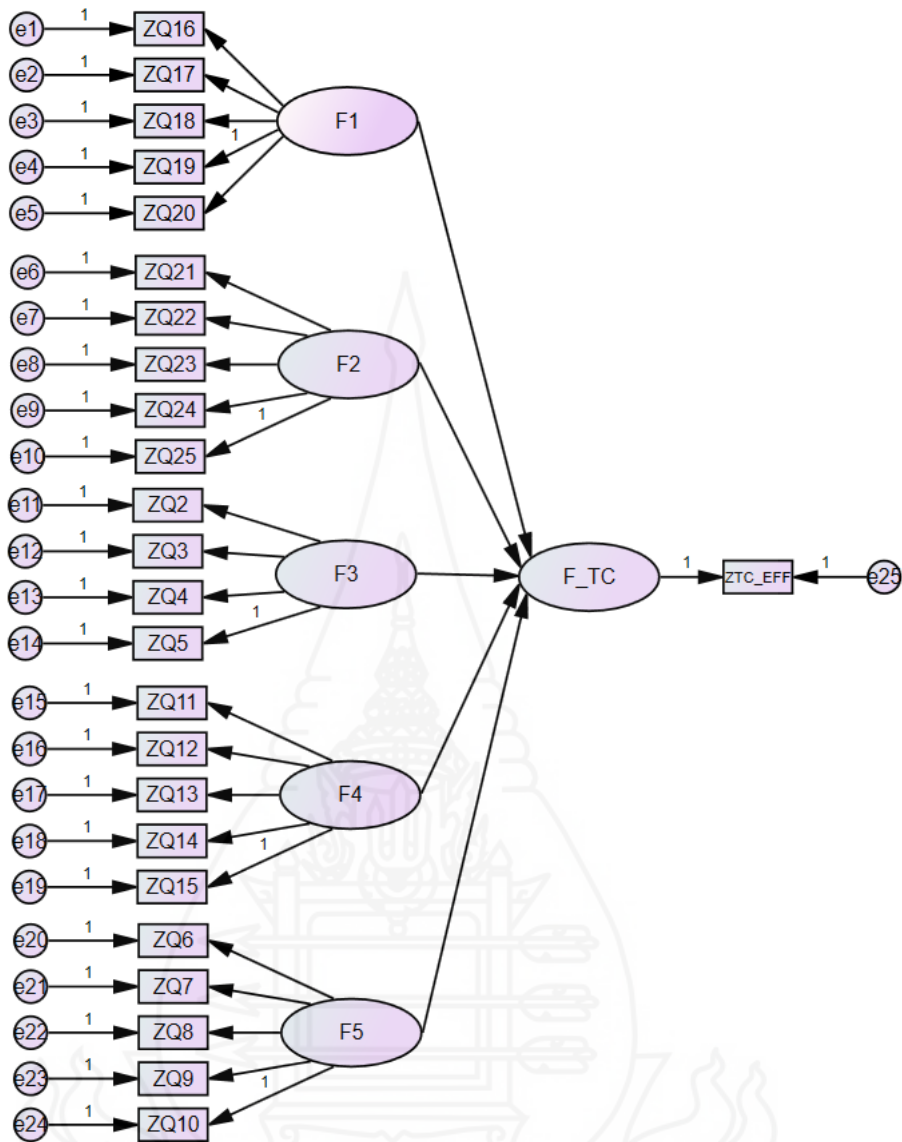


ภาพที่ 3.18 Model Specification ของ ZTC_TIME

แทนด้วยสัญลักษณ์ตามที่กำหนด สามารถแทนด้วยสมการ

$$n_1 = \lambda_{TC_TIME, F_TC} TC_TIME + \varepsilon_{e25} (e_{25}) \quad (3.33)$$

$n_1 = \lambda_{ZScoreF_TC, ScoreZscoreF_TC} + \varepsilon_{e25}$ และตัวแปรแฝง F_TC มีตัวแปรประจักษ์เพียง 1 ตัว คือ ZTC_TIME นั่นคือ λ_{TC_TIME, F_TC} จะถูกกำหนดให้มีค่าเป็น 1



ภาพที่ 3.19 แบบจำลองสมการโครงสร้างตั้งต้น

3.4.5 การประเมินความเหมาะสมของแบบจำลอง ผู้วิจัยนำตัวแปรในแต่ละปัจจัย มาวิเคราะห์ว่าปัจจัยใดบ้างที่ผ่านหรือไม่ผ่าน โดยเกณฑ์การประเมินความสอดคล้องของโมเดล ตั้งแบบกับข้อมูลเชิงประจักษ์ที่ต้องนำมาพิจารณาค้างนี้

ตารางที่ 3.11 เกณฑ์การประเมินความสอดคล้องของโมเดล

การประเมินความสอดคล้องของโมเดล		เกณฑ์
CMIN-P	ค่าระดับความน่าจะเป็นของไคสแควร์	$P > 0.05$
CMIN/DF	ค่าไคสแควร์สัมพัทธ์	$CMIN/DF < 3$
GFI	ค่าดัชนีวัดระดับความสอดคล้อง	$GFI > 0.90$
RMSEA	ค่าดัชนีรากของค่าเฉลี่ยกำลังสองของการประมาณค่าความคลาดเคลื่อน	$RMSEA < 0.08$

ผลการวิเคราะห์ ได้แก่ ค่าประมาณ (Estimate) ของสัมประสิทธิ์ความสัมพันธ์ (Regression Weight) ระหว่างตัวแปรต่าง ๆ ทั้งหมดในแบบจำลอง หากไม่มีความสำคัญทางสถิติแบบนัยสำคัญต่อกัน (ค่า P มากกว่า 0.05 สมควรตัดออกจากแบบจำลองและเมื่อตัดออกจะต้องทำการวิเคราะห์ใหม่) ดังตารางที่ 3.11

ตารางที่ 3.12 ค่าประมาณการ (Estimate) ของสัมประสิทธิ์ความสัมพันธ์ (Regression Weight) (Regression Weight)

เส้นการเชื่อมโยง	Estimate	S.E.	C.R.	P
F_TC <--- F1	0.091	0.138	0.66	0.509
F_TC <--- F2	-0.148	0.145	-1.021	0.307
F_TC <--- F3	0.002	0.111	0.016	0.988
F_TC <--- F4	-0.348	0.29	-1.199	0.23
F_TC <--- F5	0.064	0.196	0.327	0.743

เมื่อประมวลผลแล้ว ตรวจสอบค่า Model Fit Summary พบว่าค่า $P = 0.00$ หมายความว่าแบบจำลองไม่สอดคล้องกับข้อมูลประจักษ์ ดังตารางที่ 3.13

ตารางที่ 3.13 ค่า Model Fit Summary

Model	NPAR	CMIN	DF	P	CMIN/DF
Default model	54	422.287	271	.000	1.558
Saturated model	325	.000	.000		

Model	NPAR	CMIN	DF	P	CMIN/DF
Independence model	25	2092.997	300	.000	6.977

การปรับปรุงแบบจำลอง ระบุความต้องการที่ Modification indices โดยพิจารณาความสัมพันธ์โดยการเชื่อมโยงความสัมพันธ์แบบ Two-Direction ระหว่างตัวแปรที่ทำให้ค่า χ^2 ลดลงมากที่สุดเพื่อให้ในการปรับปรุงสมการ

เพื่อตรวจสอบผลการดำเนินงาน โดยพิจารณาจาก Model Fit Summary โดยจะตรวจสอบวัดความเหมาะสมของค่าต่าง ๆ ดังนี้

P-value for test of close fit ควรค่ามากกว่า 0.05 หากค่า P-value for test of close fit มีค่ามากกว่า 0.05 ก็ยังสามารถทำ SEM ได้ในกรณีที่ N มากกว่า 200 หรือจำนวนกลุ่มตัวอย่างมีขนาดโตมาก ให้พิจารณาที่ค่า GFI, AGFI, RMSEA ให้อยู่ในข้อมูลที่กำหนด

Goodness of fit Index (GFI) ค่าที่สอดคล้องกับข้อมูลควรมากกว่า 0.9

Root Mean Square Error (RMSE) ค่าที่สอดคล้องกับข้อมูลควรน้อยกว่า 0.8
ดังตารางที่ 3.14

ตารางที่ 3.14 ค่า Modification indices

เส้นการเชื่อมโยง	M.I	Per Change
F3 <---> F4	6.718	0.139
F1 <---> F3	6.449	0.041
e22 <---> e18	6.864	0.174

ตารางที่ 3.15 ค่า CMIN

Model	NPAR	CMIN	DF	P	CMIN/DF
Default model	35	71.252	70	0.436	1.018
Saturated model	105	.000	0		
Independence model	14	818.043	91	.000	8.989

ตารางที่ 3.16 ค่า GFI และ AGFI

Model	RMR	GFI	AGFI	PGFI
Default model	0.091	0.905	0.857	0.603
Saturated model	.000	1.000		
Independence model	0.252	0.506	0.43	0.439

ตารางที่ 3.17 ค่า RMSEA

Model	RMSEA	LO 90	HI 90	PCLOSE
Default model	0.014	.000	0.065	0.839
Independence model	0.303	0.284	0.322	.000

3.4.6 การประมาณค่าเฉลี่ยปริมาณผิดพลาดเชิงสัมพัทธ์ (MMRE)

เมื่อได้สมการของแบบจำลอง การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค ภายใต้รูปแบบการทำงานแบบอโจล์ ผู้วิจัยได้ใช้สมการค่าตัวแปร ZTC_EFF, ZTC_ERR, ZTC_TIME ของกลุ่มตัวอย่าง 88 ตัวอย่างนำค่าตัวแปรอิสระคำนวณหาค่าตัวแปร F_TC ผลลัพธ์ที่ได้จะเป็นค่าประมาณของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค ภายใต้รูปแบบการทำงานแบบอโจล์ แล้วนำไปเปรียบเทียบกับค่าของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโจล์ที่ได้จากแบบสอบถาม เพื่อนำมาคำนวณหา MRE ของแบบสอบถามตัวอย่าง จนครบ 30 ตัวอย่าง แล้วทำการหาค่าเฉลี่ย MMRE ได้เป็นค่าร้อยละความคลาดเคลื่อนของการประมาณการของแบบจำลอง ซึ่งค่าที่ได้จะแสดงในบทถัดไป

บทที่ 4 ผลการวิจัย

การวิจัยครั้งนี้ผู้วิจัยได้ดำเนินการวิจัยของการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบอใจล์และได้สรุปผลการวิจัยตามขั้นตอนดังนี้

- 4.1 ผลการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบอใจล์
- 4.2 ผลการวิเคราะห์องค์ประกอบประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบอใจล์
- 4.3 การประมาณค่าความแม่นยำ

4.1 ผลการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบอใจล์

ผลจากการเก็บข้อมูลการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและแบบมือภายใต้รูปแบบการทำงานแบบอใจล์ โดยแบ่งกลุ่มการทดสอบออกเป็น 2 กลุ่ม คือ กลุ่มที่ทำ การทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โปรเจค และกลุ่มการทดสอบซอฟต์แวร์มากกว่า 1 ครั้ง ต่อ 1 โปรเจค หรือเรียกว่าการทดสอบแบบถดถอย (Regression Testing) สามารถวิเคราะห์ผลได้ดังนี้

ตารางที่ 4.1 ผลการทดสอบของกลุ่มที่ทำการทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โปรเจก

จำนวน การ ทดสอบ	จำนวน กรณี ทดสอบ รวม	เวลาที่ใช้		ผลการทดสอบสำเร็จ (ข้อ)		ผลการทดสอบ ผิดพลาด (ข้อ)		ผลการทดสอบสำเร็จ (%)		ผลการทดสอบ ผิดพลาด (%)	
		ทดสอบแบบ อัตโนมัติ	ทดสอบแบบ มือ	อัตโนมัติ	แบบ มือ	อัตโนมัติ	แบบมือ	อัตโนมัติ	แบบมือ	อัตโนมัติ	แบบ มือ
		34	769	12:02:23	19:17:27	760	748	9	21	98.83	97.27

จากตารางที่ 4.1 สามารถอธิบายได้ว่าการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและแบบมือภายใต้รูปแบบการทำงานแบบอโงิล์ของกลุ่มที่ทำการทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โปรเจก มีจำนวนการทดสอบทั้งหมด 34 ครั้ง โดยมีกรณีทดสอบรวมทั้งหมด 769 กรณีทดสอบ พบว่า

1. การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโงิล์ ใช้เวลาในการทำการทดสอบรวมทั้งสิ้น 12:02:23 นาที และ การทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโงิล์ ใช้เวลาในการทำการทดสอบรวมทั้งสิ้น 19:17:27 นาที โดยที่ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโงิล์สามารถลดเวลาในการทดสอบซอฟต์แวร์ทั้งสิ้น 07:15:04 นาทีในการทำแบบทดสอบทั้งหมด 769 กรณีทดสอบ

2. การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโงิล์ ได้ผลสำเร็จรวมทั้งสิ้น 760 ข้อ โดยคิดเป็นร้อยละ 98.83 ของกรณีทดสอบทั้งหมด และ การทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโงิล์ ได้ผลสำเร็จรวมทั้งสิ้น 748 ข้อ โดยคิดเป็นร้อยละ 97.27 ของกรณีทดสอบทั้งหมด โดยที่ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโงิล์ มีค่าความสำเร็จมากกว่าการทดสอบแบบมืออยู่ที่ 12 ข้อ โดยคิดเป็นร้อยละ 1.56 ของผลสำเร็จในการทดสอบ

3. การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ ได้ผลข้อผิดพลาดรวมทั้งสิ้น 9 ข้อ โดยคิดเป็นร้อยละ 1.17 ของกรณีทดสอบทั้งหมด และ การทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโต้ ได้ผลข้อผิดพลาดรวมทั้งสิ้น 21 ข้อ โดยคิดเป็นร้อยละ 2.73 ของกรณีทดสอบทั้งหมด โดยที่ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ มีค่าความผิดพลาดน้อยกว่าการทดสอบแบบมืออยู่ที่ 12 ข้อ โดยคิดเป็นร้อยละ 1.56 ของผลความผิดพลาดในการทดสอบ

สรุปได้ว่า การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ แบบ 1 ครั้ง ต่อ 1 โปรเจกต์ ใช้เวลาในการทดสอบที่น้อยกว่า มีค่าความสำเร็จที่สูงกว่า และค่าความผิดพลาด น้อยกว่าการทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโต้

ตารางที่ 4.2 ผลการทดสอบของกลุ่มทดสอบซอฟต์แวร์มากกว่า 1 ครั้ง ต่อ 1 โปรเจกต์ หรือเรียกว่าการทดสอบแบบถดถอย (Regression Testing)

จำนวน การ ทดสอบ	จำนวน กรณี ทดสอบ รวม	เวลาที่ใช้		ผลการทดสอบสำเร็จ (ข้อ)		ผลการทดสอบ ผิดพลาด (ข้อ)		ผลการทดสอบสำเร็จ (%)		ผลการทดสอบ ผิดพลาด (%)	
		ทดสอบ แบบ อัตโนมัติ	ทดสอบแบบ มือ	อัตโนมัติ แบบ มือ	แบบ มือ	อัตโนมัติ แบบ มือ	แบบ มือ	อัตโนมัติ แบบ มือ	แบบ มือ		
74	3,685	53:11:49	165:44:35	3,810	3,713	55	152	98.58	96.07	1.42	3.93

จากตารางที่ 4.2 สามารถอธิบายได้ว่าในการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กและแบบมือภายใต้รูปแบบการทำงานแบบอโต้ ของกลุ่มที่ทำการทดสอบซอฟต์แวร์มากกว่า 1 ครั้ง ต่อ 1 โปรเจกต์ หรือเรียกว่าการทดสอบแบบถดถอย (Regression Testing) มีจำนวนการทดสอบทั้งหมด 74 ครั้ง โดยมีกรณีทดสอบรวมทั้งสิ้น 3,685 กรณีทดสอบ พบว่า

1) การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ ใช้เวลาในการทำการทดสอบรวมทั้งสิ้น 53:11:49 นาที และ การทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโต้ ใช้เวลาในการทำการทดสอบรวมทั้งสิ้น 19:17:27 นาที โดยที่ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้สามารถลดเวลาในการทดสอบซอฟต์แวร์ทั้งสิ้น 165:44:35 นาทีในการทำแบบทดสอบทั้งหมด 3,685 กรณีทดสอบ

2) การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ ได้ผลสำเร็จรวมทั้งสิ้น 3,810 ข้อ โดยคิดเป็นร้อยละ 98.58 ของกรณีทดสอบทั้งหมด และ การทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโต้ ได้ผลสำเร็จรวมทั้งสิ้น 3,713 ข้อ โดยคิดเป็นร้อยละ 96.07 ของกรณีทดสอบทั้งหมด โดยที่ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ มีค่าความสำเร็จมากกว่าการทดสอบแบบมืออยู่ที่ 97 ข้อ โดยคิดเป็นร้อยละ 2.51 ของผลสำเร็จในการทดสอบ

3) การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ ได้ผลข้อผิดพลาดรวมทั้งสิ้น 55 ข้อ โดยคิดเป็นร้อยละ 1.42 ของกรณีทดสอบทั้งหมด และ การทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโต้ ได้ผลข้อผิดพลาดรวมทั้งสิ้น 152 ข้อ โดยคิดเป็นร้อยละ 3.93 ของกรณีทดสอบทั้งหมด โดยที่ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ มีค่าความผิดพลาดน้อยกว่าการทดสอบแบบมืออยู่ที่ 97 ข้อ โดยคิดเป็นร้อยละ 2.51 ของผลความผิดพลาดในการทดสอบ

สรุปได้ว่า การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบอโต้ ทั้งแบบ 1 ครั้ง ต่อ 1 โปรเจก และ แบบมากกว่า 1 ครั้ง ต่อ 1 โปรเจก หรือเรียกว่า การทดสอบแบบถดถอย (Regression Testing) ใช้เวลาในการทดสอบที่น้อยกว่า มีค่าความสำเร็จที่สูงกว่า และค่าความผิดพลาด น้อยกว่าการทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบอโต้

4.1.1 การประมาณค่าความแม่นยำของการทดสอบซอฟต์แวร์ภายใต้รูปแบบการทำงานแบบอโต้

หลักจากผู้วิจัยได้ทำการสรุปผลการทดสอบซอฟต์แวร์ ผู้วิจัยได้นำค่า Time used ของการทดสอบซอฟต์แวร์ทั้งการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กและการทดสอบซอฟต์แวร์แบบมือ มาเปรียบเทียบกันเพื่อคำนวณหาค่า MRE จนครบ 108 ตัวอย่าง จากนั้นทำการหาค่า MMRE (Average absolute MRE 100%) ได้เป็นค่าร้อยละของความคลาดเคลื่อนเฉลี่ย ซึ่งได้แสดงผลตามตารางที่ 4.1, 4.2, 4.3 และ 4.4 ตามลำดับ

ตารางที่ 4.3 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์
เท่ากับ 1 ครั้ง ต่อ 1 โปรแกรม

ผลการคำนวณค่า MRE และ MMRE		
จำนวนการทดสอบ 1 ครั้ง ต่อ 1 โปรแกรม		
เวลาที่ใช้ในการทดสอบแบบมือ (นาที)	เวลาที่ใช้ในการทดสอบแบบอัตโนมัติ (นาที)	MRE
32.00	10.52	0.67
37.11	0.20	0.99
8.12	8.12	0.00
265.31	13.12	0.95
135.21	2.05	0.98
85.13	19.17	0.77
32.00	10.52	0.67
67.21	17.49	0.74
101.34	21.11	0.79
43.19	12.17	0.72
33.03	17.01	0.49
36.57	17.49	0.52
39.43	7.12	0.82
82.27	47.21	0.43
82.31	3.00	0.96
35.27	9.55	0.73
89.14	41.42	0.54
81.56	41.02	0.50
45.00	1.20	0.97
95.33	41.23	0.57
103.03	21.22	0.79
109.55	22.52	0.79
15.22	0.10	0.99
56.12	23.31	0.58
174.52	56.52	0.68
165.00	59.35	0.64
180.00	120.01	0.33
185.00	59.58	0.68

ผลการคำนวณค่า MRE และ MMRE		
จำนวนการทดสอบ 1 ครั้ง ต่อ 1 โป้รเจค		
เวลาที่ใช้ในการทดสอบแบบมือ (นาทึ)	เวลาที่ใช้ในการทดสอบแบบอัตโนมัติ (นาทึ)	MRE
35.45	19.41	0.45
35.45	19.00	0.46
107.00	35.29	0.67
81.03	41.22	0.49
75.00	41.25	0.45
20.00	0.31	0.98
	MMRE	19.76

ผลการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและแบบมือภายใต้รูปแบบการทำงานแบบบอจี้ล ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โป้รเจค พบว่า มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 19.76 %

ตารางที่ 4.4 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 2 ครั้ง ต่อ 1 โป้รเจค

ผลการคำนวณค่า MRE และ MMRE		
จำนวนการทดสอบ 2 ครั้ง ต่อ 1 โป้รเจค		
เวลาที่ใช้ในการทดสอบแบบมือ (นาทึ)	เวลาที่ใช้ในการทดสอบแบบอัตโนมัติ (นาทึ)	MRE
43.51	0.47	0.99
15.17	0.22	0.99
118.51	0.22	1.00
129.51	47.34	0.63
41.12	5.52	0.87
116.19	22.21	0.81
86.03	32.54	0.62
73.57	31.24	0.58
263.41	173.11	0.34
213.12	155.43	0.27
45.00	15.30	0.66
53.32	3.05	0.94

ผลการคำนวณค่า MRE และ MMRE		
จำนวนการทดสอบ 2 ครั้ง ต่อ 1 โป้รเจค		
เวลาที่ใช้ในการทดสอบแบบมือ (นาทึ)	เวลาที่ใช้ในการทดสอบแบบอัตโนมัติ (นาทึ)	MRE
106.26	47.34	0.55
110.21	47.27	0.57
156.12	133.23	0.15
39.10	7.19	0.82
45.00	3.21	0.93
53.00	12.17	0.77
49.50	0.21	1.00
356.43	270.21	0.24
32.14	7.54	0.77
346.05	285.42	0.18
35.01	21.44	0.39
330.22	269.54	0.18
363.46	271.31	0.25
155.31	45.32	0.71
392.21	270.55	0.31
176.19	172.21	0.02
84.30	47.45	0.44
45.00	24.12	0.46
54.05	28.12	0.48
48.57	25.22	0.48
50.00	8.23	0.84
	MMRE	19.22

ผลการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและแบบมือภายใต้รูปแบบการทำงานแบบอไ้จล์ ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 2 ครั้ง ต่อ 1 โป้รเจค พบว่า มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 19.22 %

ตารางที่ 4.5 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์
เท่ากับ 3 ครั้ง ต่อ 1 โปรแกรม

ผลการคำนวณค่า MRE และ MMRE		
จำนวนการทดสอบ 3 ครั้ง ต่อ 1 โปรแกรม		
เวลาที่ใช้ในการทดสอบแบบมือ (นาที)	เวลาที่ใช้ในการทดสอบแบบอัตโนมัติ (นาที)	MRE
195.43	42.57	0.78
237.17	43.37	0.82
422.21	129.02	0.69
445.56	128.57	0.71
308.42	147.01	0.52
320.34	149.45	0.53
345.56	152.11	0.56
366.34	151.57	0.59
201.54	121.02	0.40
212.11	59.33	0.72
192.59	120.01	0.38
185.23	59.33	0.68
181.02	59.53	0.67
135.43	49.12	0.64
139.43	49.33	0.65
132.43	47.01	0.65
135.43	48.58	0.64
135.21	2.09	0.98
121.24	2.07	0.98
212.18	124.02	0.42
108.39	22.12	0.80
157.51	57.34	0.64
151.23	59.51	0.61
141.51	49.32	0.65
	MMRE	15.70

ผลการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและแบบมือภายใต้รูปแบบการทำงานแบบอโใจล์ ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 3 ครั้ง ต่อ 1 โปรเจก พบว่า มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 15.70 %

ตารางที่ 4.6 ผลการคำนวณค่า MRE และ MMRE ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 4 ครั้ง ต่อ 1 โปรเจก

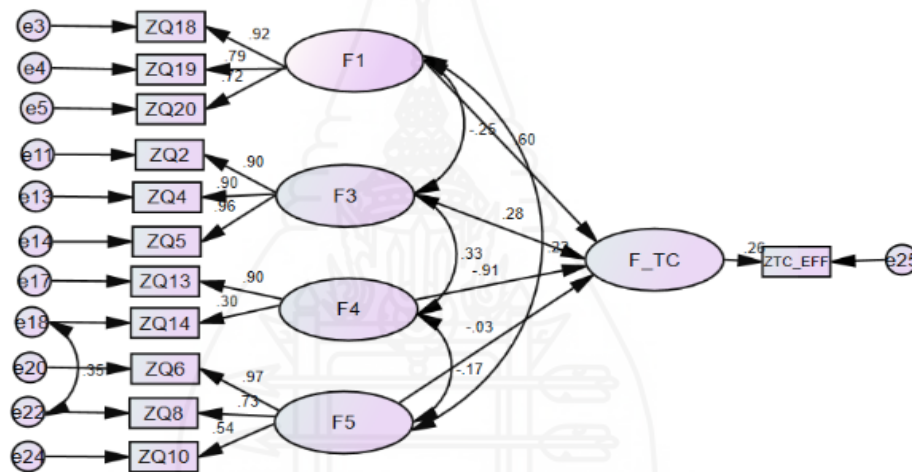
ผลการคำนวณค่า MRE และ MMRE		
จำนวนการทดสอบ 4 ครั้ง ต่อ 1 โปรเจก		
เวลาที่ใช้ในการทดสอบแบบมือ (นาที)	เวลาที่ใช้ในการทดสอบแบบอัตโนมัติ (นาที)	MRE
173.29	57.33	0.67
173.29	57.33	0.67
232.21	121.59	0.48
221.12	122.56	0.45
79.42	12.43	0.84
57.12	25.43	0.55
28.00	8.52	0.70
59.31	9.20	0.84
366.34	151.57	0.59
192.55	145.32	0.25
89.00	32.33	0.64
176.19	172.21	0.02
53.54	25.12	0.53
35.47	12.22	0.66
166.12	135.23	0.19
15.00	0.30	0.98
103.25	52.33	0.49
151.00	161.54	0.07
	MMRE	9.61

ผลการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คและแบบมือภายใต้รูปแบบการทำงานแบบอโใจล์ ของกลุ่มตัวอย่างที่ทำการทดสอบซอฟต์แวร์เท่ากับ 4 ครั้ง ต่อ 1 โปรเจก พบว่า มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 9.61 %

4.2 ผลการวิเคราะห์ห้องค์ประกอบประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบโอไจล์

ผลการวิเคราะห์จากการปรับปรุงตัวแบบสมการ จากกลุ่มตัวอย่างจำนวน 88 ตัวอย่าง ผลการวิจัยมีรายละเอียดได้เป็น 3 แบบจำลองสมการโครงสร้าง โดยผู้วิจัยได้ทำการปรับปรุงแบบจำลองสมมุติจนได้แบบจำลองที่เหมาะสมและผ่านเกณฑ์การพิจารณาความเหมาะสมดังต่อไปนี้

4.2.1 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบโอไจล์จากขนาดซอฟต์แวร์



ภาพที่ 4.1 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบโอไจล์

จากแบบจำลองสมการ โครงสร้างที่พัฒนาขึ้นจากการประเมินแบบสอบถาม พบว่ามี 4 ปัจจัยที่มีผลต่อประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบโอไจล์ ซึ่งมีดังนี้ F1 คือ การพัฒนาระบบ F3 คือ การทดสอบระบบแบบอัตโนมัติ F4 คือ การวิเคราะห์ระบบ F5 คือ การรับความต้องการของระบบ โดยแบบจำลองที่ผ่านการวิเคราะห์ได้ แสดงขนาดและทิศทางของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบโอไจล์ มีองค์ประกอบต่าง ๆ คือ จำนวนกรณีทดสอบต่อการทำงานแบบโอไจล์ (F_TC) คือ ค่าร้อยละของการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบโอไจล์ โดยสามารถเขียนสมการได้ ดังนี้

$$ZTC_EFF = 0.26 * F_TC \quad (4.1)$$

$$F_TC = F1 + 0.60 * F3 + 0.28 * F4 + (-0.91) * F5 + (-0.03) \quad (4.2)$$

$$F1 = 0.92 * ZQ18 + 0.79 * ZQ19 + 0.72 * ZQ20 \quad (4.3)$$

$$F3 = 0.90 * ZQ2 + 0.90 * ZQ4 + 0.96 * ZQ5 \quad (4.4)$$

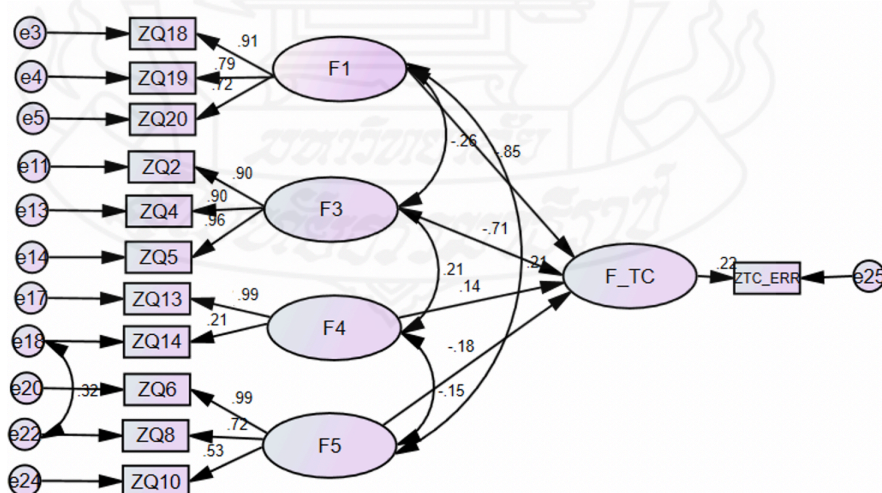
$$F4 = 0.90 * ZQ13 + 0.30 * ZQ14 \quad (4.5)$$

$$F5 = 0.97 * ZQ6 + 0.73 * ZQ8 + 0.54 * ZQ10 \quad (4.6)$$

ตารางที่ 4.7 ค่าสถิติที่ตรวจสอบโมเดลของค่าสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบไฮล์จากขนาดซอฟต์แวร์ (ZTC_EFF)

ค่าสถิติ	เกณฑ์	ค่าสถิติที่ได้จากการวิเคราะห์
CMIN-p	$P > 0.05$	0.216
CMIN/df	$CMIN/df < 3$	1.157
GFI	$GFI > 0.90$	0.913
RMSEA	$RMSEA < 0.08$	0.043

4.2.2 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบไฮล์จากข้อผิดพลาดของการทดสอบ



ภาพที่ 4.2 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กสำหรับการพัฒนาระบบแบบไฮล์จากข้อผิดพลาดของการทดสอบ

จากแบบจำลองสมการโครงสร้างที่พัฒนาขึ้นจากการประเมินแบบสอบถาม พบว่ามี 4 ปัจจัยที่มีผลต่อประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์เช่นเดียวกับ การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบออนไลน์จากขนาดซอฟต์แวร์ โดยสามารถเขียนสมการได้ ดังนี้

$$ZTC_ERR = 0.22 * F_TC \quad (4.7)$$

$$F_TC = F1+(-0.85)*F3+(-0.71)*F4+0.14*F5+(-0.18) \quad (4.8)$$

$$F1 = 0.91*ZQ18+0.79*ZQ19+0.72*ZQ20 \quad (4.9)$$

$$F3 = 0.90*ZQ2+0.90*ZQ4+0.96*ZQ5 \quad (4.10)$$

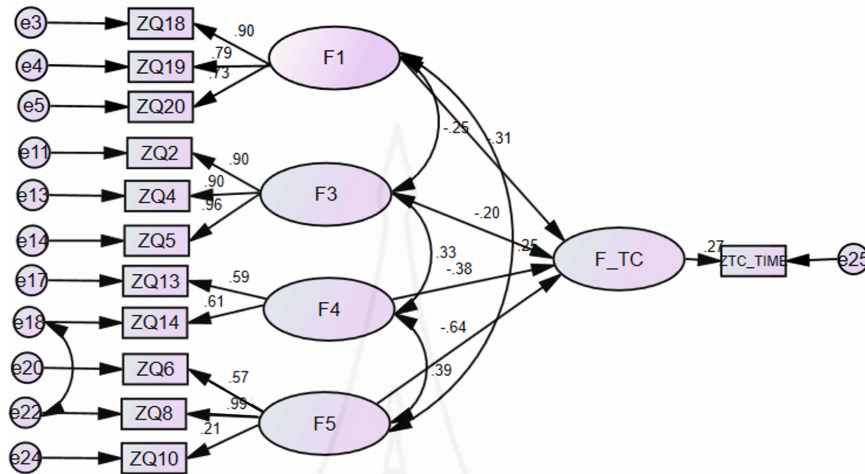
$$F4 = 0.99*ZQ13+0.21*ZQ14 \quad (4.11)$$

$$F5 = 0.99*ZQ6+0.72*ZQ8+0.53*ZQ10 \quad (4.12)$$

ตารางที่ 4.8 ค่าสถิติที่ตรวจสอบโมเดลของค่าสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบออนไลน์จากข้อผิดพลาดของการทดสอบ (ZTC_ERR)

ค่าสถิติ	เกณฑ์	ค่าสถิติที่ได้จากการวิเคราะห์
CMIN-p	P > 0.05	0.273
CMIN/df	CMIN/df < 3	1.116
GFI	GFI > 0.90	0.915
RMSEA	RMSEA < 0.08	0.036

4.2.3 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วย โรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอใจล์จากเวลาของการทดสอบ



ภาพที่ 4.3 แบบจำลองสมการ โครงสร้างประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรม
เวิร์คสำหรับการพัฒนาระบบแบบบอใจล์จากเวลาของการทดสอบ

จากแบบจำลองสมการ โครงสร้างที่พัฒนาขึ้นจากการประเมินแบบสอบถาม พบว่ามี 4 ปัจจัยที่มีผลต่อประสิทธิภาพการทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบบอใจล์เช่นเดียวกับ การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบบอใจล์จากขนาดซอฟต์แวร์และ โดยสามารถเขียนสมการได้ ดังนี้

$$ZTC_TIME = 0.27 * F_TC \quad (4.7)$$

$$F_TC = F1 + (-0.31)*F3 + (-0.20)*F4 + (-0.38)*F5 + (-0.64) \quad (4.8)$$

$$F1 = 0.90*ZQ18 + 0.79*ZQ19 + 0.73*ZQ20 \quad (4.9)$$

$$F3 = 0.90*ZQ2 + 0.90*ZQ4 + 0.96*ZQ5 \quad (4.10)$$

$$F4 = 0.59*ZQ13 + 0.61*ZQ14 \quad (4.11)$$

$$F5 = 0.57*ZQ6 + 0.99*ZQ8 + 0.21*ZQ10 \quad (4.12)$$

ตารางที่ 4.9 ค่าสถิติที่ตรวจสอบโมเดลของค่าสมการ โครงสร้างประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอใจล์จากเวลาของการทดสอบ (TIME_ERR)

ค่าสถิติ	เกณฑ์	ค่าสถิติที่ได้จากการวิเคราะห์
CMIN-p	P > 0.05	0.07

ค่าสถิติ	เกณฑ์	ค่าสถิติที่ได้จากการวิเคราะห์
CMIN/df	CMIN/df < 3	1.581
GFI	GFI > 0.90	0.910
RMSEA	RMSEA < 0.08	0.072

เมื่อ

ZQ2 คือ การทดสอบแบบอัตโนมัติมีความน่าเชื่อถือในการนำไปใช้กับภาพแบบการทำงานแบบออนไลน์

ZQ4 คือ การทดสอบแบบอัตโนมัติมีประสิทธิภาพ – ความเร็ว เมื่อนำมาใช้กับภาพแบบการทำงานแบบออนไลน์

ZQ5 คือ การทดสอบแบบอัตโนมัติมีความถูกต้อง – แม่นยำ ของผลการทดสอบเมื่อนำมาใช้กับภาพแบบการทำงานแบบออนไลน์

ZQ6 คือ ความสามารถของผู้จัดทำเอกสารความต้องการของระบบของผู้ใช้งาน มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ8 คือ เอกสารความต้องการของระบบของผู้ใช้งาน ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ10 คือ เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ13 คือ การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ14 คือ การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ18 คือ การพัฒนาระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ19 คือ การพัฒนาระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

ZQ20 คือ การพัฒนาระบบ มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบออนไลน์

4.3 การประมาณค่าความแม่นยำ

หลังจากที่ได้ผลการของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบออนไลน์ ผู้วิจัยได้ทำการประมาณค่าตัวแปร จำนวนกรณีทดสอบของซอฟต์แวร์ (F_TC) ของกลุ่มตัวอย่างอีก 30 ตัวอย่าง ซึ่งนำค่าตัวแปรอิสระไปคำนวณหา

ค่าของตัวแปร ZTC_EFF, ZTC_ERR และ ZTC_TIME ทำการแปลงค่า Z ให้อยู่ในรูปแบบ x (ค่าข้อมูลใด ๆ)

ตารางที่ 4.10 ข้อมูลที่นำมาทดสอบ

ตัวแปร	จำนวน	ค่าเฉลี่ย	ส่วนเบี่ยงเบนมาตรฐาน
ZTC_EFF	30	29.83	0.63
ZTC_ERR	30	9.16	0.05
ZTC_TIME	30	4.97	0.73

ค่าที่ได้จะเป็นค่าประมาณของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบไฮไลต์ แล้วนำมาเปรียบเทียบกับค่าประมาณของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบไฮไลต์จริงที่ได้จากแบบสอบถามเพื่อคำนวณหาค่า MRE (Average absolute MRE 100%) ได้เป็นค่าร้อยละความคลาดเคลื่อนของการประมาณการแบบจำลอง ซึ่งได้แสดงผลตามตารางที่ 4.11, 4.12 และ 4.13

ตารางที่ 4.11 ผลการคำนวณค่า MRE และ MMRE ของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์กภายใต้รูปแบบการทำงานแบบไฮไลต์จากขนาดซอฟต์แวร์

Number	Actual Efficiency	Predicted Efficiency	MRE
1	30	48.73	0.384
2	30	48.73	0.384
3	30	48.73	0.384
4	35	51.88	0.325
5	35	51.88	0.325
6	25	45.58	0.452
7	30	48.73	0.384
8	35	51.88	0.325
9	30	48.73	0.384
10	30	48.73	0.384
11	25	45.58	0.452
12	25	45.58	0.452
13	25	45.58	0.452
14	30	48.73	0.384
15	30	48.73	0.384
16	30	48.73	0.384

Number	Actual Efficiency	Predicted Efficiency	MRE
17	35	51.88	0.325
18	30	48.73	0.384
19	35	51.88	0.325
20	30	48.73	0.384
21	25	45.58	0.452
22	25	45.58	0.452
23	25	45.58	0.452
24	30	48.73	0.384
25	30	48.73	0.384
26	30	48.73	0.384
27	30	48.73	0.384
28	35	51.88	0.325
29	30	48.73	0.384
30	30	48.73	0.384
		MMRE	11.647

ผลการทดสอบของค์ประกอบการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์ ด้วยเทคนิคสมการเชิงโครงสร้าง พบว่า ประสิทธิภาพของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์จากขนาดซอฟต์แวร์ มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 11.647 %

ตารางที่ 4.12 ผลการคำนวณค่า MRE และ MMRE ของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์จากข้อผิดพลาดของการทดสอบ

Number	Actual Efficiency	Predicted Efficiency	MRE
1	5	9.41	0.47
2	10	9.66	0.04
3	5	9.41	0.47
4	10	9.66	0.04
5	5	9.41	0.47
6	8	9.56	0.16
7	20	10.16	0.97
8	5	9.41	0.47
9	12	9.76	0.23
10	7	9.51	0.26
11	9	9.61	0.06

Number	Actual Efficiency	Predicted Efficiency	MRE
12	2	9.26	0.78
13	25	10.41	1.40
14	10	9.66	0.04
15	5	9.41	0.47
16	5	9.41	0.47
17	10	9.66	0.04
18	5	9.41	0.47
19	15	9.91	0.51
20	15	9.91	0.51
21	20	10.16	0.97
22	10	9.66	0.04
23	5	9.41	0.47
24	3	9.31	0.68
25	8	9.56	0.16
26	6	9.46	0.37
27	10	9.66	0.04
28	15	9.91	0.51
29	12	9.76	0.23
30	13	9.81	0.33
		MMRE	12.11

ผลการทดสอบของคํ่าประกอบการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบขอใจล์ ด้วยเทคนิคสมการเชิงโครงสร้าง พบว่า ประสิทธิภาพของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบขอใจล์จากข้อผิดพลาดของการทดสอบ มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 12.11 %

ตารางที่ 4.13 ผลการคํ่านวนค่า MRE และ MMRE ของประสิทธิภาพการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบขอใจล์จากเวลาของการทดสอบ

Number	Actual Efficiency	Predicted Efficiency	MRE
1	6	9.35	0.358
2	2	6.43	0.689
3	8	10.81	0.260
4	8	10.81	0.260
5	6	9.35	0.358
6	1	5.7	0.825
7	4	7.89	0.493
8	1	5.7	0.825

Number	Actual	Efficiency	Predicted	Efficiency	MRE
9	5		8.62		0.420
10	6		9.35		0.358
11	3		7.16		0.581
12	1		5.7		0.825
13	4		7.89		0.493
14	6		9.35		0.358
15	7		10.08		0.306
16	4		7.89		0.493
17	1		5.7		0.825
18	8		10.81		0.260
19	5		8.62		0.420
20	6		9.35		0.358
21	6		9.35		0.358
22	2		6.43		0.689
23	5		8.62		0.420
24	3		7.16		0.581
25	3		7.16		0.581
26	4		7.89		0.493
27	5		8.62		0.420
28	3		7.16		0.581
29	3		7.16		0.581
30	8		10.81		0.260
			MMRE		14.73

ผลการทดสอบองค์ประกอบการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค ภายใต้รูปแบบการทำงานแบบไฮส ด้วยเทคนิคสมการเชิงโครงสร้าง พบว่า ประสิทธิภาพของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮสจาก เวลาของการทดสอบ มีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 14.73 %

บทที่ 5

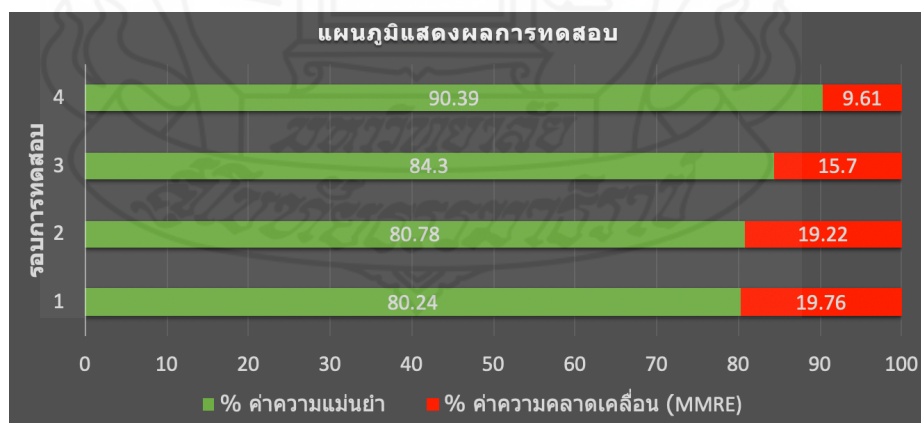
สรุปผล อภิปรายผล และข้อเสนอแนะ

การสรุปผล อภิปรายผล และข้อเสนอแนะ การวิเคราะห์ห้อยค์ประกอบ การทดสอบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คสำหรับการพัฒนาระบบแบบอใจล์ ผู้วิจัยได้สรุปตามขั้นตอน ดังนี้

- 5.1 สรุปผลการวิจัย
- 5.2 อภิปรายผล
- 5.3 ข้อเสนอแนะ

5.1 สรุปผลการวิจัย

5.1.1 ผู้วิจัยได้วิเคราะห์ห้อยค์ประกอบของการทดสอบระบบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอใจล์ โดยการเปรียบเทียบกับ การทดสอบระบบแบบมือภายใต้รูปแบบการทำงานแบบอใจล์ โดยทำการหาค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) โดยแบ่งตามกลุ่มการทดสอบ ดังนี้ การทดสอบซอฟต์แวร์เท่ากับ 1 ครั้ง ต่อ 1 โปรเจก ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 19.76 % การทดสอบซอฟต์แวร์เท่ากับ 2 ครั้ง ต่อ 1 โปรเจก ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 19.22 % การทดสอบซอฟต์แวร์เท่ากับ 3 ครั้ง ต่อ 1 โปรเจก ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 15.70 % และ การทดสอบซอฟต์แวร์เท่ากับ 4 ครั้ง ต่อ 1 โปรเจก ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 9.61 % ดังภาพที่ 5.1



ภาพที่ 5.1 แผนภูมิแสดงผลการทดสอบระบบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอใจล์

5.1.2 ผู้วิจัยได้วิเคราะห์องค์ประกอบของการทดสอบระบบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์จากขนาดซอฟต์แวร์ โดยใช้วิธีทางสถิติ คือ เทคนิคสมการเชิงโครงสร้าง จนได้แบบจำลองที่เหมาะสม และผ่านเกณฑ์พิจารณาความเหมาะสมของแบบจำลอง โดยได้ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 11.647 %

5.1.3 ผู้วิจัยได้วิเคราะห์องค์ประกอบของการทดสอบระบบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์จากข้อผิดพลาดของการทดสอบ โดยใช้วิธีทางสถิติ คือ เทคนิคสมการเชิงโครงสร้าง จนได้แบบจำลองที่เหมาะสม และผ่านเกณฑ์พิจารณาความเหมาะสมของแบบจำลอง โดยได้ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 12.11 %

5.1.4 ผู้วิจัยได้วิเคราะห์องค์ประกอบของการทดสอบระบบแบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์จากเวลาของการทดสอบ โดยใช้วิธีทางสถิติ คือ เทคนิคสมการเชิงโครงสร้าง จนได้แบบจำลองที่เหมาะสม และผ่านเกณฑ์พิจารณาความเหมาะสมของแบบจำลอง โดยได้ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) เท่ากับ 14.73 %



ภาพที่ 5.3 แผนภูมิแสดงผลค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MMRE) ของแบบจำลองสมการโครงสร้าง

5.2 อภิปรายผล

5.2.1 จากการวิจัยพบว่า การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์ มีประสิทธิภาพมากกว่าการทดสอบซอฟต์แวร์แบบมือภายใต้รูปแบบการทำงานแบบไฮไลต์ ในด้าน เวลาที่ใช้ในการทดสอบ ผลสำเร็จในการทดสอบ และลดความผิดพลาดจากการทดสอบ ในปริมาณที่กรณีทดสอบมีจำนวนที่เท่ากัน และ การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบไฮไลต์ เหมาะสมที่จะนำไปทำการทดสอบแบบถดถอย (Regression Testing) โดยสอดคล้องกับ ฌ็องร์ด็องน์ หาญวรวงศ์ (2556) ได้ทำ

การวิจัยเรื่อง การออกแบบและพัฒนาระบบการทดสอบสำหรับทดสอบซอฟต์แวร์แบบอัตโนมัติโดยใช้โครงสร้างยูไอ โดยมีผลการวิจัยว่า กรณีสอบแบบอัตโนมัติโดยใช้โครงสร้างยูไอช่วยให้การพัฒนาการทดสอบสำหรับทดสอบซอฟต์แวร์แบบอัตโนมัติมีความสะดวกและรวดเร็ว และ เศรษฐพงษ์ อิ่มสุวรรณ (2560) ได้ทำการวิจัยเรื่อง การใช้ระบบอัตโนมัติทดสอบซอฟต์แวร์เว็บแอปพลิเคชัน โดยมีผลการวิจัยว่า การทดสอบแบบอัตโนมัติที่นำมาประยุกต์ใช้กับการทดสอบซอฟต์แวร์แอปพลิเคชันทำให้ลดระยะเวลาในการทดสอบซอฟต์แวร์แอปพลิเคชัน และสามารถช่วยลดแรงงาน ค่าใช้จ่าย ความผิดพลาดในการบันทึกข้อมูล มากกว่าการทดสอบซอฟต์แวร์แอปพลิเคชันด้วยมือ และ Stanislav Stresnjak และ Zeljko Hocenski (2554) ได้ทำการวิจัยเรื่อง Usage of Robot Framework in Automation of Functional Test Regression โดยมีผลการวิจัยพบว่า การทดสอบระบบแบบมือเป็นกระบวนการที่ใช้เวลานานในการทำการทดสอบแบบถดถอย (Regression Testing) การทดสอบระบบแบบอัตโนมัติจึงเป็นการบวนการทดสอบที่ตอบโต้มากกว่า โดยโรบอทเฟรมเวิร์ค (Robot Framework) เป็นเครื่องมือที่นิยมใช้ทำการทดสอบแบบอัตโนมัติ และสามารถทดสอบได้หลายสภาพแวดล้อมไม่ว่าจะเป็น การทดสอบหน้าประสานผู้ใช้งาน (User Interface) ส่วนต่อประสานแอปพลิเคชัน (API) โดยทำการทดสอบได้อย่างมีประสิทธิภาพ รวดเร็ว และง่ายต่อการทดสอบแบบถดถอยหรือทดสอบแบบรวม

5.1.2 จากการวิจัยพบว่าองค์ประกอบของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโใจล์ มี 4 ปัจจัย คือ F1, F3, F4 และ F5 ทั้งนี้ผู้วิจัยได้ตั้งชื่อปัจจัยใหม่ เพื่อให้ครอบคลุมและเข้าใจง่าย คือ

F1 คือ องค์ประกอบด้านการพัฒนาระบบ

F3 คือ องค์ประกอบด้านการทดสอบระบบแบบอัตโนมัติ

F4 คือ องค์ประกอบด้านการวิเคราะห์ระบบ

F5 คือ องค์ประกอบด้านการรับความต้องการของระบบ

จากองค์ประกอบของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโใจล์ พบว่า

องค์ประกอบด้านการพัฒนาระบบ (F1) = +.14 ได้แก่ 1) การพัฒนาระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโใจล์ 2) การพัฒนาระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโใจล์ 3) การพัฒนาระบบ มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโใจล์

องค์ประกอบด้านการทดสอบระบบแบบอัตโนมัติ (F3) = +.14 ได้แก่ 1) การทดสอบแบบอัตโนมัติมีความน่าเชื่อถือในการนำไปใช้กับภาพแบบการทำงานแบบอโใจล์ 2) การทดสอบแบบอัตโนมัติมีประสิทธิภาพ – ความเร็ว เมื่อนำมาใช้กับภาพแบบการทำงานแบบอโใจล์ 3) การทดสอบ

แบบอัตโนมัติมีความถูกต้อง – แม่นยำ ของผลการทดสอบ เมื่อนำมาใช้กับภาพแบบการทำงานแบบอโงไจล์

องค์ประกอบด้านการวิเคราะห์ระบบ (F4) = +.16 ได้แก่ 1) การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโงไจล์ 2) การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโงไจล์

องค์ประกอบด้านการรับความต้องการของระบบ (F5) = +.16 ได้แก่ 1) เอกสารความต้องการของระบบของผู้ใช้งาน ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโงไจล์ 2) เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอโงไจล์

องค์ประกอบของการทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์ค มีองค์ประกอบที่เกี่ยวข้องกับภาพแบบการทำงานภายใต้รูปแบบการทำงานแบบอโงไจล์ คือ 1) การรับความต้องการของระบบ ที่ต้องอาศัยเอกสารความต้องการของผู้ใช้ระบบที่ถูกต้อง – แม่นยำ เพื่อช่วยลดความซับซ้อนของการออกแบบกรณีทดสอบแบบอัตโนมัติ และช่วยให้ผู้ทดสอบระบบและผู้ใช้งานระบบทำงานร่วมกันอย่างมีประสิทธิภาพ โดยสอดคล้องกับผลการวิจัยของ Rijwan Khan , Akhilesh Kumar Srivastava and Dilkeswar Pandey (2559) ได้ทำการวิจัยเรื่อง Agile Approach for Software Testing Process โดยผลการวิจัยพบว่า ในการทำงานในภาพแบบของ Agile จำเป็นจะต้องมีนักทดสอบระบบเป็นผู้กำหนดกรณีทดสอบตามพฤติกรรมของลูกค้า โดยการทำงานภาพแบบ Agile ออกแบบขึ้นมาเพื่อให้กระบวนการทางธุรกิจและความต้องการทางเทคนิคเกิดการผสานร่วมกันในการนำไปใช้เพื่อให้ได้บรรลุวัตถุประสงค์ในการทำงาน ส่วนกระบวนการทดสอบระบบเป็นส่วนที่ช่วยในการตรวจสอบ ยืนยันผลกระทบ ข้อปัญหา รวมถึงข้อบกพร่องต่าง ๆ ดังนั้น Agile Testing เป็นกระบวนการทำงานที่ผสมผสานกันระหว่างกระบวนการทำงานแบบ Agile และกระบวนการทดสอบระบบ โดยทีมพัฒนาและลูกค้านมีส่วนร่วมในการทำงานด้วยกัน ทำให้ซอฟต์แวร์เกิดประสิทธิภาพสูงสุด 2) การวิเคราะห์ระบบ ที่ต้องอาศัยความถูกต้อง – แม่นยำ เพื่อลดความซับซ้อนของการทดสอบระบบ 3) การพัฒนาระบบ มีผลต่อความถูกต้องและระยะเวลาของการทดสอบ 4) การทดสอบระบบแบบอัตโนมัติ เมื่อนำมาใช้กับกระบวนการทำงานในภาพแบบของอโงไจล์แล้ว ทำให้มีประสิทธิภาพของการทดสอบ มีความถูกต้องแม่นยำ ช่วยลดเวลาการทดสอบ และทำให้ผลการทดสอบมีความน่าเชื่อถือ โดยสอดคล้องกับงานวิจัยของ Sandeep Sivanandan และ Yogeesh C. B (2557) ได้ทำการวิจัยเรื่อง Agile Development Cycle: Approach to Design an Effective Model Based Testing with Behaviour Driven Automation Framework โดยผลการวิจัยพบว่า การทดสอบระบบแบบอัตโนมัติมีผลต่อกระบวนการในกรณีทดสอบที่นำไปใช้กับกรอบการทำงานและพฤติกรรมการพัฒนาภาพแบบการทดสอบแบบอัตโนมัติโดยอาศัยภาพแบบการทำงานแบบ Agile เพื่อช่วยให้การทำงานในแต่ละขั้นตอนสามารถนำกลับมาทดสอบซ้ำได้อย่างมี

ประสิทธิภาพพอใจ เกิดความยืดหยุ่นในการใช้งาน สามารถบำรุงรักษาและดูแลได้ง่าย รวมถึง Robot Framework ยังเป็นเครื่องมือที่ใช้ในการสั่งงานแบบอัตโนมัติ และปรับปรุงขั้นตอนการพัฒนาให้ยืดหยุ่นตามบริบทของงานได้

5.3 ข้อเสนอแนะ

การวิจัยเรื่อง การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอทเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบพอใจ มีข้อเสนอแนะดังนี้

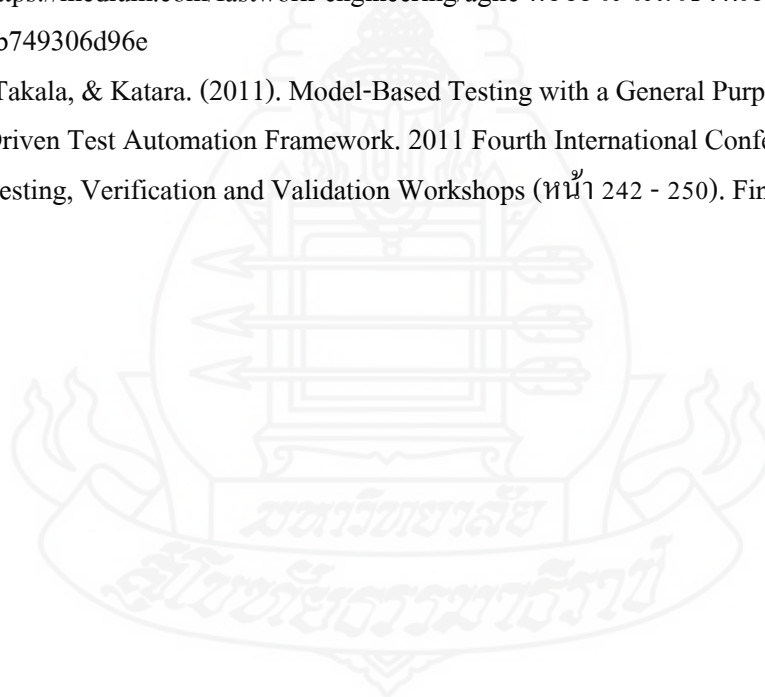
- 1) การทดสอบแบบอัตโนมัติโดยใช้โรบอทเฟรมเวิร์ค ควรนำไปพัฒนาต่อด้วยกระบวนการที่ใช้สำหรับการรวบรวมซอฟต์แวร์ที่มีการพัฒนาแยกส่วนกันอย่างอัตโนมัติ (Continuous Integration) โดยใช้ Jenkins เพื่อทำการตั้งเวลาในการทดสอบแบบถดถอย (Regression Testing) แบบอัตโนมัติ
- 2) การทดสอบแบบอัตโนมัตินั้นเหมาะสมกับการทดสอบแบบ Logic Test โดยมีข้อจำกัดในการทดสอบแบบ Semantic Test เพราะการทดสอบแบบอัตโนมัติอาจมีคุณภาพไม่เท่ากับการทดสอบโดยใช้คน
- 3) การวิจัยครั้งนี้เป็นการวัดผลเปรียบเทียบการทดสอบแบบอัตโนมัติและการทดสอบแบบมือ โดยเปรียบเทียบเป็นรอบการทดสอบเพื่อเก็บผลเรื่องเวลา และข้อผิดพลาด ของแต่ละรูปแบบการทดสอบ จึงไม่ได้ทำการทดสอบซ้ำให้ผลการทดสอบของแต่ละรอบผ่านเท่ากับ 100 เปอร์เซ็นต์
- 4) ควรเพิ่มขนาดตัวแปรสำหรับการทำแบบจำลองสมการ โครงสร้างโดยใช้วิธีการ Bootstrap เพื่อเพิ่มประสิทธิภาพของแบบจำลอง
- 5) ในการพัฒนาระบบแบบพอใจควรระบุตัวชี้วัดถึงความสัมพันธ์ระหว่างทีมพัฒนาและผู้ใช้งาน (User)

บรรณานุกรม

- ณัฐชากานต์ เดวิส. (2562). การนำแนวคิด Agile มาใช้ในการบริหารโครงการ. นครปฐม: Nidtep e-Journal.
- ณัฐรัตน์ หารวรงค์. (2556). การออกแบบและการพัฒนาการสร้างกรณีทดสอบซอฟต์แวร์แบบอัตโนมัติโดยใช้โครงสร้างยูไอ. กรุงเทพมหานคร: คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.
- นันท์นันทน์ ช่วยชู. (2558). การสร้างกรณีทดสอบโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม. สงขลา: สาขาวิชาการจัดการเทคโนโลยีสารสนเทศ มหาวิทยาลัยสงขลานครินทร์.
- สกรณีย์ บุญบง. (2556). การสร้างกรณีทดสอบสำหรับการทดสอบระดับรวมหน่วยแบบเพิ่มทีละหน่วย โดยอัตโนมัติจากกรณีทดสอบระดับหน่วย. นครราชสีมา: สาขาวิชา วิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.
- สนธิพรรณ จิตตั้งสมบูรณ์ . (2559). ปัจจัยที่ส่งผลต่อความตั้งใจในการใช้งานแนวคิด การพัฒนาซอฟต์แวร์รูปแบบDevOps. กรุงเทพมหานคร: คณะพาณิชยศาสตร์และการบัญชี มหาวิทยาลัยธรรมศาสตร์.
- เศรษฐพงษ์ อิ่มสุวรรณ. (2561). การใช้ระบบอัตโนมัติทดสอบเว็บแอปพลิเคชัน : กรณีศึกษาบริษัทด้านเทคโนโลยีสารสนเทศและการสื่อสาร. กรุงเทพมหานคร: วิทยาลัยนวัตกรรมการศึกษาและเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์.
- อรรถนรินทร์ สานุกตร์. (2561). Web Automated Test ด้วย Robot Framework เบื้องต้น. เข้าถึงได้จาก Sanook: <https://learning.sanook.com/60738/>
- BENJAWAN T. (2019). Launch Chrome on Android from Java Automation Test. เข้าถึงได้จาก Medium: <https://medium.com/@benba>
- Evvari, K. (2012). Software Test Management Tool Evaluation Framework. Tartu, Estonia: FACULTY OF MATHEMATICS AND COMPUTER SCIENCE INSTITUTE OF COMPUTER SCIENCE, UNIVERSITY OF TARTU .
- Harsha & Kumari. (2017). Software Test Automation with Robot Framework. International Journal on Recent and Innovation Trends in Computing and Communication (หน้า 432 – 433). Mysuru, India: Research Department, Auricle Technologies Pvt. Ltd.,.
- Na & Huaichang. (2015). Extension And Application Based on Robot Testing Framework. 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS). Beijing, China: IEEE.
- Robot Framework. (2021). Robot Framework. เข้าถึงได้จาก Robot Framework: <https://robotframework.org/>

บรรณานุกรม (ต่อ)

- Sivanandan & Yogeesh . (2014). Agile Development Cycle: Approach to Design an Effective Model Based Testing with Behaviour Driven Automation Framework. 2014 20th Annual International Conference on Advanced Computing and Communications (ADCOM). Karnataka, INDIA: Research Gate.
- Gojarea, Joshib & Gaigawarec. (2015). Analysis and Design of Selenium WebDriver Automation Testing, Procedia Computer Science 50 : 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15) (หน้า 341 – 346). India: Elsevier B.V.
- Stresnjak & Hocenski. (2011). Usage of Robot Framework in Automation of Functional Test Regression. ICSEA 2011 : The Sixth International Conference on Software Engineering Advances. Osijek, Croatia: Research Gate.
- Akarasomcheep, T. (2561). Agile คืออะไร เริ่มใช้งานอย่างไร. เข้าถึงได้จาก Medium: <https://medium.com/fastwork-engineering/agile-คืออะไร-เริ่มใช้งานอย่างไร-ab749306d96e>
- Pajunen, Takala, & Katara. (2011). Model-Based Testing with a General Purpose Keyword-Driven Test Automation Framework. 2011 Fourth International Conference on Software Testing, Verification and Validation Workshops (หน้า 242 - 250). Finland: IEEE.



ภาคผนวก



ภาคผนวก ก

แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงาน
แบบออนไลน์โดยผู้เชี่ยวชาญ (IOC)





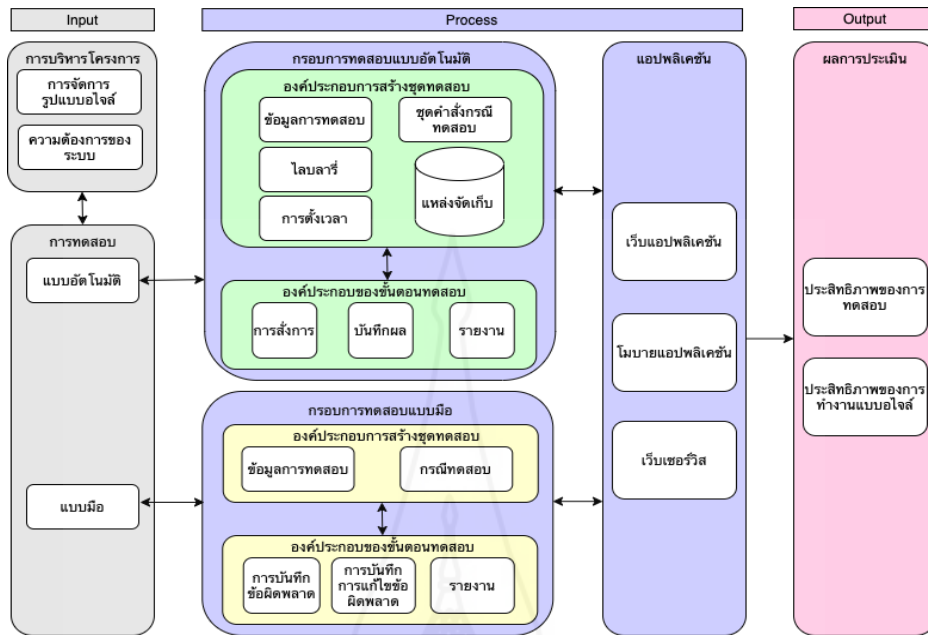
**แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ
ภายใต้รูปแบบการทำงานแบบออนไลน์**

คำชี้แจง

1. แบบสอบถามฉบับนี้มีทั้งหมด 16 หน้า มีความประสงค์ที่จะรวบรวมข้อมูลความคิดเห็นของผู้เชี่ยวชาญในการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ เพื่อนำข้อมูลไปวิเคราะห์ประเมินผล การทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ ซึ่งเป็นส่วนหนึ่งของวิทยานิพนธ์ มหาวิทยาลัยสุโขทัยธรรมมาธิราช โดยมีอาจารย์ ดร.ศรันย์ นาคถนอม เป็นที่ปรึกษา และ รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์ เป็นที่ปรึกษาร่วม
2. แบบสอบถามฉบับนี้เป็นการสอบถามความคิดเห็นประเภท Rating Scale เกี่ยวกับประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์
3. แบบสอบถามฉบับนี้เป็นการสอบถามความคิดเห็นเกี่ยวกับประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ โดยแบบสอบถามได้กำหนดองค์ประกอบ เพื่อเป็นกรอบให้ผู้เชี่ยวชาญ ท่านสามารถเพิ่มหรือตัดองค์ประกอบใด ๆ ได้ โดยท่านสามารถเสนอแนะความคิดเห็นในแบบประเมินฉบับนี้ ความคิดเห็นของท่านจะเป็นประโยชน์อย่างยิ่งในการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ ให้มีความสมบูรณ์ยิ่งขึ้น

ขอกราบขอบพระคุณในความกรุณาของท่านที่ได้แสดงความคิดเห็นที่เป็นประโยชน์ต่อการวิจัยครั้งนี้

นาย อภิสิตธิ์ เชนนาม
ผู้วิจัย



รูปที่ 1 แบบจำลองการทดสอบแบบภายใต้รูปแบบการทำงานแบบอจีล์

Agile คือ กรอบแนวคิด และวิธีการทำงานอย่างคล่องแคล่วรวดเร็ว เพื่อปรับเปลี่ยนให้สามารถตอบสนองความต้องการของลูกค้าได้อย่างรวดเร็วในสภาวะที่มีความไม่แน่นอน โดยมีวงจรชีวิตการทำงานแบบอจีล์ หรือ Agile SDLC Model ดังนี้

1. **ปัญหาหรือความต้องการของระบบ (Requirement)** คือ ปัญหาหรือความต้องการของระบบ (Requirement) ได้มาจากการที่ผู้ใช้งาน (User) นั้นมีความต้องการหรือพบปัญหาเกี่ยวกับระบบที่ใช้งานอยู่และต้องการให้มีการแก้ไขหรือเพิ่มฟีเจอร์ (Feature) ให้กับระบบ ผู้ใช้งานจะให้เอกสารปัญหาหรือความต้องการ (Requirement) มาในรูปแบบ User Story ซึ่งทีมงานอจีล์ (Agile Methodology) จะต้องรับปัญหาหรือความต้องการ (Requirement) ไปสร้างเป็นงานตามลักษณะงานของตัวเอง

2. **ออกแบบระบบ (Design)** คือ ผู้วิเคราะห์และออกแบบระบบ (System Analyst) เป็นผู้รับ ปัญหาหรือความต้องการ (Requirement) จากผู้ใช้งาน (User) และมาทำการวิเคราะห์และออกแบบแนวทางการพัฒนาซอฟต์แวร์เพื่อให้ทีมพัฒนาระบบ (Developer) นำไปพัฒนาให้ตรงกับความต้องการของผู้ใช้งาน (User)

3. **พัฒนาระบบ (Development)** คือ ผู้พัฒนาระบบ (Developer / Programmer) เป็นผู้ที่ทำหน้าที่พัฒนาซอฟต์แวร์ตามที่ผู้วิเคราะห์และออกแบบระบบ (System Analyst) ได้ออกแบบไว้โดยสอดคล้องกับปัญหาหรือความต้องการของระบบ (Requirement) ของผู้ใช้งาน

(User) ให้สำเร็จถึงจะนำซอฟต์แวร์ที่เสร็จสมบูรณ์ส่งให้ทีมทดสอบระบบ (Software Tester / QA) ทำการตรวจสอบ

4. ทดสอบระบบ (Software Testing) เป็นการตรวจสอบความถูกต้องของซอฟต์แวร์หลังจากที่ผู้พัฒนาระบบ (Developer / Programmer) ทำการพัฒนาเสร็จสมบูรณ์ ซึ่งการทดสอบระบบนั้นต้องใช้เวลาและความละเอียดรอบคอบ ซึ่งการทำงานแบบอไจล์ (Agile Methodology) เป็นการทำงานที่ต้องอาศัยความรวดเร็วและถูกต้องแม่นยำ ผู้จัดทำวิจัยจึงได้นำการทดสอบแบบอัตโนมัติ (Automation Testing) โดยใช้โรบอตเฟรมเวิร์ค (Robot Framework) มาแทนการทดสอบระบบแบบมือ (Manual Testing) เพื่อเป็นการลดเวลาและความผิดพลาดที่จะเกิดขึ้นจากมนุษย์ (Human Error)

4.1 วิธีการทดสอบระบบแบบมือ (Manual Testing)

คือการจำลองหรือออกแบบกรณีทดสอบ เพื่อยืนยันความถูกต้องของระบบ ผู้ทดสอบซอฟต์แวร์ (Software Tester) ทำการทดสอบซอฟต์แวร์/เปรียบเทียบผลการทดสอบ (Test Execution) โดยผู้ทดสอบจะทำการทดสอบตามกรณีทดสอบ (Test Cases) โดยผู้ทดสอบเอง โดยอ้างอิงความถูกต้องของการทำงานของระบบจากเอกสารความต้องการของลูกค้า (User Requirement) โดยแบ่งขั้นตอนการทดสอบระบบแบบมือ (Manual Testing) ออกเป็น 6 ขั้นตอน ดังนี้

1 การวิเคราะห์ความต้องการของผู้ใช้ระบบ (Requirement Analysis) การวิเคราะห์ความต้องการของผู้ใช้ระบบ เป็นขั้นตอนการสำคัญของการทดสอบระบบโดยเป็นกระบวนการตรวจสอบข้อมูลขั้นพื้นฐานในการทดสอบในแต่ละกรณี รวมไปถึงการออกแบบข้อกำหนดการทดสอบ และ ความเสี่ยงต่าง ๆ ที่จะมีโอกาสเกิดขึ้นในระบบ

2 การวางแผนการทดสอบระบบ (Test Plan Creation) การวางแผนการทดสอบระบบเป็นขั้นตอนการออกแบบแนวทางรวมถึงวัตถุประสงค์ของการทดสอบ โดยอาศัยการวิเคราะห์ความต้องการของผู้ใช้ระบบ ในการเขียนกรณีทดสอบรวมไปถึงการกำหนดสภาพแวดล้อมการทดสอบ และระบุเกณฑ์ความสำเร็จว่าต้องอยู่ที่ระดับใดในกรณีที่ทำทดสอบเสร็จสมบูรณ์

3 การออกแบบเอกสารกรณีทดสอบระบบ (Test Case Creation) เป็นการพัฒนาและจัดลำดับความสำคัญในแต่ละกรณีทดสอบ โดยใช้การสร้างข้อมูลการทดสอบในการทดสอบตามกรณีทดสอบต่าง ๆ

4 การทดสอบระบบตามเอกสารกรณีการทดสอบระบบ (Test Case Execution) คือการดำเนินการทดสอบระบบตามกรณีทดสอบที่พัฒนาไว้ โดยใช้ผู้ทดสอบ

ระบบเป็นผู้ที่ระบุและเก็บผลการทดสอบว่าแต่ละกรณีเป็นไปตามความต้องการของผู้ใช้ระบบหรือไม่

5 การรายงานข้อผิดพลาดของระบบ (Defect Logging) คือการบันทึกข้อผิดพลาดของระบบที่ผู้ทดสอบระบบพบข้อบกพร่องระหว่างการทดสอบ โดยการรายงานข้อผิดพลาดนี้จะต้องส่งไปให้ผู้ทดสอบระบบเป็นผู้แก้ไขให้เป็นไปตามความต้องการของผู้ใช้ระบบ

6 การทดสอบระบบซ้ำหลังจากเจอข้อผิดพลาดของระบบ (Re-Verification) คือการทดสอบกรณีทดสอบที่ผิดพลาดซ้ำหลังจากที่ผู้พัฒนาระบบได้ทำการแก้ไขข้อผิดพลาดจากการรายงานข้อผิดพลาด เพื่อพิสูจน์ว่ากรณีทดสอบนั้นถูกต้องตามความต้องการของผู้ใช้ระบบ

4.2 การทดสอบระบบแบบอัตโนมัติ (Automation Testing)

คือการสร้างชุดกรณีทดสอบโดยสั่งงานผ่านซอฟต์แวร์ โดยนิยมนำมาใช้ในการทดสอบเชิงถดถอย เพื่อลดเวลาและข้อผิดพลาดจากการทดสอบโดยคน ในการทดสอบในกรณีที่ระบบมีขนาดใหญ่ และสามารถนำชุดทดสอบมาใช้ซ้ำได้ตามต้องการเนื่องจากเป็นโปรแกรมที่ทำงานด้วยคอมพิวเตอร์ การทำงานแต่ละครั้งจึงไม่แตกต่างกัน โดยมีโครงสร้างที่ใช้ในการกำหนดแนวทางในการพัฒนาและบำรุงรักษาดังนี้

1. Project Management คือการทราบถึงรูปแบบการทำงานขององค์กร ว่าใช้รูปแบบการทำงานแบบไหนในการทดสอบระบบแบบอัตโนมัติ
2. Test Management เป็นกระบวนการสร้างเอกสารกรณีทดสอบต่าง ๆ ตามเอกสารความต้องการของลูกค้า เพื่อทดสอบว่าผลลัพธ์เป็นไปตามที่คาดหวัง
3. Test Generating Layer เป็นกระบวนการที่จะสร้าง Test Script ขึ้นมาเพื่อใช้ในการทดสอบ ซึ่งการสร้าง Test Script นั้นต้องประกอบไปด้วย ข้อมูลการทดสอบ (Test Data) เป็นข้อมูลที่ถูกสร้างขึ้นเพื่อเป็นข้อมูล Input สำหรับการทดสอบระบบ Test Script คือการพัฒนากรณีทดสอบให้อยู่ในรูปแบบของการ Coding เพื่อให้สามารถ Execute Test ได้ตามภาษาโปรแกรมที่ Framework นั้น ๆ รองรับ และมี Test Library เป็นตัวเพิ่มความสามารถของ Test Script ซึ่งการทำ Test Script ในแต่ละครั้งนั้นควรจะต้องมี Object Repository เพื่อใช้ในการเก็บ Version ของ Test Script เมื่อมี Test Script ที่ครอบคลุมทุกกรณีทดสอบแล้วสิ่งต่อมาคือการ ทดสอบแบบอัตโนมัติ (Test Execution Layer) ซึ่งการรันทดสอบแบบอัตโนมัตินั้นไม่ว่าจะรันทดสอบด้วย Environment ไหน หรือ ระบบไหน ต้องสามารถเก็บผลการทดสอบและสามารถออกรายงานผลการทดสอบได้

5. การขึ้นระบบ (Deployment) คือ ผู้พัฒนาระบบ (Developer / Programmer / Devops) ทำหน้าที่นำซอฟต์แวร์ที่ผ่านกระบวนการทดสอบเรียบร้อยแล้วขึ้นไปให้ผู้ใช้งาน (User) จริงได้ใช้งาน



แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์

คำชี้แจง

แบบสอบถามงานชุดนี้ มีวัตถุประสงค์เพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ โดยมีข้อคำถามแบ่งออกเป็น 5 หมวดดังต่อไปนี้

1. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Automation Testing with Agile Testing)
2. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการรับปัญหาหรือความต้องการของระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Requirement with Agile Testing)
3. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการออกแบบระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (System Analyze with Agile Testing)
4. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการพัฒนาระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)
5. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการขึ้นระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)

โดยส่วนประกอบของแบบสอบถาม ประกอบด้วย 3 ตอนดังนี้

ตอนที่ 1 ข้อมูลส่วนตัวของผู้ประเมิน

ตอนที่ 2 : การแสดงความคิดเห็นของผู้ตอบแบบสอบถามเกี่ยวกับการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์

ตอนที่ 3 : ข้อเสนอแนะ

ตอนที่ 1 ข้อมูลส่วนตัวของผู้ประเมิน

1. ตำแหน่งงาน
2. ประเภทของหน่วยงาน เอกชน รัฐบาล
3. ระดับการศึกษา

ตอนที่ 2 : การแสดงความคิดเห็นของผู้ตอบแบบสอบถามเกี่ยวกับการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ ที่พัฒนาขึ้น ประกอบด้วยส่วนของข้อคำถามที่อยู่ด้านซ้ายมือ และมาตรฐานประมาณค่าที่อยู่ด้านขวามือ

หมวดที่ 1 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Automation Testing with Agile Testing)

กรณารอกข้อมูลจำนวนกรณีทดสอบ (Test Cases) และระดับความซ้ำซ้อน ในกระบวนการทำงานแบบอไจล์ เพื่อประเมินผลของระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ โดยค่าระดับความซ้ำซ้อน มีดังนี้

- Low = ระดับความซ้ำซ้อนต่ำ
- Average = ระดับความซ้ำซ้อนปานกลาง
- High = ระดับความซ้ำซ้อนสูง

คำชี้แจง : โปรดใส่เครื่องหมาย ลงในช่อง และเพิ่มเติมข้อเสนอแนะลงในช่องว่าง

1.1 ขนาดของซอฟต์แวร์ (size, test cases)

ปัญหาหรือความต้องการของระบบ (Requirement)	จำนวนกรณีทดสอบ (Test Cases)	ระดับความซ้ำซ้อน (Redundancy Level)
1. จำนวนกรณีทดสอบของแต่ละหน่วยของระบบ	<input type="checkbox"/> Low <input type="checkbox"/> Average <input type="checkbox"/> High

ตัวอย่าง หากซอฟต์แวร์ มีจำนวนกรณีทดสอบ (Test Case) จำนวน 30 Test Cases ต่อ Function, Object or Application point และมีระดับความยาก / สลับซับซ้อนค่อนข้างสูง (ตามความคิดเห็นของผู้ทดสอบระบบ) ให้กรอข้อมูลดังนี้

ปัญหาหรือความต้องการของระบบ (Requirement)	จำนวนกรณีทดสอบ (Test Cases)	ระดับความซ้ำซ้อน (Redundancy Level)
1. จำนวนกรณีทดสอบของแต่ละหน่วยของระบบ30.....	<input type="checkbox"/> Low <input type="checkbox"/> Average <input checked="" type="checkbox"/> High

ตัวชี้วัด	เห็นด้วย	ไม่เห็นด้วย	ไม่แน่ใจ	ข้อเสนอแนะ
กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ				
1.2 การทดสอบแบบอัตโนมัติมีความน่าเชื่อถือในการนำไปใช้กับรูปแบบการทำงานแบบอไจล์				

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและ ลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ	เห็นด้วย	ไม่เห็นด้วย	ไม่แน่ใจ	ข้อเสนอแนะ
1.3 การทดสอบแบบอัตโนมัติช่วยลดความซับซ้อน ของระบบที่พัฒนาขึ้นภายใต้รูปแบบการทำงาน แบบอไจล์				
1.4 การทดสอบแบบอัตโนมัติมีประสิทธิภาพ – ความเร็ว เมื่อนำมาใช้กับรูปแบบการทำงานแบบอ ไจล์				
1.5 การทดสอบแบบอัตโนมัติมีความถูกต้อง – แม่นยำ ของผลการทดสอบ เมื่อนำมาใช้กับรูปแบบ การทำงานแบบอไจล์				

หมวดที่ 2 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการรับปัญหาหรือความต้องการของระบบกับการ
ทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Requirement with Agile
Testing)

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและ ลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ	เห็นด้วย	ไม่เห็นด้วย	ไม่แน่ใจ	ข้อเสนอแนะ
2.1 ความสามารถของผู้จัดทำเอกสารความต้องการ ของระบบของผู้ใช้งาน มีผลกับการทดสอบระบบ แบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
2.2 เอกสารความต้องการของระบบของผู้ใช้งาน มี ผลต่อความน่าเชื่อถือของการทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
2.3 เอกสารความต้องการของระบบของผู้ใช้งาน ช่วยลดความซับซ้อนในการทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
2.4 เอกสารความต้องการของระบบของผู้ใช้งาน มี ผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์				
2.5 เอกสารความต้องการของระบบของผู้ใช้งาน มี ผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบ แบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				

หมวดที่ 3 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการออกแบบระบบกับการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ (System Analyze with Agile Testing)

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ	เห็นด้วย	ไม่เห็นใจ	ไม่เห็นด้วย	ข้อเสนอแนะ
3.1 ความสามารถของผู้วิเคราะห์ระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
3.2 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อความน่าเชื่อถือของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
3.3 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
3.4 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
3.5 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				

หมวดที่ 4 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการพัฒนาระบบกับการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ	เห็นด้วย	ไม่เห็นใจ	ไม่เห็นด้วย	ข้อเสนอแนะ

4.1 ความสามารถของผู้พัฒนาระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
4.2 การพัฒนาระบบ มีผลต่อความน่าเชื่อถือของทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
4.3 การพัฒนาระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
4.4 การพัฒนาระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				
4.5 การพัฒนาระบบ มีผลต่อความถูกต้อง – แม่นยำของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์				

หมวดที่ 5 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการขึ้นระบบกับการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ (Deployment with Agile Testing)

ตัวชี้วัด	เห็นด้วย	ไม่เห็นใจ	ไม่เห็นด้วย	ข้อเสนอแนะ
กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ				
5.1 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ ส่งผลต่อความน่าเชื่อถือของระบบ				
5.2 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ ช่วยลดความเสี่ยงในการขึ้นระบบจริงไม่สำเร็จ				
5.3 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ มีผลต่อประสิทธิภาพ – ความเร็ว ของระบบในขั้นตอนการขึ้นระบบจริง				
5.4 การทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ มีผลต่อความถูกต้อง – แม่นยำของระบบในขั้นตอนการขึ้นระบบจริง				

ตัวชี้วัด กระบวนการทำงานแบบบอใจล์มีผลต่อระยะเวลาและ ลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ	เห็นด้วย	ไม่เห็นใจ	ไม่เห็นด้วย	ข้อเสนอแนะ
5.5 การขึ้นระบบจริง โดยผ่านการทดสอบแบบ อัตโนมัติภายใต้รูปแบบการทำงานแบบบอใจล์ มีผล ต่อความพึงพอใจ ในการใช้งานระบบของผู้ใช้				

ตอนที่ 3 : ข้อเสนอแนะ

.....

.....

.....

.....

.....

.....

.....



คำอธิบายตัวชี้วัด

หมวดที่ 1 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Automation Testing with Agile Testing)

ตัวชี้วัด	คำอธิบาย
1.1 ขนาดของซอฟต์แวร์	เป็นการพิจารณาขนาดของระบบที่จะทดสอบว่ามีขนาดเท่าไร โดยพิจารณาจาก กรณีทดสอบ (Test Case) ที่ครอบคลุมทุกหน่วยการทำงานย่อยระบบ และมีระดับความยาก / สลับซับซ้อนเท่าไร (ตามความคิดเห็นของผู้ทดสอบระบบ)
1.2 การทดสอบแบบอัตโนมัติมีความน่าเชื่อถือในการนำไปใช้กับรูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติโดยใช้กระบวนการทำงานแบบอไจล์ มีผลต่อระยะเวลาและลดข้อผิดพลาดในการทดสอบ และทำให้ระบบที่พัฒนาขึ้นมีความน่าเชื่อถือ
1.3 การทดสอบแบบอัตโนมัติช่วยลดความซับซ้อนของระบบที่พัฒนาขึ้นภายใต้รูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติ มีผลต่อระยะเวลาและลดข้อผิดพลาดในการทดสอบ และช่วยลดความซับซ้อน ในกระบวนการทำงานแบบอไจล์
1.4 การทดสอบแบบอัตโนมัติมีประสิทธิภาพ – ความเร็ว เมื่อนำมาใช้กับรูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติภายใต้กระบวนการทำงานแบบอไจล์ มีผลต่อระยะเวลา ลดข้อผิดพลาด และทำให้ระบบมีประสิทธิภาพ
1.5 การทดสอบแบบอัตโนมัติมีความถูกต้อง – แม่นยำ ของผลการทดสอบ เมื่อนำมาใช้กับรูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติภายใต้กระบวนการทำงานแบบอไจล์ มีผลต่อระยะเวลา ลดข้อผิดพลาด ของผลการทดสอบ

หมวดที่ 2 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการรับปัญหาหรือความต้องการของระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Requirement with Agile Testing)

ตัวชี้วัด	คำอธิบาย
2.1 ความสามารถของผู้จัดทำเอกสารความต้องการของระบบของผู้ใช้งาน มีผลกับการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์	ความสามารถของผู้จัดทำเอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ที่นำไปใช้ในการทดสอบแบบอัตโนมัติ
2.2 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความน่าเชื่อถือของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งส่งผลต่อความน่าเชื่อถือของการทดสอบระบบทดสอบแบบอัตโนมัติ
2.3 เอกสารความต้องการของระบบของผู้ใช้งาน ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งช่วยลดความซับซ้อนของการทดสอบระบบทดสอบแบบอัตโนมัติ
2.4 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งมีผลต่อระยะเวลาในการทดสอบระบบทดสอบแบบอัตโนมัติ
2.5 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความถูกต้อง – แม่นยำของการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งมีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบทดสอบแบบอัตโนมัติ

หมวดที่ 3 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการออกแบบระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (System Analyze with Agile Testing)

ตัวชี้วัด	คำอธิบาย
3.1 ความสามารถของผู้วิเคราะห์ระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์	ผู้วิเคราะห์ระบบที่สามารถออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ต่อการออกแบบกรณีทดสอบได้ครอบคลุม
3.2 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อ	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลต่อการออกแบบกรณีทดสอบได้คล

ตัวชี้วัด	คำอธิบาย
ความน่าเชื่อถือของทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ครอบคลุม ซึ่งทำให้ทดสอบระบบได้อย่างครบถ้วน และทำให้ระบบมีความน่าเชื่อถือ
3.3 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ช่วยลดความซ้ำซ้อนในการวางแผนและออกแบบกรณีทดสอบ
3.4 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ต่อการออกแบบกรณีทดสอบได้อย่างครอบคลุม ซึ่งทำให้ทดสอบระบบได้อย่างครบถ้วน และช่วยลดระยะเวลาในการทดสอบ
3.5 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ต่อการออกแบบกรณีทดสอบได้อย่างครอบคลุม ซึ่งทำให้ทดสอบระบบได้อย่างครบถ้วน และช่วยลดระยะเวลาในการทดสอบ

หมวดที่ 4 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการพัฒนาระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)

ตัวชี้วัด	คำอธิบาย
4.1 ความสามารถของผู้พัฒนาระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ความสามารถของผู้พัฒนาระบบ ที่สามารถพัฒนาระบบได้เป็นไปตามเอกสารความต้องการของลูกค้า และตรงตามการวิเคราะห์และออกแบบระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์
4.2 การพัฒนาระบบ มีผลต่อความน่าเชื่อถือของทดสอบระบบแบบ	การพัฒนาระบบได้ตรงตามเอกสารความต้องการของลูกค้า และตรงตามการวิเคราะห์และออกแบบ

ตัวชี้วัด	คำอธิบาย
อัตโนมัติภายใต้รูปแบบการทำงานแบบบอ ไจล์	ระบบ ส่งผลให้การทดสอบได้ครอบคลุม ซึ่งทำให้ ทดสอบระบบได้อย่างครบถ้วน และทำให้ระบบมี ความน่าเชื่อถือ
4.3 การพัฒนาระบบ ช่วยลดความ ซับซ้อนในการทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการทำงานแบบบอ ไจล์	การพัฒนาระบบได้ตรงตามเอกสารความต้องการ ของลูกค้า และตรงตามการวิเคราะห์และออกแบบ ระบบ ช่วยลดความซับซ้อนในการทดสอบ
4.4 การพัฒนาระบบ มีผลต่อระยะเวลา ในการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบบอไจล์	การพัฒนาระบบได้ตรงตามเอกสารความต้องการ ของลูกค้า และตรงตามการวิเคราะห์และออกแบบ ระบบ ส่งผลให้ระยะเวลาการทดสอบระบบแบบ อัตโนมัติลดลง หรือทำการทดสอบสำเร็จได้ตาม เป้าหมาย
4.5 การพัฒนาระบบ มีผลต่อความ ถูกต้อง – แม่นยำ ของการทดสอบระบบ แบบอัตโนมัติภายใต้รูปแบบการทำงาน แบบบอไจล์	การพัฒนาระบบได้ตรงตามเอกสารความต้องการ ของลูกค้า และตรงตามการวิเคราะห์และออกแบบ ระบบ ส่งผลให้การทดสอบระบบแบบอัตโนมัติมี ความถูกต้อง แม่นยำ และลดการเกิดข้อผิดพลาดที่ จะเกิดขึ้นได้ของระบบ

หมวดที่ 5 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการขึ้นระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบ
การทำงานแบบบอไจล์ (Deployment with Agile Testing)

ตัวชี้วัด	คำอธิบาย
5.1 การขึ้นระบบจริง โดยผ่านการ ทดสอบแบบอัตโนมัติภายใต้รูปแบบการ ทำงานแบบบอไจล์ ส่งผลต่อความ น่าเชื่อถือของระบบ	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตาม เอกสารความต้องการของลูกค้า สามารถสร้างความ เชื่อถือให้กับผู้ใช้งานจริงได้ ในกระบวนการพัฒนา ซอฟต์แวร์แบบบอไจล์ หลังจากการขึ้นระบบใน สภาพแวดล้อมจริง
5.2 การขึ้นระบบจริง โดยผ่านการ ทดสอบแบบอัตโนมัติภายใต้รูปแบบการ ทำงานแบบบอไจล์ ช่วยลดความเสี่ยงใน	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตาม เอกสารความต้องการของลูกค้า สามารถช่วยลด ความเสี่ยงให้การยกเลิก หรือ การขึ้นระบบบน

การขึ้นระบบในสภาพแวดล้อมจริงไม่สำเร็จ	สภาพแวดล้อมจริงไม่สำเร็จ เนื่องด้วยสาเหตุของระบบไม่สมบูรณ์
5.3 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้ล มีผลต่อประสิทธิภาพ – ความเร็ว ของระบบในขั้นตอนการขึ้นระบบจริง	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตามเอกสารความต้องการของลูกค้า สามารถทำให้การขึ้นระบบบนสภาพแวดล้อมจริงได้อย่างรวดเร็วและมีประสิทธิภาพ
5.4 การทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้ล มีผลต่อความถูกต้อง – แม่นยำ ของระบบในขั้นตอนการขึ้นระบบจริง	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตามเอกสารความต้องการของลูกค้า สามารถทำให้การขึ้นระบบบนสภาพแวดล้อมจริงได้อย่างถูกต้องและแม่นยำ
5.5 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้ล มีผลต่อความพึงพอใจในการใช้งานระบบของผู้ใช้	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตามเอกสารความต้องการของลูกค้า สามารถสร้างความพึงพอใจให้กับผู้ใช้งานจริงได้ ในกระบวนการพัฒนาซอฟต์แวร์แบบบอจี้ล หลังจากการขึ้นระบบในสภาพแวดล้อมจริง

ภาคผนวก ข

แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงาน
แบบออนไลน์





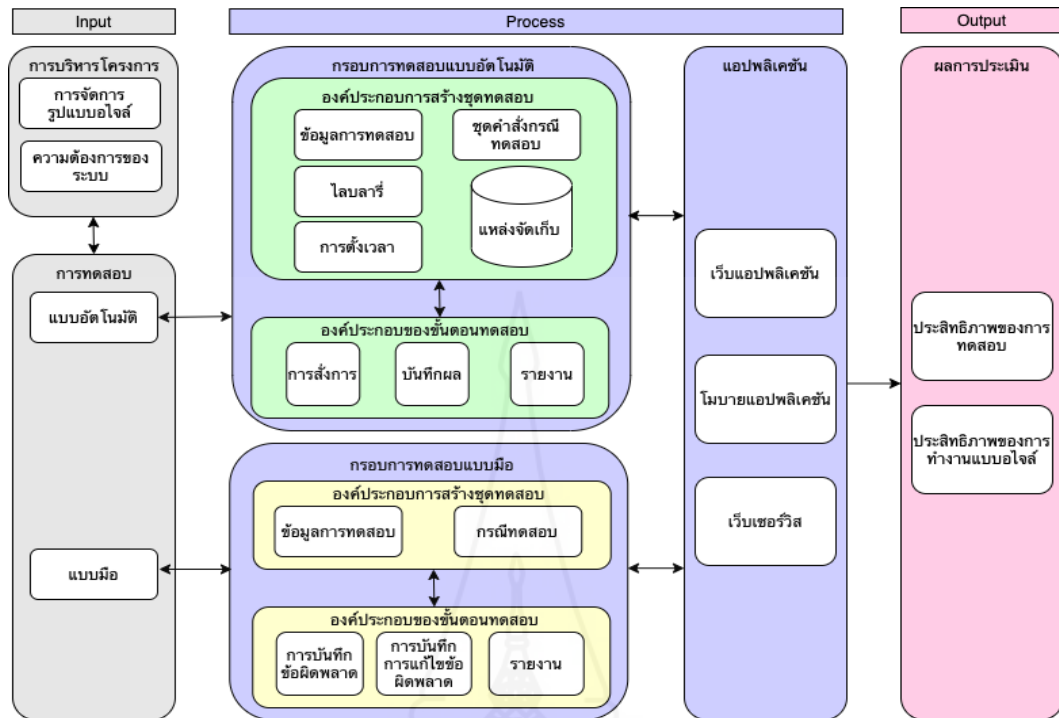
**แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ
ภายใต้รูปแบบการทำงานแบบออนไลน์**

คำชี้แจง

1. แบบสอบถามฉบับนี้มีทั้งหมด 17 หน้า มีความประสงค์ที่จะรวบรวมข้อมูลความคิดเห็นของผู้เชี่ยวชาญในการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ เพื่อนำข้อมูลไปวิเคราะห์ประเมินผล การทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ ซึ่งเป็นส่วนหนึ่งของวิทยานิพนธ์ มหาวิทยาลัยสุโขทัยธรรมาธิราช โดยมีอาจารย์ ดร.ศรันย์ นาคถนอม เป็นที่ปรึกษา และ รองศาสตราจารย์ ดร.วรัญญา ปุณณวัฒน์ เป็นที่ปรึกษาร่วม
2. แบบสอบถามฉบับนี้เป็นการสอบถามความคิดเห็นประเภท Rating Scale เกี่ยวกับประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์
3. แบบสอบถามฉบับนี้เป็นการสอบถามความคิดเห็นเกี่ยวกับประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ โดยแบบสอบถามได้กำหนดองค์ประกอบ เพื่อเป็นกรอบให้ผู้เชี่ยวชาญ ท่านสามารถเพิ่มหรือตัดองค์ประกอบใด ๆ ได้ โดยท่านสามารถเสนอแนะความคิดเห็นในแบบประเมินฉบับนี้ ความคิดเห็นของท่านจะเป็นประโยชน์อย่างยิ่งในการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบออนไลน์ ให้มีความสมบูรณ์ยิ่งขึ้น

ขอกราบขอบพระคุณในความกรุณาของท่านที่ได้แสดงความคิดเห็นที่เป็นประโยชน์ต่อการวิจัยครั้งนี้

นาย อภิลิทธิ์ เชนนาม
ผู้วิจัย



รูปที่ 1 แบบจำลองการทดสอบแบบภายใต้รูปแบบการทำงานแบบอไจล์

Agile คือ กรอบแนวคิด และวิธีการทำงานอย่างคล่องแคล่วรวดเร็ว เพื่อปรับเปลี่ยนให้สามารถตอบสนองความต้องการของลูกค้าได้อย่างรวดเร็วในสภาวะที่มีความไม่แน่นอน โดยมีวงจรชีวิตการทำงานแบบอไจล์ หรือ Agile SDLC Model ดังนี้

1. **ปัญหาหรือความต้องการของระบบ (Requirement)** คือ ปัญหาหรือความต้องการของระบบ (Requirement) ได้มาจากการที่ผู้ใช้งาน (User) นั้นมีความต้องการหรือพบปัญหาเกี่ยวกับระบบที่ใช้งานอยู่และต้องการให้มีการแก้ไขหรือเพิ่มฟีเจอร์ (Feature) ให้กับระบบ ผู้ใช้งานจะให้เอกสารปัญหาหรือความต้องการ (Requirement) มาในรูปแบบ User Story ซึ่งทีมงานอไจล์ (Agile Methodology) จะต้องรับปัญหาหรือความต้องการ (Requirement) ไปสร้างเป็นงานตามลักษณะงานของตัวเอง

2. **ออกแบบระบบ (Design)** คือ ผู้วิเคราะห์และออกแบบระบบ (System Analyst) เป็นผู้รับ ปัญหาหรือความต้องการ (Requirement) จากผู้ใช้งาน (User) และมาทำการวิเคราะห์และออกแบบแนวทางการพัฒนาซอฟต์แวร์เพื่อให้ทีมพัฒนาระบบ (Developer) นำไปพัฒนาให้ตรงกับความต้องการของผู้ใช้งาน (User)

3. **พัฒนาระบบ (Development)** คือ ผู้พัฒนาระบบ (Developer / Programmer) เป็นผู้ที่ทำหน้าที่พัฒนาซอฟต์แวร์ตามที่ผู้วิเคราะห์และออกแบบระบบ (System Analyst) ได้

ออกแบบไว้โดยสอดคล้องกับปัญหาหรือความต้องการของระบบ (Requirement) ของผู้ใช้งาน (User) ให้สำเร็จถึงจะนำซอฟต์แวร์ที่เสร็จสมบูรณ์ส่งให้ทีมทดสอบระบบ (Software Tester / QA) ทำการตรวจสอบ

4. ทดสอบระบบ (Software Testing) เป็นการตรวจสอบความถูกต้องของซอฟต์แวร์หลังจากที่ผู้พัฒนาระบบ (Developer / Programmer) ทำการพัฒนาเสร็จสมบูรณ์ ซึ่งการทดสอบระบบนั้นต้องใช้เวลาและความละเอียดรอบคอบ ซึ่งการทำงานแบบอไจล์ (Agile Methodology) เป็นการทำงานที่ต้องอาศัยความรวดเร็วและถูกต้องแม่นยำ ผู้จัดทำวิจัยจึงได้นำการทดสอบแบบอัตโนมัติ (Automation Testing) โดยใช้โรบอตเฟรมเวิร์ค (Robot Framework) มาแทนการทดสอบระบบแบบมือ (Manual Testing) เพื่อเป็นการลดเวลาและความผิดพลาดที่จะเกิดขึ้นจากมนุษย์ (Human Error)

4.1 วิธีการทดสอบระบบแบบมือ (Manual Testing)

คือการจำลองหรือออกแบบกรณีทดสอบ เพื่อยืนยันความถูกต้องของระบบ ผู้ทดสอบซอฟต์แวร์ (Software Tester) ทำการทดสอบซอฟต์แวร์/เปรียบเทียบผลการทดสอบ (Test Execution) โดยผู้ทดสอบจะทำการทดสอบตามกรณีทดสอบ (Test Cases) โดยผู้ทดสอบเอง โดยอ้างอิงความถูกต้องของการทำงานของระบบจากเอกสารความต้องการของลูกค้า (User Requirement) โดยแบ่งขั้นตอนการทดสอบระบบแบบมือ (Manual Testing) ออกเป็น 6 ขั้นตอน ดังนี้

1 การวิเคราะห์ความต้องการของผู้ใช้ระบบ (Requirement Analysis) การวิเคราะห์ความต้องการของผู้ใช้ระบบ เป็นขั้นตอนการสำคัญของการทดสอบระบบโดยเป็นกระบวนการตรวจสอบข้อมูลขั้นพื้นฐานในการทดสอบในแต่ละกรณี รวมไปถึงการออกแบบข้อกำหนดการทดสอบ และ ความเสี่ยงต่าง ๆ ที่จะมีโอกาสเกิดขึ้นในระบบ

2 การวางแผนการทดสอบระบบ (Test Plan Creation) การวางแผนการทดสอบระบบเป็นขั้นตอนการออกแบบแนวทางรวมถึงวัตถุประสงค์ของการทดสอบ โดยอาศัยการวิเคราะห์ความต้องการของผู้ใช้ระบบ ในการเขียนกรณีทดสอบรวมไปถึงการกำหนดสภาพแวดล้อมการทดสอบ และระบุเกณฑ์ความสำเร็จว่าต้องอยู่ที่ระดับใดในกรณีที่ทำกรทดสอบเสร็จสมบูรณ์

3 การออกแบบเอกสารกรณีทดสอบระบบ (Test Case Creation) เป็นการพัฒนาและจัดลำดับความสำคัญในแต่ละกรณีทดสอบ โดยใช้การสร้างข้อมูลการทดสอบในการทดสอบตามกรณีทดสอบต่าง ๆ

4 การทดสอบระบบตามเอกสารกรณีการทดสอบระบบ (Test Case Execution) คือการดำเนินการทดสอบระบบตามกรณีทดสอบที่พัฒนาไว้ โดยใช้ผู้ทดสอบระบบเป็นผู้ที่ระบุและเก็บผลการทดสอบว่าแต่ละกรณีเป็นไปตามความต้องการของผู้ใช้ระบบหรือไม่

5 การรายงานข้อผิดพลาดของระบบ (Defect Logging) คือการบันทึกข้อผิดพลาดของระบบที่ผู้ทดสอบระบบพบข้อบกพร่องระหว่างการทดสอบ โดยการรายงานข้อผิดพลาดนี้จะต้องส่งไปให้ผู้ทดสอบระบบเป็นผู้แก้ไขให้เป็นไปตามความต้องการของผู้ใช้ระบบ

6 การทดสอบระบบซ้ำหลังจากเจอข้อผิดพลาดของระบบ (Re-Verification) คือการทดสอบกรณีทดสอบที่ผิดพลาดซ้ำหลังจากที่ผู้พัฒนาระบบได้ทำการแก้ไขข้อผิดพลาดจากการรายงานข้อผิดพลาด เพื่อพิสูจน์ว่ากรณีทดสอบนั้นถูกต้องตามความต้องการของผู้ใช้ระบบ

4.2 การทดสอบระบบแบบอัตโนมัติ (Automation Testing)

คือการสร้างชุดกรณีทดสอบโดยสั่งงานผ่านซอฟต์แวร์ โดยนิยมนำมาใช้ในการทดสอบเชิงถดถอย เพื่อลดเวลาและข้อผิดพลาดจากการทดสอบโดยคน ในการทดสอบในกรณีที่ระบบมีขนาดใหญ่ และสามารถนำชุดทดสอบมาใช้ซ้ำได้ตามต้องการเนื่องจากเป็นโปรแกรมที่ทำงานด้วยคอมพิวเตอร์ การทำงานแต่ละครั้งจึงไม่แตกต่างกัน โดยมีโครงสร้างที่ใช้ในการกำหนดแนวทางในการพัฒนาและบำรุงรักษาดังนี้

1. Project Management คือการทราบถึงรูปแบบการทำงานขององค์กร ว่าใช้รูปแบบการทำงานแบบไหนในการทดสอบระบบแบบอัตโนมัติ
2. Test Management เป็นกระบวนการสร้างเอกสารกรณีทดสอบต่าง ๆ ตามเอกสารความต้องการของลูกค้า เพื่อทดสอบว่าผลลัพธ์เป็นไปตามที่คาดหวัง
3. Test Generating Layer เป็นกระบวนการที่จะสร้าง Test Script ขึ้นมาเพื่อใช้ในการทดสอบ ซึ่งการสร้าง Test Script นั้นต้องประกอบไปด้วย ข้อมูลการทดสอบ (Test Data) เป็นข้อมูลที่ถูกสร้างขึ้นเพื่อเป็นข้อมูล Input สำหรับการทดสอบระบบ Test Script คือการพัฒนากรณีทดสอบให้อยู่ในรูปแบบของการ Coding เพื่อให้สามารถ Execute Test ได้ตามภาษาโปรแกรมที่ Framework นั้น ๆ รองรับ และมี Test Library เป็นตัวเพิ่มความสามารถของ Test Script ซึ่งการทำ Test Script ในแต่ละครั้งนั้นควรจะต้องมี Object Repository เพื่อใช้ในการเก็บ Version ของ Test Script เมื่อมี Test Script ที่ครอบคลุมทุกกรณีทดสอบแล้วสิ่งต่อมาคือการ ทดสอบแบบอัตโนมัติ (Test Execution Layer) ซึ่งการรันทดสอบแบบอัตโนมัติ นั้นไม่ว่าจะรันทดสอบด้วย

Environment ไหน หรือ ระบบไหน ต้องสามารถเก็บผลการทดสอบและสามารถออกรายงานผลการทดสอบได้

5. การขึ้นระบบ (Deployment) คือ ผู้พัฒนาระบบ (Developer / Programmer / Devops) ทำหน้าที่นำซอฟต์แวร์ที่ผ่านกระบวนการทดสอบเรียบร้อยแล้วขึ้นไปให้ผู้ใช้งาน (User) จริงได้ใช้งาน



แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์

คำชี้แจง

แบบสอบถามงานชุดนี้ มีวัตถุประสงค์เพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ โดยมีข้อคำถามแบ่งออกเป็น 5 หมวดดังต่อไปนี้

1. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Automation Testing with Agile Testing)
2. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการรับปัญหาหรือความต้องการของระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Requirement with Agile Testing)
3. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการออกแบบระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (System Analyze with Agile Testing)
4. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการพัฒนาระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)
5. ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการขึ้นระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)

โดยส่วนประกอบของแบบสอบถาม ประกอบด้วย 3 ตอนดังนี้

ตอนที่ 1 ข้อมูลส่วนตัวของผู้ประเมิน

ตอนที่ 2 : การแสดงความคิดเห็นของผู้ตอบแบบสอบถามเกี่ยวกับการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์

ตอนที่ 3 : ข้อเสนอแนะ

ตอนที่ 1 ข้อมูลส่วนตัวของผู้ประเมิน

1. ตำแหน่งงาน
2. ประเภทของหน่วยงาน เอกชน รัฐบาล
3. ระดับการศึกษา

ตอนที่ 2 : การแสดงความคิดเห็นของผู้ตอบแบบสอบถามเกี่ยวกับการประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ ที่พัฒนาขึ้น ประกอบด้วยส่วนของข้อคำถามที่อยู่ด้านซ้ายมือ และมาตรฐานประมาณค่าที่อยู่ด้านขวามือ

หมวดที่ 1 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Automation Testing with Agile Testing)

กรณารอกข้อมูลจำนวนกรณีทดสอบ (Test Cases) และระดับความซ้ำซ้อน ในกระบวนการทำงานแบบอไจล์ เพื่อประเมินผลของระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ โดยค่าระดับความซ้ำซ้อน มีดังนี้

Low	= ระดับความซ้ำซ้อนต่ำ
Average	= ระดับความซ้ำซ้อนปานกลาง
High	= ระดับความซ้ำซ้อนสูง

คำชี้แจง : โปรดใส่เครื่องหมาย ลงในช่อง และเพิ่มเติมข้อเสนอแนะลงในช่องว่าง

A1 ขนาดของซอฟต์แวร์ (size, test cases)

A1.1 จำนวนกรณีทดสอบ (Test Cases)

A1.2 ระดับความซ้ำซ้อน (Redundancy Level)

ปัญหาหรือความต้องการของระบบ (Requirement)	จำนวนกรณีทดสอบ (Test Cases)	ระดับความซ้ำซ้อน (Redundancy Level)
1. จำนวนกรณีทดสอบของแต่ละหน่วย ของระบบ	<input type="checkbox"/> Low <input type="checkbox"/> Average <input type="checkbox"/> High

ตัวอย่าง หากซอฟต์แวร์ มีจำนวนกรณีทดสอบ (Test Case) จำนวน 30 Test Cases ต่อ Function, Object or Application point และมีระดับความยาก / สลับซับซ้อนค่อนข้างสูง (ตามความคิดเห็นของผู้ทดสอบระบบ) ให้กรอข้อมูลดังนี้

ปัญหาหรือความต้องการของระบบ (Requirement)	จำนวนกรณีทดสอบ (Test Cases)	ระดับความซ้ำซ้อน (Redundancy Level)
1. จำนวนกรณีทดสอบของแต่ละหน่วย ของระบบ30.....	<input type="checkbox"/> Low <input type="checkbox"/> Average <input checked="" type="checkbox"/> High

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อ ระยะเวลาและลดข้อผิดพลาดจากการ ทดสอบแบบอัตโนมัติ	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
A2 การทดสอบแบบอัตโนมัติมีความ น่าเชื่อถือในการนำไปใช้กับรูปแบบ การทำงานแบบอไจล์						
A3 การทดสอบแบบอัตโนมัติช่วยลด ความซับซ้อนของระบบที่พัฒนาขึ้น ภายใต้รูปแบบการทำงานแบบอไจล์						
A4 การทดสอบแบบอัตโนมัติมี ประสิทธิภาพ – ความเร็ว เมื่อนำมาใช้ กับรูปแบบการทำงานแบบอไจล์						
A5 การทดสอบแบบอัตโนมัติมีความ ถูกต้อง – แม่นยำ ของผลการทดสอบ เมื่อนำมาใช้กับรูปแบบการทำงาน แบบอไจล์						

หมวดที่ 2 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการรับปัญหาหรือความต้องการของระบบกับการ
ทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Requirement with Agile
Testing)

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผล ต่อระยะเวลาและลดข้อผิดพลาดจาก การทดสอบแบบอัตโนมัติ	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
B1 ความสามารถของผู้จัดทำเอกสาร ความต้องการของระบบของผู้ใช้งาน มีผลกับการทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการทำงาน แบบอไจล์						
B2 เอกสารความต้องการของระบบ ของผู้ใช้งาน มีผลต่อความน่าเชื่อถือ						

ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
B3 เอกสารความต้องการของระบบของผู้ใช้งาน ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
B4 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
B5 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						

หมวดที่ 3 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการออกแบบระบบกับการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ (System Analyze with Agile Testing)

ตัวชี้วัด	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
กระบวนการทำงานแบบอไจล์มีผลต่อระยะเวลาและลดข้อผิดพลาดจากการทดสอบแบบอัตโนมัติ						
C1 ความสามารถของผู้วิเคราะห์ระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
C2 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อความน่าเชื่อถือของทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
C3 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ ช่วย						

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อ ระยะเวลาและลดข้อผิดพลาดจากการ ทดสอบแบบอัตโนมัติ	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
ลดความซับซ้อนในการทดสอบระบบ แบบอัตโนมัติภายใต้รูปแบบการ ทำงานแบบอไจล์						
C4 การวิเคราะห์และออกแบบระบบ ตามความต้องการของผู้ใช้ระบบ มีผล ต่อระยะเวลาในการทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการ ทำงานแบบอไจล์						
C5 การวิเคราะห์และออกแบบระบบ ตามความต้องการของผู้ใช้ระบบ มีผล ต่อความถูกต้อง – แม่นยำ ของการ ทดสอบระบบแบบอัตโนมัติภายใต้ รูปแบบการ ทำงานแบบอไจล์						

หมวดที่ 4 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการพัฒนาระบบกับการทดสอบแบบอัตโนมัติ
ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)

ตัวชี้วัด กระบวนการทำงานแบบอไจล์มีผลต่อ ระยะเวลาและลดข้อผิดพลาดจากการ ทดสอบแบบอัตโนมัติ	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
D1 ความสามารถของผู้พัฒนาระบบ มี ผลกับการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์						
D2 การพัฒนาระบบ มีผลต่อความ น่าเชื่อถือของทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการทำงาน แบบอไจล์						
D3 การพัฒนาระบบ ช่วยลดความ ซับซ้อนในการทดสอบระบบแบบ						

อัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
D4 การพัฒนาระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						
D5 การพัฒนาระบบ มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์						

หมวดที่ 5 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการขึ้นระบบกับการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ (Deployment with Agile Testing)

ตัวชี้วัด	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
E1 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ ส่งผลต่อความน่าเชื่อถือของระบบ						
E2 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ ช่วยลดความเสี่ยงในการขึ้นระบบจริงไม่สำเร็จ						
E3 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ มีผลต่อประสิทธิภาพ – ความเร็ว ของระบบในขั้นตอนการขึ้นระบบจริง						
E4 การทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์ มีผลต่อความถูกต้อง – แม่นยำ ของระบบในขั้นตอนการขึ้นระบบจริง						

ตัวชี้วัด กระบวนการทำงานแบบบอจี้มีผลต่อ ระยะเวลาและลดข้อผิดพลาดจากการ ทดสอบแบบอัตโนมัติ	ระดับความคิดเห็น					ข้อเสนอแนะ
	5 มาก ที่สุด	4 มาก	3 ปาน กลาง	2 น้อย	1 น้อย ที่สุด	
E5 การขึ้นระบบจริง โดยผ่านการ ทดสอบแบบอัตโนมัติภายใต้รูปแบบ การทำงานแบบบอจี้มีผลต่อความพึง พอใจ ในการใช้งานระบบของผู้ใช้						

ตอนที่ 3 : ข้อเสนอแนะ

.....

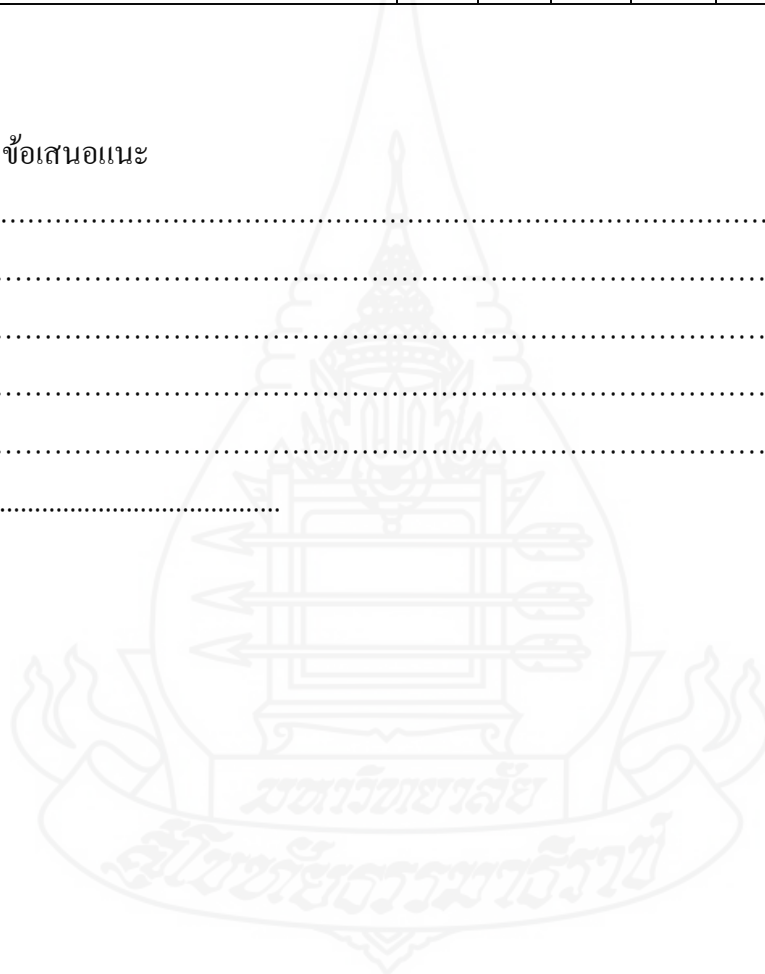
.....

.....

.....

.....

.....



คำอธิบายตัวชี้วัด

หมวดที่ 1 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Automation Testing with Agile Testing)

ตัวชี้วัด	คำอธิบาย
1.1 ขนาดของซอฟต์แวร์	เป็นการพิจารณาขนาดของระบบที่จะทดสอบว่ามีขนาดเท่าไร โดยพิจารณาจาก กรณีทดสอบ (Test Case) ที่ครอบคลุมทุกหน่วยการทำงานย่อยระบบ และมีระดับความยาก / สลับซับซ้อนเท่าไร (ตามความคิดเห็นของผู้ทดสอบระบบ)
1.2 การทดสอบแบบอัตโนมัติมีความน่าเชื่อถือในการนำไปใช้กับรูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติโดยใช้กระบวนการทำงานแบบอไจล์ มีผลต่อระยะเวลาและลดข้อผิดพลาดในการทดสอบ และทำให้ระบบที่พัฒนาขึ้นมีความน่าเชื่อถือ
1.3 การทดสอบแบบอัตโนมัติช่วยลดความซับซ้อนของระบบที่พัฒนาขึ้นภายใต้รูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติ มีผลต่อระยะเวลาและลดข้อผิดพลาดในการทดสอบ และช่วยลดความซับซ้อน ในกระบวนการทำงานแบบอไจล์
1.4 การทดสอบแบบอัตโนมัติมีประสิทธิภาพ – ความเร็ว เมื่อนำมาใช้กับรูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติภายใต้กระบวนการทำงานแบบอไจล์ มีผลต่อระยะเวลา ลดข้อผิดพลาด และทำให้ระบบมีประสิทธิภาพ
1.5 การทดสอบแบบอัตโนมัติมีความถูกต้อง – แม่นยำ ของผลการทดสอบ เมื่อนำมาใช้กับรูปแบบการทำงานแบบอไจล์	ซอฟต์แวร์ที่ถูกทดสอบแบบอัตโนมัติภายใต้กระบวนการทำงานแบบอไจล์ มีผลต่อระยะเวลา ลดข้อผิดพลาด ของผลการทดสอบ

หมวดที่ 2 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการรับปัญหาหรือความต้องการของระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Requirement with Agile Testing)

ตัวชี้วัด	คำอธิบาย
2.1 ความสามารถของผู้จัดทำเอกสารความต้องการของระบบของผู้ใช้งาน มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ความสามารถของผู้จัดทำเอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบที่นำไปใช้ในการทดสอบแบบอัตโนมัติ
2.2 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความน่าเชื่อถือของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งส่งผลต่อความน่าเชื่อถือของการทดสอบระบบทดสอบแบบอัตโนมัติ
2.3 เอกสารความต้องการของระบบของผู้ใช้งาน ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งช่วยลดความซับซ้อนของการทดสอบระบบทดสอบแบบอัตโนมัติ
2.4 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งมีผลต่อระยะเวลาในการทดสอบระบบทดสอบแบบอัตโนมัติ
2.5 เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อความถูกต้อง – แม่นยำของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	เอกสารความต้องการของระบบของผู้ใช้งาน มีผลต่อการออกแบบกรณีทดสอบ ซึ่งมีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบทดสอบแบบอัตโนมัติ

หมวดที่ 3 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการออกแบบระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (System Analyze with Agile Testing)

ตัวชี้วัด	คำอธิบาย
3.1 ความสามารถของผู้วิเคราะห์ระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ผู้วิเคราะห์ระบบที่สามารถออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ต่อการออกแบบกรณีทดสอบได้ครอบคลุม
3.2 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อ	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลต่อการออกแบบกรณีทดสอบได้ครอบคลุม

ตัวชี้วัด	คำอธิบาย
ความน่าเชื่อถือของทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	คลุม ซึ่งทำให้ทดสอบระบบได้อย่างครบถ้วน และทำให้ระบบมีความน่าเชื่อถือ
3.3 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ ช่วยลดความซับซ้อนในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ช่วยลดความซ้ำซ้อนในการวางแผนและออกแบบกรณีทดสอบ
3.4 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อระยะเวลาในการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ต่อการออกแบบกรณีทดสอบได้อย่างครอบคลุม ซึ่งทำให้ทดสอบระบบได้อย่างครบถ้วน และช่วยลดระยะเวลาในการทดสอบ
3.5 การวิเคราะห์และออกแบบระบบตามความต้องการของผู้ใช้ระบบ มีผลต่อความถูกต้อง – แม่นยำ ของการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ระบบที่มีการออกแบบการทำงานของระบบได้ชัดเจน ส่งผลให้ต่อการออกแบบกรณีทดสอบได้อย่างครอบคลุม ซึ่งทำให้ทดสอบระบบได้อย่างครบถ้วน และช่วยลดระยะเวลาในการทดสอบ

หมวดที่ 4 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการพัฒนาระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบอไจล์ (Development with Agile Testing)

ตัวชี้วัด	คำอธิบาย
4.1 ความสามารถของผู้พัฒนาระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์	ความสามารถของผู้พัฒนาระบบ ที่สามารถพัฒนาระบบได้เป็นไปตามเอกสารความต้องการของลูกค้า และตรงตามการวิเคราะห์และออกแบบระบบ มีผลกับการทดสอบระบบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอไจล์
4.2 การพัฒนาระบบ มีผลต่อความน่าเชื่อถือของทดสอบระบบแบบ	การพัฒนาระบบได้ตรงตามเอกสารความต้องการของลูกค้า และตรงตามการวิเคราะห์และออกแบบ

ตัวชี้วัด	คำอธิบาย
อัตโนมัติภายใต้รูปแบบการทำงานแบบบอ ไจล์	ระบบ ส่งผลให้การทดสอบได้ครอบคลุม ซึ่งทำให้ ทดสอบระบบได้อย่างครบถ้วน และทำให้ระบบมี ความน่าเชื่อถือ
4.3 การพัฒนาระบบ ช่วยลดความ ซับซ้อนในการทดสอบระบบแบบ อัตโนมัติภายใต้รูปแบบการทำงานแบบบอ ไจล์	การพัฒนาระบบได้ตรงตามเอกสารความต้องการ ของลูกค้า และตรงตามการวิเคราะห์และออกแบบ ระบบ ช่วยลดความซับซ้อนในการทดสอบ
4.4 การพัฒนาระบบ มีผลต่อระยะเวลา ในการทดสอบระบบแบบอัตโนมัติ ภายใต้รูปแบบการทำงานแบบบอไจล์	การพัฒนาระบบได้ตรงตามเอกสารความต้องการ ของลูกค้า และตรงตามการวิเคราะห์และออกแบบ ระบบ ส่งผลให้ระยะเวลาการทดสอบระบบแบบ อัตโนมัติลดลง หรือทำการทดสอบสำเร็จได้ตาม เป้าหมาย
4.5 การพัฒนาระบบ มีผลต่อความ ถูกต้อง – แม่นยำ ของการทดสอบระบบ แบบอัตโนมัติภายใต้รูปแบบการทำงาน แบบบอไจล์	การพัฒนาระบบได้ตรงตามเอกสารความต้องการ ของลูกค้า และตรงตามการวิเคราะห์และออกแบบ ระบบ ส่งผลให้การทดสอบระบบแบบอัตโนมัติมี ความถูกต้อง แม่นยำ และลดการเกิดข้อผิดพลาดที่จะ เกิดขึ้นได้ของระบบ

หมวดที่ 5 ตัวชี้วัดที่เกี่ยวข้องของขั้นตอนการขึ้นระบบกับการทดสอบแบบอัตโนมัติ ภายใต้รูปแบบ
การทำงานแบบบอไจล์ (Deployment with Agile Testing)

ตัวชี้วัด	คำอธิบาย
5.1 การขึ้นระบบจริง โดยผ่านการ ทดสอบแบบอัตโนมัติภายใต้รูปแบบการ ทำงานแบบบอไจล์ ส่งผลต่อความ น่าเชื่อถือของระบบ	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตาม เอกสารความต้องการของลูกค้า สามารถสร้างความ เชื่อถือให้กับผู้ใช้งานจริงได้ ในกระบวนการพัฒนา ซอฟต์แวร์แบบบอไจล์ หลังจากการขึ้นระบบใน สภาพแวดล้อมจริง
5.2 การขึ้นระบบจริง โดยผ่านการ ทดสอบแบบอัตโนมัติภายใต้รูปแบบการ ทำงานแบบบอไจล์ ช่วยลดความเสี่ยงใน	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตาม เอกสารความต้องการของลูกค้า สามารถช่วยลดความ เสี่ยงให้การยกเลิก หรือ การขึ้นระบบบน

การขึ้นระบบในสภาพแวดล้อมจริงไม่สำเร็จ	สภาพแวดล้อมจริงไม่สำเร็จ เนื่องจากสาเหตุของระบบไม่สมบูรณ์
5.3 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้ล มีผลต่อประสิทธิภาพ – ความเร็ว ของระบบในขั้นตอนการขึ้นระบบจริง	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตามเอกสารความต้องการของลูกค้า สามารถทำให้การขึ้นระบบบนสภาพแวดล้อมจริงได้อย่างรวดเร็วและมีประสิทธิภาพ
5.4 การทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้ล มีผลต่อความถูกต้อง – แม่นยำ ของระบบในขั้นตอนการขึ้นระบบจริง	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตามเอกสารความต้องการของลูกค้า สามารถทำให้การขึ้นระบบบนสภาพแวดล้อมจริงได้อย่างถูกต้องและแม่นยำ
5.5 การขึ้นระบบจริง โดยผ่านการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอจี้ล มีผลต่อความพึงพอใจในการใช้งานระบบของผู้ใช้	ผลการทดสอบระบบแบบอัตโนมัติที่ถูกต้องตามเอกสารความต้องการของลูกค้า สามารถสร้างความพึงพอใจให้กับผู้ใช้งานจริงได้ ในกระบวนการพัฒนาซอฟต์แวร์แบบบอจี้ล หลังจากการขึ้นระบบในสภาพแวดล้อมจริง





ภาคผนวก ค

หนังสือขอความอนุเคราะห์เป็นผู้เชี่ยวชาญ



ที่ อว ๐๖๐๒.๒๕/๘๗๘

สาขาวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
ตำบลบางพูด อำเภอปากเกร็ด
จังหวัดนนทบุรี ๑๑๑๒๐

๑๙ สิงหาคม ๒๕๖๔

เรื่อง ขอบความอนุเคราะห์เป็นผู้เชี่ยวชาญตรวจสอบคุณภาพเครื่องมือวิจัย

เรียน ผู้ช่วยศาสตราจารย์ ดร. รุ่งทิวา เสาร์สิงห์

สิ่งที่ส่งมาด้วย แบบสอบถามประกอบการทำวิทยานิพนธ์ จำนวน ๑ ชุด

ด้วย นาย อภิสิตธิ์ เขยนาม รหัสประจำตัวนักศึกษา ๒๖๑๙๖๐๐๒๒๐ นักศึกษาปริญญาโท
หลักสูตรวิทยาศาสตรมหาบัณฑิต แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์
และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช ปัจจุบันอยู่ระหว่างการทำวิทยานิพนธ์เรื่อง "การทดสอบ
ซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอโจลส์" โดยมี
อาจารย์ ดร.ศรันย์ นาคถนอม เป็นอาจารย์ที่ปรึกษา

ในการนี้ สาขาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช พิจารณาแล้วเห็นว่า
ท่านเป็นผู้ที่มีคุณสมบัติเหมาะสม จึงใคร่ขอความอนุเคราะห์จากท่าน ได้กรุณาเป็นผู้เชี่ยวชาญในการ
ตรวจสอบความเที่ยงตรงตามวัตถุประสงค์ของแบบสอบถาม ตลอดจนให้ข้อเสนอแนะอื่น ๆ แก่นักศึกษา
เพื่อนำไปใช้สำหรับปรับปรุงเครื่องมือวิจัยต่อไป พร้อมนี้ได้แนบบแบบสอบถามประกอบการทำวิทยานิพนธ์
มาด้วยแล้ว ได้แก่ ๑. แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติภายใต้รูปแบบการ
ทำงานแบบอโจลส์ (Item - Objective Congruence Index: IOC) โดยมีรายละเอียดตามสิ่งที่ส่งมาด้วย

จึงเรียนมาเพื่อโปรดให้ความอนุเคราะห์ และขอขอบคุณมา ณ โอกาสนี้ สำหรับการดำเนินการ
นักศึกษาจะได้ประสานในรายละเอียดเพิ่มเติมกับท่านโดยตรงต่อไป

ขอแสดงความนับถือ

Sitthichai

(อาจารย์ ดร.สิทธิชัย รัชชโยธิน)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

สาขาวิชาวิทยาศาสตร์และเทคโนโลยี

โทรศัพท์ ๐ ๒๕๐๔ ๘๒๘๑

โทรสาร ๐ ๒๕๐๓ ๔๙๓๒



ที่ อว ๐๖๐๒.๒๕/๘๗๘

สาขาวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
ตำบลบางพูด อำเภอปากเกร็ด
จังหวัดนนทบุรี ๑๑๑๒๐

๑๙ สิงหาคม ๒๕๖๔

เรื่อง ขอความอนุเคราะห์เป็นผู้เชี่ยวชาญตรวจสอบคุณภาพเครื่องมือวิจัย

เรียน รองศาสตราจารย์นศัพธ์ชาณัณ ชินปัญช์ธนะ

สิ่งที่ส่งมาด้วย แบบสอบถามประกอบการทำวิทยานิพนธ์ จำนวน ๑ ชุด

ด้วย นาย อภิสิตธิ์ เขยนาม รหัสประจำตัวนักศึกษา ๒๖๑๙๖๐๐๒๒๐ นักศึกษาปริญญาโท หลักสูตรวิทยาศาสตรมหาบัณฑิต แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช ปัจจุบันอยู่ระหว่างการทำวิทยานิพนธ์เรื่อง "การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอัจล์" โดยมีอาจารย์ ดร.ศรันย์ นาคถนอม เป็นอาจารย์ที่ปรึกษา

ในการนี้ สาขาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช พิจารณาแล้วเห็นว่า ท่านเป็นผู้ที่มีคุณสมบัติเหมาะสม จึงใคร่ขอความอนุเคราะห์จากท่าน ได้กรุณาเป็นผู้เชี่ยวชาญในการตรวจสอบความเที่ยงตรงตามวัตถุประสงค์ของแบบสอบถาม ตลอดจนให้ข้อเสนอแนะอื่น ๆ แก่นักศึกษา เพื่อนำไปใช้สำหรับปรับปรุงเครื่องมือวิจัยต่อไป ทั้งนี้ได้แนบแบบสอบถามประกอบการทำวิทยานิพนธ์ มาด้วยแล้ว ได้แก่ ๑. แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอัจล์ (Item - Objective Congruence Index: IOC) โดยมีรายละเอียดตามสิ่งที่ส่งมาด้วย

จึงเรียนมาเพื่อโปรดให้ความอนุเคราะห์ และขอขอบคุณมา ณ โอกาสนี้ สำหรับการดำเนินการ นักศึกษาจะได้ประสานในรายละเอียดเพิ่มเติมกับท่านโดยตรงต่อไป

ขอแสดงความนับถือ

Sitthichai

(อาจารย์ ดร.สิทธิชัย รัชยศโยธิน)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

สาขาวิชาวิทยาศาสตร์และเทคโนโลยี

โทรศัพท์ ๐ ๒๕๐๔ ๘๑๙๓

โทรสาร ๐ ๒๕๐๓ ๔๙๓๒



ที่ อว ๐๖๐๒.๒๕/๘๗๘

สาขาวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
ตำบลบางพูด อำเภอปากเกร็ด
จังหวัดนนทบุรี ๑๑๑๒๐

๑๙ สิงหาคม ๒๕๖๔

เรื่อง ขอบความอนุเคราะห์เป็นผู้เชี่ยวชาญตรวจสอบคุณภาพเครื่องมือวิจัย

เรียน ผู้ช่วยศาสตราจารย์ ดร. จันทรัตน์ กิ่งแสง

สิ่งที่ส่งมาด้วย แบบสอบถามประกอบการทำวิทยานิพนธ์ จำนวน ๑ ชุด

ด้วย นาย อภิสิตธิ์ เขยนาม รหัสประจำตัวนักศึกษา ๒๖๑๙๖๐๐๒๒๐ นักศึกษาปริญญาโท หลักสูตรวิทยาศาสตรมหาบัณฑิต แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช ปัจจุบันอยู่ระหว่างการทำวิทยานิพนธ์เรื่อง "การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอโจล์" โดยมีอาจารย์ ดร.ศรัณย์ นาคถนอม เป็นอาจารย์ที่ปรึกษา

ในการนี้ สาขาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช พิจารณาแล้วเห็นว่าท่านเป็นผู้ที่มีคุณสมบัติเหมาะสม จึงใคร่ขอความอนุเคราะห์จากท่าน ได้กรุณาเป็นผู้เชี่ยวชาญในการตรวจสอบความเที่ยงตรงตามวัตถุประสงค์ของแบบสอบถาม ตลอดจนให้ข้อเสนอแนะอื่น ๆ แก่นักศึกษาเพื่อนำไปใช้สำหรับปรับปรุงเครื่องมือวิจัยต่อไป พร้อมนี้ได้แนบแบบสอบถามประกอบการทำวิทยานิพนธ์มาด้วยแล้ว ได้แก่ ๑. แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอโจล์ (Item - Objective Congruence Index: IOC) โดยมีรายละเอียดตามสิ่งที่ส่งมาด้วย

จึงเรียนมาเพื่อโปรดให้ความอนุเคราะห์ และขอขอบคุณมา ณ โอกาสนี้ สำหรับการดำเนินการ นักศึกษาจะได้ประสานในรายละเอียดเพิ่มเติมกับท่านโดยตรงต่อไป

ขอแสดงความนับถือ

Sitthichai

(อาจารย์ ดร.สิทธิชัย รัชยศโยธิน)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

สาขาวิชาวิทยาศาสตร์และเทคโนโลยี

โทรศัพท์ ๐ ๒๕๐๔ ๘๑๙๓

โทรสาร ๐ ๒๕๐๓ ๔๙๓๒



ที่ อว ๐๖๐๒.๒๕/๘๗๘

สาขาวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
ตำบลบางพูด อำเภอปากเกร็ด
จังหวัดนนทบุรี ๑๑๑๒๐

๑๙ สิงหาคม ๒๕๖๔

เรื่อง ขอความอนุเคราะห์เป็นผู้เชี่ยวชาญตรวจสอบคุณภาพเครื่องมือวิจัย

เรียน คุณสุนิปรา สัยยะ

สิ่งที่ส่งมาด้วย แบบสอบถามประกอบการทำวิทยานิพนธ์ จำนวน ๑ ชุด

ด้วย นาย อภิสิตธิ์ เขยนาม รหัสประจำตัวนักศึกษา ๒๖๑๙๖๐๐๒๒๐ นักศึกษาปริญญาโท หลักสูตรวิทยาศาสตรมหาบัณฑิต แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช ปัจจุบันอยู่ระหว่างการทำวิทยานิพนธ์เรื่อง "การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบอัจล์" โดยมีอาจารย์ ดร.ศรันย์ นาคถนอม เป็นอาจารย์ที่ปรึกษา

ในการนี้ สาขาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช พิจารณาแล้วเห็นว่า ท่านเป็นผู้ที่มีคุณสมบัติเหมาะสม จึงใคร่ขอความอนุเคราะห์จากท่าน ได้กรุณาเป็นผู้เชี่ยวชาญในการตรวจสอบความเที่ยงตรงตามวัตถุประสงค์ของแบบสอบถาม ตลอดจนให้ข้อเสนอแนะอื่น ๆ แก่นักศึกษา เพื่อนำไปใช้สำหรับปรับปรุงเครื่องมือวิจัยต่อไป ทั้งนี้ได้แนบแบบสอบถามประกอบการทำวิทยานิพนธ์ มาด้วยแล้ว ได้แก่ ๑. แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบอัจล์ (Item - Objective Congruence Index: IOC) โดยมีรายละเอียดตามสิ่งที่ส่งมาด้วย

จึงเรียนมาเพื่อโปรดให้ความอนุเคราะห์ และขอขอบคุณมา ณ โอกาสนี้ สำหรับการดำเนินการ นักศึกษาจะได้ประสานในรายละเอียดเพิ่มเติมกับท่านโดยตรงต่อไป

ขอแสดงความนับถือ

Sithichai

(อาจารย์ ดร.สิทธิชัย รัชยศโยธิน)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

สาขาวิชาวิทยาศาสตร์และเทคโนโลยี

โทรศัพท์ ๐ ๒๕๐๔ ๘๑๙๓

โทรสาร ๐ ๒๕๐๓ ๔๙๓๒



ที่ อว ๐๖๐๒.๒๕/๘๗๘

สาขาวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยสุโขทัยธรรมมาธิราช
ตำบลบางพูด อำเภอปากเกร็ด
จังหวัดนนทบุรี ๑๑๑๒๐

๑๙ สิงหาคม ๒๕๖๔

เรื่อง ขอความอนุเคราะห์เป็นผู้เชี่ยวชาญตรวจสอบคุณภาพเครื่องมือวิจัย

เรียน คุณศุภธินี อุษชิน

สิ่งที่ส่งมาด้วย แบบสอบถามประกอบการทำวิทยานิพนธ์ จำนวน ๑ ชุด

ด้วย นาย อภิสิตธิ์ เขยนาม รหัสประจำตัวนักศึกษา ๒๖๑๙๖๐๐๒๒๐ นักศึกษาปริญญาโท หลักสูตรวิทยาศาสตรมหาบัณฑิต แขนงวิชาเทคโนโลยีสารสนเทศและการสื่อสาร สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช ปัจจุบันอยู่ระหว่างการทำวิทยานิพนธ์เรื่อง "การทดสอบซอฟต์แวร์แบบอัตโนมัติด้วยโรบอตเฟรมเวิร์คภายใต้รูปแบบการทำงานแบบบอโจล์" โดยมีอาจารย์ ดร.ศรันย์ นาคถนอม เป็นอาจารย์ที่ปรึกษา

ในการนี้ สาขาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช พิจารณาแล้วเห็นว่า ท่านเป็นผู้ที่มีคุณสมบัติเหมาะสม จึงใคร่ขอความอนุเคราะห์จากท่าน ได้กรุณาเป็นผู้เชี่ยวชาญในการตรวจสอบความเที่ยงตรงตามวัตถุประสงค์ของแบบสอบถาม ตลอดจนให้ข้อเสนอแนะอื่น ๆ แก่นักศึกษา เพื่อนำไปใช้สำหรับปรับปรุงเครื่องมือวิจัยต่อไป ทั้งนี้ได้แนบแบบสอบถามประกอบการทำวิทยานิพนธ์ มาด้วยแล้ว ได้แก่ ๑. แบบสอบถามเพื่อประเมินประสิทธิภาพการทดสอบแบบอัตโนมัติภายใต้รูปแบบการทำงานแบบบอโจล์ (Item - Objective Congruence Index: IOC) โดยมีรายละเอียดตามสิ่งที่ส่งมาด้วย

จึงเรียนมาเพื่อโปรดให้ความอนุเคราะห์ และขอขอบคุณมา ณ โอกาสนี้ สำหรับการดำเนินการ นักศึกษาจะได้ประสานในรายละเอียดเพิ่มเติมกับท่านโดยตรงต่อไป

ขอแสดงความนับถือ

Sitthichai

(อาจารย์ ดร.สิทธิชัย รัชยศโยธิน)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

สาขาวิชาวิทยาศาสตร์และเทคโนโลยี

โทรศัพท์ ๐ ๒๕๐๔ ๘๑๙๓

โทรสาร ๐ ๒๕๐๓ ๔๙๓๒

ประวัติผู้วิจัย

ชื่อ	นายอภิสิทธิ์ เชนาม
วัน เดือน ปีเกิด	13 ธันวาคม 2535
สถานที่เกิด	นนทบุรี
ประวัติการศึกษา	บริหารธุรกิจบัณฑิต (บธ.บ.) สาขาวิชาคอมพิวเตอร์ธุรกิจ มหาวิทยาลัยศรีปทุม กรุงเทพมหานคร
สถานที่ทำงาน	บริษัท เอสซีบีเทคเอ็กซ์ จำกัด
ตำแหน่ง	QA Engineer Testing Professional.

