

การพัฒนาระบบการทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119

DEVELOPMENT OF SOFTWARE TESTING SYSTEM BASED ON
ISO 29119

ชฎาธาร รักสวัสดิ์
CHADATARN RAKSAWAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

KMITL-2019-SC-D-002-006

การพัฒนาระบบการทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119

DEVELOPMENT OF SOFTWARE TESTING SYSTEM BASED ON
ISO 29119

ชฎาธาร รักสวัสดิ์

CHADATARN RAKSAWAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

KMITL-2019-SC-D-002-006

DEVELOPMENT OF SOFTWARE TESTING SYSTEM BASED ON
ISO 29119

CHADATARN RAKSAWAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2019

KMITL-2019-SC-D-002-006

COPYRIGHT 2019

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การพัฒนากระบวนการทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119
ชื่อนักศึกษา	ชฎาธาร รักสวัสดิ์
รหัสประจำตัว	60605076
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2562
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.ปัทมา เจริญพร

บทคัดย่อ

ในกระบวนการพัฒนาซอฟต์แวร์ควรมีการทดสอบซอฟต์แวร์ เพื่อตรวจสอบและปรับปรุงให้ซอฟต์แวร์มีความถูกต้อง สมบูรณ์ คุณภาพดี และสามารถระบุ ค้นหาหรือแก้ไขข้อผิดพลาดของซอฟต์แวร์ได้ การทดสอบซอฟต์แวร์ให้มีประสิทธิภาพ ถูกต้อง และเชื่อถือได้ จำเป็นต้องอ้างอิงมาตรฐานในการทดสอบซอฟต์แวร์ โดยศึกษาและทำความเข้าใจในมาตรฐานการทดสอบให้ครบถ้วน และละเอียด รอบคอบ ซึ่งเป็นสิ่งที่ยุ่งยาก เสียเวลา และยังเพิ่มต้นทุนในการพัฒนาซอฟต์แวร์ที่มีภาระงานด้านการทดสอบซอฟต์แวร์เพิ่ม โดยงานวิจัยนี้จัดทำเพื่อใช้ในกระบวนการทดสอบซอฟต์แวร์ ซึ่งเริ่มพัฒนาระบบโดยหาความสัมพันธ์ระหว่างวงจรการทดสอบซอฟต์แวร์ และมาตรฐานการทดสอบซอฟต์แวร์ทั้ง ISO 29119 และ IEEE 829 ที่เป็นมาตรฐานด้านเอกสารการทดสอบ เพื่อศึกษาความเกี่ยวข้อง สอดคล้อง ระหว่างวงจรการทดสอบและเอกสารการทดสอบ แล้วนำมาสร้างฟังก์ชันในระบบวางแผนการทดสอบซอฟต์แวร์ให้สามารถปรับใช้ได้กับซอฟต์แวร์ทุกประเภท พร้อมทั้งวัดประสิทธิภาพทั้งในด้านข้อกำหนดเชิงบรรทัดฐานและประสิทธิภาพในการทดสอบซอฟต์แวร์ เพื่อควบคุม ตรวจสอบ และปรับปรุงคุณภาพของซอฟต์แวร์ โดยระบบช่วยลดต้นทุนและความยุ่งยากในการทดสอบซอฟต์แวร์ ฟังก์ชันภายในงานวิจัยประกอบด้วย test plan, test procedure specification, test data requirement, test environment requirement, test status, test result, measurement software performance และ documentation ซึ่งเอกสารการทดสอบที่ผู้ใช้งานจะได้รับมีความถูกต้อง ครบคลุม ครบถ้วนทุกขั้นตอนในการทดสอบ และมีเนื้อหาในเอกสารการทดสอบที่เป็นไปตามมาตรฐานสากล

คำสำคัญ : การทดสอบซอฟต์แวร์ การพัฒนาซอฟต์แวร์ ประสิทธิภาพของซอฟต์แวร์ มาตรฐานสากล ในการทดสอบซอฟต์แวร์ เอกสารการทดสอบ

Thesis Title	Development of Software Testing System Based on ISO 29119
Student Name	Chadarn Raksawat
Student ID	60605076
Degree	Master of Science (Computer Science)
Department	Computer Science
Year	2019
Thesis Advisor	Dr. Pattama Charoenporn

Abstract

In software development process there should be software testing to inspection and improve the software to be correctness, completely, good quality and able to identify, finding or fix software errors. Software testing to be efficient, correctness and reliability that are necessary to refer to the software testing standard. By studying and understanding the testing standards to completely and thoroughly which is complicated, time-consuming, and increases the cost of software development with additional software testing workloads. In this research, develops for use in the software testing process. That developed by finding a relationship between software testing life cycle and software testing standards in ISO 29119 and IEEE 829 that are software test documentation standard. To study the related between the testing life cycle and the test documentation and then use to create functions in the software testing planning system that can be used in all types of software and measure the performance of a normative requirement and software testing performance to control, verify and improve software quality. Including the system able to reduce the cost and difficulty of software testing. The user will receive the test documents that are correctness, cover and completeness in every step of the test, and the content of the test document that meets international standards.

Keywords : software testing, software development, software performance, software testing standard, software documentation

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือ และการสนับสนุนอย่างดียิ่งจาก ดร.ปัทมา เจริญพร ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ ได้กรุณาให้คำแนะนำ คำปรึกษา ตรวจสอบ แก้ไขข้อมูลต่าง ๆ และร่างวิทยานิพนธ์ฉบับนี้อ่างละเอียดมาโดยตลอด ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ผู้วิจัยขอขอบพระคุณ ดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร และรองศาสตราจารย์ ดร.จิรพร วีระพันธ์ ซึ่งเป็นคณะกรรมการสอบวิทยานิพนธ์ ผู้ที่ให้คำแนะนำ พร้อมแนวทางการแก้ไขปัญหาต่าง ๆ ในการทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้

ในท้ายที่สุดนี้ ผู้วิจัยผู้จัดทำขอขอบพระคุณอาจารย์ และผู้มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในวิทยานิพนธ์ฉบับนี้ลุล่วงได้

ชฎาธาร รักสวัสดิ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	3
1.3 ขอบเขตของงานวิจัย	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 ขั้นตอนการวิจัย	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 วงจรการพัฒนากระบวนการ (Software Development Life Cycle: SDLC)	5
2.1.1 ขั้นตอนการพัฒนากระบวนการ ประกอบด้วย 7 ขั้นตอน	5
2.2 การทดสอบซอฟต์แวร์ (Software Testing)	6
2.2.1 ขั้นตอนการทดสอบซอฟต์แวร์	7
2.2.2 ประเภทของกระบวนการทดสอบ (Testing Process)	7
2.2.3 รูปแบบในการทดสอบซอฟต์แวร์ (Perform the testing of the system)	8
2.3 มาตรฐานการทดสอบซอฟต์แวร์ ISO 29119	9
2.3.1 ISO/IEC 29119-1, Concepts and Definitions: แนวความคิดและการนิยาม	9
2.3.2 ISO/IEC 29119-2, Test Process: กระบวนการทดสอบ	10
2.3.3 ISO/IEC 29119-3, Test Documentation: เอกสารการทดสอบ	14
2.3.4 ISO/IEC 29119-4, Test Techniques: เทคนิคการทดสอบ	15
2.3.5 ISO/IEC 29119-5, Keyword Driven Testing: เทคนิคการทดสอบ แบบอัตโนมัติ	17
2.4 มาตรฐานเอกสารทดสอบซอฟต์แวร์ (IEEE Standard 829)	17
2.4.1 เอกสารทดสอบซอฟต์แวร์ประกอบด้วย 10 ประเภท	17

สารบัญ (ต่อ)

	หน้า
2.5 สิทธิการใช้งานในระบบ (Participation Organizations)	18
2.6 ตัวชี้วัดการทดสอบซอฟต์แวร์ (Software Testing Metrics)	19
2.7 ฐานข้อมูล (Database)	20
2.7.1 นิยามและคำศัพท์พื้นฐานเกี่ยวกับระบบฐานข้อมูล	21
2.7.2 คำศัพท์ในฐานข้อมูลที่เกี่ยวข้อง	21
2.8 โปรแกรม Visual Basic Studio	26
2.8.1 ส่วนคอนโทรล (Controls)	26
2.8.2 ส่วนโค้ด (Coding)	27
2.8.3 ส่วนฐานข้อมูล (Database)	27
2.8.4 ส่วนรายงาน (Report)	28
2.9 โปรแกรม Microsoft SQL Server Management Studio	28
2.9.1 ส่วนประกอบของ SQL Server	28
2.9.2 ประเภทของข้อมูลที่จัดเก็บใน SQL Server	30
2.10 Crystal Report	31
2.10.1 การทำงานของ Crystal report	31
2.10.2 ส่วนประกอบของ Crystal report	32
2.11 งานวิจัยที่เกี่ยวข้อง	36
บทที่ 3 วิธีการดำเนินงานวิจัย	41
3.1 เครื่องมือที่ใช้ในการวิจัย	43
3.2 ขั้นตอนการพัฒนาาระบบ	43
3.2.1 สิทธิการใช้งานระบบ	44
3.2.2 สร้างความสัมพันธ์ระหว่างขั้นตอนการทดสอบและมาตรฐาน เอกสารการทดสอบ	45
3.2.3 การแบ่งกลุ่มการดำเนินงาน และรายละเอียดของแต่ละฟังก์ชัน	46
3.3 การออกแบบระบบทดสอบซอฟต์แวร์	50
3.3.1 แผนภาพการไหลของข้อมูล (Data Flow Diagram)	51
3.3.2 แผนภาพยูสเคส (Use Case Diagram)	56
3.3.3 ER-Diagram	66
บทที่ 4 ผลการวิจัยและการอภิปรายผล	68

สารบัญ (ต่อ)

	หน้า
4.1 ผลการใช้งานของระบบทดสอบซอฟต์แวร์	68
4.2 ผลแบบสำรวจความพึงพอใจของผู้ใช้งานระบบ	105
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	107
5.1 สรุปผลการดำเนินการ	107
5.2 ข้อเสนอแนะ	107
เอกสารอ้างอิง	109
ภาคผนวก ก	112
ภาคผนวก ข	115
ประวัติผู้เขียน	130

สารบัญตาราง

ตารางที่	หน้า
2.1 สิทธิการใช้งานภายในระบบ	19
3.1 คำอธิบายการกำหนดข้อมูลสมาชิก (Member Information)	58
3.2 คำอธิบายการตรวจสอบสิทธิการใช้งานระบบ (Verify Permission)	58
3.3 คำอธิบายการเข้าสู่ระบบ (Login)	59
3.4 คำอธิบายการตั้งค่าทีม (Setting Team)	59
3.5 คำอธิบายการกำหนดทีมในการทดสอบ (Define Team Member)	60
3.6 คำอธิบายการสร้างข้อมูลองค์กร (Define Organization Information)	60
3.7 คำอธิบายการตั้งค่าลำดับความสำคัญของความต้องการ (Setting Requirement Priority)	61
3.8 คำอธิบายการสร้างโปรเจกการทดสอบ (Define Test Project)	61
3.9 คำอธิบายการสร้างแผนการทดสอบ (Define Test Plan)	62
3.10 คำอธิบายการกำหนดการทดสอบ (Define Test Specification)	63
3.11 คำอธิบายการดำเนินการทดสอบ (Test Execution)	63
3.12 คำอธิบายการกำหนดสถานะการทดสอบ (Approval)	64
3.13 คำอธิบายการกำหนดสถานะการทดสอบ (Project Status)	64
3.14 คำอธิบายการวัดประสิทธิภาพของการทดสอบ (Performance Measurement)	65
3.15 คำอธิบายการสร้างรายงาน (Report)	65
4.1 แบบสำรวจความพึงพอใจของผู้ใช้งานระบบ	105

สารบัญรูป

รูปที่	หน้า
2.1 แผนภาพกระบวนการในการพัฒนาซอฟต์แวร์	5
2.2 ขั้นตอนการทดสอบซอฟต์แวร์	7
2.3 รูปแบบการทดสอบซอฟต์แวร์	8
2.4 มาตรฐานในการทดสอบซอฟต์แวร์ (ISO 29119)	9
2.5 กระบวนการจัดการทดสอบ	11
2.6 กระบวนการทดสอบขององค์กร	11
2.7 กระบวนการสร้างข้อกำหนดการทดสอบขององค์กร	12
2.8 กระบวนการสร้างแผนการทดสอบ (Test Plan)	13
2.9 การวัดประสิทธิภาพการทดสอบซอฟต์แวร์ (Measure Software Test)	13
2.10 กระบวนการทดสอบแบบไดนามิกส่วนของกรณีทดสอบ (Test Case)	14
2.11 กระบวนการทดสอบแบบไดนามิกส่วนของขั้นตอนทดสอบ (Test Procedure)	14
2.12 กระบวนการทำงานภายใน IEEE 829	18
2.13 ประเภทของเอนทิตี	22
2.14 ตัวอย่างการใช้เอนทิตีเชิงความสัมพันธ์	22
2.15 ตัวอย่างแอตทริบิวต์	22
2.16 ประเภทของแอตทริบิวต์	23
2.17 ตัวอย่างแอตทริบิวต์แบบธรรมดา	23
2.18 ตัวอย่างคอมโพสิตแอตทริบิวต์	24
2.19 ตัวอย่างแอตทริบิวต์ที่มีหลายค่า	24
2.20 ตัวอย่างดีไรฟ์แอตทริบิวต์	24
2.21 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง	25
2.22 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม	25
2.23 ตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม	25
2.24 โปรแกรม Visual Basic Studio	26
2.25 หน้าออกแบบ (Design)	27
2.26 ส่วนโค้ด (Coding)	27
2.27 ฐานข้อมูล (Database) ใน visual basic studio	28
2.28 ส่วนหน้ารายงานโดยใช้ Crystal Report	28
2.29 โปรแกรม SQL Server	29

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.30 การเชื่อมต่อฐานข้อมูลกับเซิร์ฟเวอร์	29
2.31 การทำงานของ crystal report	32
2.32 พื้นที่การทำงานภายใน crystal report	32
2.33 ส่วนประกอบของ crystal report	33
2.34 Database Fields ใน crystal report	33
2.35 รูปแบบของ Formula Fields	34
2.36 รูปแบบ Group Name Fields	34
2.37 รูปแบบ Special Fields	34
2.38 รูปแบบ Unbound Fields	35
2.39 การจัดกลุ่มข้อมูลเพื่อทำรายงาน	35
2.40 รายงานที่แสดงข้อมูลในรูปแบบกราฟ	36
2.41 การสร้างรายงานส่วนย่อย	36
2.42 โปรแกรมสร้างกรณีทดสอบ (Modern Test Case Management Software)	40
3.1 แผนภาพการทำงานของระบบ	41
3.2 ภาพรวมขั้นตอนการดำเนินงานวิจัย	44
3.3 แผนภาพความสัมพันธ์ระหว่างขั้นตอนการทดสอบ มาตรฐานการทดสอบ และมาตรฐานเอกสารการทดสอบ	45
3.4 แผนภาพแสดงการแบ่งกลุ่มตามมาตรฐาน ISO 29119	47
3.5 แผนภาพการไหลของข้อมูล Context Diagram	51
3.6 แผนภาพการไหลของข้อมูล DFD Level 1	52
3.7 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการกำหนดข้อมูลองค์กร	54
3.8 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการโปรเจกต์ทดสอบ	54
3.9 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการข้อกำหนดการทดสอบ	55
3.10 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการดำเนินการทดสอบ	55
3.11 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการวัดประสิทธิภาพ	56
3.12 แผนภาพยูสเคส	57
3.13 แผนภาพ ER-Diagram	67
4.1 สัมผัสสมาชิก	68
4.2 หน้าจอการใช้งานหลัก	69

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 หน้าจอเมนูการใช้งานของผู้จัดการโปรเจค (Project Manager)	70
4.4 หน้าจอเมนูการใช้งานของทีมทดสอบ (Tester Team)	70
4.5 หน้าจอเมนูการตั้งค่าทีม (Define Team)	71
4.6 หน้าจอเมนูการตั้งค่าลำดับความสำคัญ (Priority)	71
4.7 หน้าจอการตั้งค่าประเภทของความต้องการ (Requirement Type)	72
4.8 กำหนดกลุ่มสมาชิกในทีม	72
4.9 กำหนดหน้าที่สมาชิกในทีม	73
4.10 แสดงประเภทของสมาชิกในทีม	73
4.11 ข้อมูลองค์กร หรือ ลูกค้า	74
4.12 ข้อมูลองค์กร หรือ ลูกค้า	74
4.13 กำหนดนโยบาย และกลยุทธ์ขององค์กร	75
4.14 กำหนดนโยบาย และกลยุทธ์ขององค์กร	75
4.15 หน้าจอหลักของ Project Test Plan	76
4.16 หน้าจอ Test Project ในส่วน Project Type	76
4.17 รายละเอียดขององค์กรที่มีอยู่ในระบบ	77
4.18 โปรเจคทดสอบในส่วนฟังก์ชัน (Function)	77
4.19 โปรเจคทดสอบในส่วนกำหนดความต้องการ (Requirement)	78
4.20 ประเภทของความต้องการ (Requirement Type)	78
4.21 ลำดับความสำคัญของความต้องการ	79
4.22 โปรเจคทดสอบในส่วนความเสี่ยงของผลิตภัณฑ์ (Product Risk)	79
4.23 โปรเจคทดสอบในส่วนความเสี่ยง (Risk)	80
4.24 โปรเจคทดสอบในส่วนนโยบาย (Test Policy)	80
4.25 โปรเจคทดสอบในส่วนกลยุทธ์ (Strategy)	81
4.26 โปรเจคทดสอบในส่วน Schedule	81
4.27 โปรเจคทดสอบในส่วน Staffing	82
4.28 กำหนดฟังก์ชัน	82
4.29 Function และ Requirement ของระบบ	83
4.30 กำหนดลำดับความสำคัญของ Test Item	83
4.31 กำหนดความเสี่ยงของโปรเจค	84

สารบัญญรูป (ต่อ)

รูปที่	หน้า
4.32 กำหนดความเสี่ยงของผลิตภัณฑ์	84
4.33 กำหนดกลยุทธ์การทดสอบ	85
4.34 กำหนดวัน เวลา กิจกรรม ในแผนการทดสอบ	85
4.35 กำหนดบทบาทหน้าที่การทดสอบในแผนการทดสอบ	86
4.36 คุณสมบัติการทดสอบ	86
4.37 ตั้งค่า Feature ที่ต้องการทดสอบ	87
4.38 กำหนดเงื่อนไขการทดสอบ	87
4.39 กำหนดขั้นตอนการทดสอบ	88
4.40 กำหนดกรณีทดสอบ	88
4.41 กำหนดสภาพแวดล้อมสำหรับการทดสอบ	89
4.42 กำหนดขั้นตอนการทดสอบ	89
4.43 รายงานผลการทดสอบ	90
4.44 กรณีสอบรายงานผลการทดสอบ	90
4.45 รายงานเหตุการณ์ทดสอบ	91
4.46 แนบไฟล์เหตุการณ์ทดสอบเพิ่มเติม	91
4.47 รายงานผลการทดสอบ	92
4.48 รายงานผลการทดสอบส่วนกรณีทดสอบ	92
4.49 ผล Normative Requirement	93
4.50 วัดค่าประสิทธิภาพ	94
4.51 วัดค่าประสิทธิภาพในส่วนกรณีทดสอบ	94
4.52 สถานการณ์ดำเนินงานของกรณีทดสอบ	95
4.53 แสดงผลการวัดประสิทธิภาพ	95
4.54 การอนุมัติผลการทดสอบ	96
4.55 เมนูหลักของการรายงานผล	96
4.56 เมนูพิมพ์รายงานขององค์กร	97
4.57 ตัวอย่างรายงานข้อมูลองค์กร พร้อมนโยบายการทดสอบ	98
4.58 ตัวอย่างรายงานข้อมูลองค์กรในส่วนกลยุทธ์การทดสอบ	98
4.59 เมนูพิมพ์รายงานของแผนการทดสอบ	99
4.60 ตัวอย่างรายงานแผนการทดสอบ	99

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.61 เมนูพิมพ์รายงานของกรณีทดสอบ	100
4.62 ตัวอย่างรายงานกรณีทดสอบ	100
4.63 เมนูพิมพ์รายงานของเหตุการณ์การทดสอบ	101
4.64 ตัวอย่างรายงานเหตุการณ์การทดสอบ	101
4.65 เมนูพิมพ์รายงานของการทดสอบที่สมบูรณ์	102
4.66 แบบรายงานการทดสอบที่สมบูรณ์	103
4.67 เมนูการพิมพ์รายงานของสถานการณ์ทดสอบ	103
4.68 ตัวอย่างรายงานประสิทธิภาพการทดสอบ	104
4.69 ตัวอย่างรายงานประสิทธิภาพการทดสอบแบบกราฟ	104

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้เทคโนโลยีมีการพัฒนาอย่างก้าวหน้าและรวดเร็ว เมื่อโลกในปัจจุบันขับเคลื่อนด้วยเทคโนโลยี จึงเรียกว่า ยุค Digital 4.0 หรือ “ดิจิทัล 4.0” หรือ “ดิจิทัลไทยแลนด์” เป็นวลีที่คนไทยเริ่มจะได้ยินบ่อยขึ้นในช่วงหลายปีมานี้ [1] ซึ่งเทคโนโลยีเป็นที่สนใจของคนทุกมุมโลก ทุกเพศ ทุกวัย ทุกสาขาอาชีพ อย่างแพร่หลายและมีการนำมาใช้ในการทำงาน ชีวิตประจำวัน และการเรียน การศึกษา ส่งผลให้ในปัจจุบันนี้มีหลักสูตรที่เกี่ยวกับเทคโนโลยีเพิ่มเข้าไปด้วย เพื่อให้เด็ก ๆ รู้เท่าทันถึงภัยที่มาพร้อมกับโลกยุคดิจิทัล ซึ่งเทคโนโลยีและโลกดิจิทัลเปลี่ยนไปอย่างรวดเร็ว และเคลื่อนที่ไม่มีหยุด องค์กรธุรกิจต่าง ๆ จึงจำเป็นต้องปรับตัวให้ทันตามเทรนด์ พัฒนานวัตกรรมเพื่อต่อยอดธุรกิจบนการแข่งขันที่รวดเร็วและรอบด้านจาก SME (Small and Medium Enterprises) ที่แปลว่า วิชากิจ ขนาดกลางและขนาดย่อมให้กลายเป็น Smart Enterprise ที่มีศักยภาพสูงขึ้น จากบริการธรรมดาให้กลายเป็น High Value Service [2] เพื่อความมั่นคง มั่งคั่ง และยั่งยืนของธุรกิจ [1] โดยหลายธุรกิจมีการนำซอฟต์แวร์มาปรับใช้ในการทำงานมากขึ้นเพื่อเพิ่มประสิทธิภาพการทำงาน ลดต้นทุน ลดระยะเวลา ควบคุมคุณภาพของงาน เพิ่มโอกาสทางการตลาดและโอกาสในการทำงานด้วยตลาดของซอฟต์แวร์มีขนาดใหญ่ที่เกิดจากการแข่งขันที่เพิ่มขึ้นและมีจำนวนผู้ใช้งานจำนวนมาก ส่งผลให้ในปัจจุบันนี้มีซอฟต์แวร์ออกสู่ท้องตลาดเพิ่มขึ้นมากในรูปแบบที่หลากหลายเพื่อให้ตอบสนองต่อกลุ่มธุรกิจที่มีหลายประเภท โดยส่วนใหญ่ที่ออกสู่ท้องตลาดจะอยู่ในรูปแบบซอฟต์แวร์ประยุกต์ที่ประกอบด้วยซอฟต์แวร์สำเร็จและซอฟต์แวร์ใช้งานเฉพาะ ซึ่งซอฟต์แวร์สำเร็จ (software package) เป็นซอฟต์แวร์ที่มีความนิยมใช้กันสูงมาก ดังนั้นบริษัทที่จะผลิตซอฟต์แวร์ขายจะต้องมีคุณสมบัติที่เป็นจุดเด่นเฉพาะตัว เพื่อดึงดูดลูกค้าและต้องสร้างความน่าเชื่อถือให้แก่ลูกค้า สร้างความมั่นใจให้แก่ลูกค้าเมื่อใช้ซอฟต์แวร์ของทางบริษัทว่าจะไม่ส่งผลให้ธุรกิจของลูกค้าเกิดความเสียหายเป็นอันขาด ดังนั้นการทดสอบซอฟต์แวร์ก่อนที่จะทำการส่งมอบให้ลูกค้าถือเป็นสิ่งที่สำคัญ หากขาดการทดสอบซอฟต์แวร์ก่อนส่งมอบให้ลูกค้าทำให้เกิด ค่าใช้จ่ายในการแก้ไขซอฟต์แวร์สูง ซอฟต์แวร์มีคุณภาพต่ำ ส่งผลต่อซอฟต์แวร์ทั้งระบบทำให้เกิด failure ระบบไม่สามารถใช้งานได้ ส่งผลให้ลูกค้าขาดความเชื่อมั่นในซอฟต์แวร์ของบริษัท

การทดสอบซอฟต์แวร์เป็นกระบวนการในการตรวจสอบความสอดคล้องของแอปพลิเคชันหรือโปรแกรมของซอฟต์แวร์ และระบุหรือค้นหาความผิดพลาด (error) ของซอฟต์แวร์ พร้อมทั้งสามารถระบุถึงแนวทางการเกิดปัญหาและวิธีในการแก้ไขปัญหาที่เกิดขึ้น เพื่อให้สามารถทำงานได้ถูกต้องตรงตามข้อกำหนดของลูกค้าในเรื่องคุณภาพของซอฟต์แวร์ การทดสอบซอฟต์แวร์จะต้อง

ทำการศึกษาค้นคว้าหาข้อมูลก่อนที่จะทดสอบซอฟต์แวร์ทั้งขั้นตอนในการทดสอบซอฟต์แวร์ ความเหมาะสมที่จะนำมาปรับใช้กับซอฟต์แวร์ที่ได้พัฒนาขึ้น หรือแผนการทดสอบซอฟต์แวร์ที่บริษัทมีอยู่แล้ว ด้วยซอฟต์แวร์มีหลายประเภทและหลายขนาด โดยที่ยังมีคุณลักษณะความต้องการเฉพาะตัว การที่จะหาข้อสรุปมาว่าว่าการทดสอบที่ได้ทำอยู่นั้นดีหรือไม่นั้นทำได้ยาก อีกทั้งการทดสอบซอฟต์แวร์เพียงอย่างเดียวนั้นไม่สามารถที่จะกำจัดข้อบกพร่องในซอฟต์แวร์ได้อย่างสมบูรณ์ จำเป็นต้องมีการประเมินผลประสิทธิภาพของซอฟต์แวร์เพิ่มเติมเพื่อให้ทราบถึงการเปลี่ยนแปลงของซอฟต์แวร์หลังจากการทดสอบ ด้วยปัญหาดังกล่าวการที่นำมาตราฐานสากลในการทดสอบซอฟต์แวร์ จึงเป็นวิธีการแก้ปัญหาที่ครอบคลุมและครบถ้วนที่สุด ทั้งในด้านการแก้ไขหรือป้องกันข้อผิดพลาด ข้อบกพร่องที่เกิดขึ้นในกระบวนการทดสอบ อีกทั้งยังควบคุมการทดสอบซอฟต์แวร์ที่ครอบคลุม ครบถ้วนตามขั้นตอนการทดสอบ สามารถนำไปใช้งานได้จริง [3] อีกทั้งยังสามารถสร้างความน่าเชื่อถือให้แก่ผู้ใช้งานระบบ ดังนั้นการทดสอบตามมาตรฐานสากลจึงเป็นสิ่งที่ควรทำ [4] แต่ด้วยการนำมาตราฐานการทดสอบมาใช้ในการทดสอบซอฟต์แวร์เป็นสิ่งที่ยุ่งยาก เสียเวลา และยิ่งเพิ่มต้นทุนของโปรเจกต์อีกด้วย ทำให้บริษัทโดยทั่วไปไม่ได้เห็นถึงความจำเป็น

ทางผู้วิจัยเห็นถึงความจำเป็นพร้อมทั้งหาแนวทางในการแก้ไขปัญหาดังต้น จึงได้พัฒนาระบบที่ช่วยในการทดสอบซอฟต์แวร์ที่เข้ามาช่วยในด้านการจัดเตรียมเอกสารทดสอบซอฟต์แวร์ ซึ่งระบบสามารถสร้างเอกสารการทดสอบที่ครอบคลุมทั้งวงจรการทดสอบซอฟต์แวร์ที่เป็นไปตามมาตรฐานสากลและสามารถนำไปใช้งานได้จริง ซึ่งขั้นตอนในการดำเนินงานของระบบถูกอ้างอิงตามขั้นตอนการทดสอบซอฟต์แวร์ที่มาตรฐาน ISO 29119 ส่วนที่ 2 ซึ่งเป็นกระบวนการทดสอบ (test process) และ มาตรฐาน ISO 29119 ส่วนที่ 3 และ IEEE 829 เป็นมาตรฐานด้านเอกสารทดสอบซอฟต์แวร์ พร้อมทั้งยังใช้วงจรการทดสอบซอฟต์แวร์ (software testing life cycle) โดยทางผู้วิจัยได้อ้างอิงมาตรฐาน ISO 29119 และมาตรฐาน IEEE 829 ที่เป็นมาตรฐานสากลในการทดสอบซอฟต์แวร์ [5] ซึ่งมั่นใจว่าเป็นมาตรฐานที่ได้มีการตรวจสอบความถูกต้องทั้งกระบวนการทำงาน และความเหมาะสมที่สามารถนำมาปรับใช้กับงานในยุคนี้ [6][7] และภายในระบบนี้ได้มีการวัดประสิทธิภาพของซอฟต์แวร์ที่ทำการทดสอบ โดยมีการวัดประสิทธิภาพ 2 ประเภท คือด้านข้อกำหนดเชิงบรรทัดฐานและด้านความสามารถในการทดสอบซอฟต์แวร์ เพื่อให้ทราบถึงคุณภาพ ประสิทธิภาพ และความสมบูรณ์ของซอฟต์แวร์ก่อนที่จะส่งมอบให้ลูกค้าหรือนำไปใช้งานจริง ระบบที่พัฒนาขึ้นมา นั้นสามารถนำไปประยุกต์ใช้กับบริษัทได้อย่างง่าย และสามารถนำไปใช้งานได้จริง พร้อมทั้งความถูกต้องในการใช้งานเนื่องจากระบบถูกพัฒนาขึ้นมาโดยอาศัยทั้งมาตรฐานสากลการทดสอบและวงจรการทดสอบที่เป็นที่รู้จักโดยทั่วไป

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อสร้างเอกสารทดสอบซอฟต์แวร์ที่ถูกต้องตามมาตรฐานการทดสอบและมาตรฐานเอกสารทดสอบ
- 2) เพื่อสร้างรูปแบบเอกสารการทดสอบซอฟต์แวร์เป็นพื้นฐาน
- 3) เพื่อตรวจสอบ และปรับปรุงให้ซอฟต์แวร์มีความถูกต้อง สมบูรณ์ คุณภาพที่ดี
- 4) เพื่อวัดประสิทธิภาพของการทดสอบซอฟต์แวร์
- 5) เพื่อสร้างความเชื่อมั่นและความเป็นมืออาชีพในกลยุทธ์ทางการตลาดของบริษัท
- 6) เพื่อควบคุมระยะเวลาการดำเนินงานให้เป็นไปตามกำหนด

1.3 ขอบเขตงานวิจัย

- 1) สามารถปรับใช้ได้ในองค์กรธุรกิจขนาดเล็กถึงกลางที่มีพนักงานอยู่ระหว่าง 50 – 200 คน
- 2) ความต้องการ (Requirement) ที่ใช้ในการทดสอบเป็นความต้องการที่ได้รับการยืนยันมาจากนักวิเคราะห์ระบบเรียบร้อยแล้ว
- 3) ระบบการทดสอบซอฟต์แวร์จัดทำขึ้นเพื่อให้ทีมทดสอบซอฟต์แวร์ใช้งานเท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เพื่อให้ได้เอกสารการทดสอบซอฟต์แวร์ที่ถูกต้องตามมาตรฐานการทดสอบและมาตรฐานเอกสารทดสอบ
- 2) เพื่อเป็นแนวทางในการทดสอบซอฟต์แวร์ให้กับผู้เริ่มต้นในกระบวนการทดสอบซอฟต์แวร์
- 3) เพื่อควบคุมขั้นตอน และระยะเวลาในการทดสอบซอฟต์แวร์
- 4) เพื่อให้ทราบถึงประสิทธิภาพในการทดสอบซอฟต์แวร์
- 5) สามารถนำรูปแบบเอกสารการทดสอบมาปรับใช้ได้ง่ายขึ้น

1.5 ขั้นตอนของการวิจัย

- 1) ศึกษาค้นคว้า รวบรวมข้อมูล ทฤษฎี และงานวิจัยที่เกี่ยวข้องกับการพัฒนาโปรแกรม
- 2) วิเคราะห์ ศึกษาจุดเด่นของมาตรฐานสากลที่เหมาะสมในการพัฒนาโปรแกรม
- 3) ศึกษาวงจรการทดสอบซอฟต์แวร์
- 4) วิเคราะห์ ศึกษาแนวทางในการทดสอบซอฟต์แวร์
- 5) หาความสัมพันธ์ของวงจรการทดสอบซอฟต์แวร์ และมาตรฐานการทดสอบซอฟต์แวร์
- 6) พัฒนาระบบสร้างเอกสารการทดสอบซอฟต์แวร์
- 7) พัฒนาระบบสมัครสมาชิกเพื่อใช้งานระบบ เพื่อใช้ในการทดสอบซอฟต์แวร์
- 8) ทดสอบความถูกต้องของโปรแกรมที่ใช้ในสร้างเอกสารประกอบการทดสอบโปรแกรม

- 9) นำระบบไปให้ผู้ที่มิหน้าทีในด้านการทดสอบใช้งานจริง เพื่อหาข้อผิดพลาด และแนวทางแก้ไข
- 10) วิเคราะห์ผลการวิจัย และสรุปผลการวิจัย
- 11) จัดทำเอกสารและนำเสนอรายงานวิทยานิพนธ์ฉบับสมบูรณ์

บทที่ 2

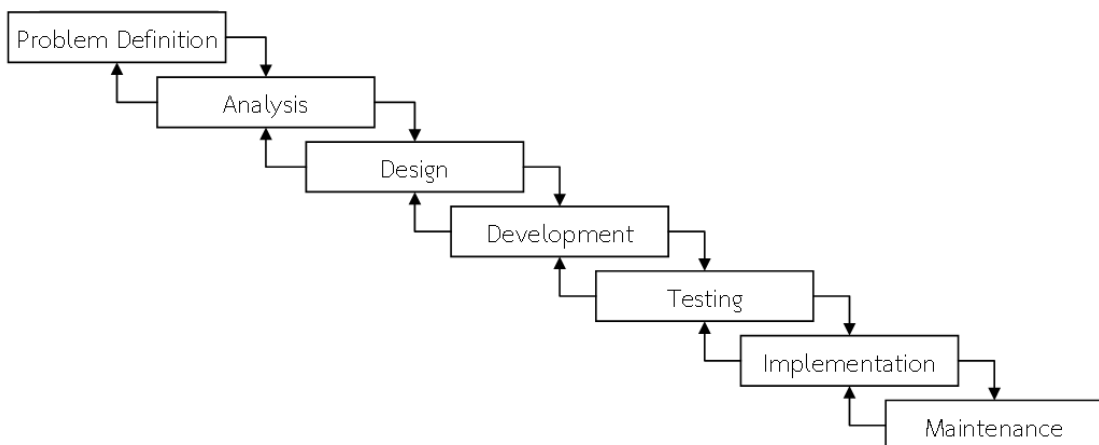
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ส่วนบทที่ 2 กล่าวถึงความรู้และทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119 ซึ่งพบว่าเทคโนโลยี และภาษาทางคอมพิวเตอร์ที่ใช้ในการพัฒนา คือ Visual Basic Studio, ภาษา vb.net, Crystal Report, SQL, Microsoft SQL Server, และส่วนสุดท้ายคืองานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119 โดยผู้จัดทำได้ศึกษาและรวบรวมข้อมูลไว้ดังนี้

2.1 วงจรการพัฒนา ระบบ (Software Development Life Cycle: SDLC)

วงจรการพัฒนา ระบบเป็นกระบวนการที่ใช้ในการพัฒนาระบบงานเพื่อให้เป็นไปในทิศทางเดียวกัน ลดข้อบกพร่องให้น้อยที่สุด และเพื่อแก้ปัญหาทางธุรกิจและตอบสนองต่อความต้องการของผู้ใช้งาน ซึ่งช่วยให้นักวิเคราะห์ระบบสามารถดำเนินการได้อย่างมีแนวทางและเป็นขั้นตอนพร้อมทั้งสามารถควบคุมระยะเวลาและงบประมาณในการพัฒนาระบบงาน [8]

2.1.1 ขั้นตอนการพัฒนา ระบบ ประกอบด้วย 7 ขั้นตอน ดังนี้



รูปที่ 2.1 แผนภาพกระบวนการในการพัฒนาซอฟต์แวร์

1. การกำหนดปัญหา (Problem Definition) เป็นขั้นตอนแรกของขั้นตอนในการพัฒนา เนื่องด้วยจะต้องทราบปัญหาที่แท้จริงก่อน เพื่อหาแนวทางในการแก้ไขปัญหาพร้อมทั้งศึกษาความเป็นไปได้ของระบบที่จะพัฒนาขึ้นมา จากนั้นทำการจัดเตรียมทรัพยากรทั้งมนุษย์และอุปกรณ์ และกำหนดระยะเวลาในการดำเนินงาน

2. การวิเคราะห์ (Analysis) เป็นขั้นตอนในการศึกษาและวิเคราะห์ถึงขั้นตอนการดำเนินงานของระบบที่ต้องรวบรวมและวิเคราะห์ความต้องการจากผู้ใช้งานระบบเพื่อนำมาสร้างระบบใหม่ จากนั้นสร้างแบบจำลองขึ้นมา เช่น Data Flow Diagram: DFD, Context Diagram

3. การออกแบบ (Design) เป็นขั้นตอนที่นำผลลัพธ์จากการวิเคราะห์มาพัฒนาให้เป็นแบบจำลองเชิงกายภาพโดยงานออกแบบสถาปัตยกรรมระบบที่เกี่ยวข้องทั้งฮาร์ดแวร์ ซอฟต์แวร์ และระบบเครือข่าย พร้อมด้วยออกแบบหน้าจอทั้งข้อมูลรับเข้า (Input data) การออกแบบแผนงานระบบ การออกแบบฐานข้อมูล สร้างโปรแกรมต้นแบบ (Prototype) และการออกแบบโปรแกรม

4. การพัฒนา (Development) เป็นขั้นตอนในการพัฒนาโปรแกรม ซึ่งจะพัฒนาตามแผนการดำเนินงานที่ได้ระบุไว้ในส่วนการออกแบบทั้งข้อมูลการออกแบบต่าง ๆ การเลือกภาษาในการพัฒนาโปรแกรมที่เหมาะสม เลือกเครื่องมือ (Tools) ที่นำมาช่วยพัฒนาโปรแกรม ซึ่งการพัฒนาโปรแกรมจะถูกดำเนินการโดยทีมโปรแกรมเมอร์ที่ต้องเขียนชุดคำสั่งเพื่อสร้างเป็นระบบงานขึ้นมา

5. การทดสอบ (Testing) เมื่อดำเนินการในขั้นตอนการพัฒนาเสร็จสิ้นแล้ว โปรแกรมที่ได้มานั้นจะยังไม่สามารถนำไปใช้งานได้ (Deploy) ต้องทำการทดสอบเพื่อหาข้อผิดพลาดก่อนทั้งการทดสอบทางไวยากรณ์ภาษาทางคอมพิวเตอร์ ตรวจสอบผลลัพธ์ที่ได้ ตรวจสอบว่าโปรแกรมที่พัฒนามานั้นตรงต่อความต้องการของลูกค้าหรือไม่ เป็นต้น

6. การติดตั้ง (Implementation) เมื่อทดสอบโปรแกรมที่ได้พัฒนาเรียบร้อยแล้ว จึงนำไปติดตั้ง ณ สภาพแวดล้อมจริงของลูกค้า ซึ่งก่อนนำไปติดตั้งจำเป็นต้องทำการศึกษาถึงสภาพแวดล้อมของพื้นที่จริงก่อนเพื่อหาแนวทางป้องกันและแก้ไข จัดทำคู่มือระบบ ฝึกอบรมผู้ใช้งาน และประเมินผลการใช้งานของระบบใหม่

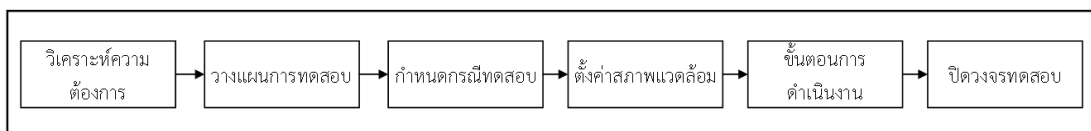
7. การบำรุงรักษา (Maintenance) เป็นขั้นตอนสุดท้ายของวงจรการพัฒนาระบบที่จะต้องมันตรวจสอบ ดูแลให้อยู่ในสภาพที่พร้อมใช้งานตลอดเวลา นอกจากนั้นยังต้องทำการแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นจากระบบ อาจต้องมีการพัฒนาบางส่วนของโปรแกรมเพิ่ม และ หากผู้ใช้มีความต้องการเพิ่มเติม ทางทีมพัฒนาต้องมีการเตรียมแผนรองรับสถานการณ์ต่าง ๆ ที่อาจเกิดขึ้นได้เสมอ

2.2 การทดสอบซอฟต์แวร์ (Software Testing)

การทดสอบซอฟต์แวร์ คือ กระบวนการตรวจสอบซอฟต์แวร์ก่อนที่จะส่งมอบหรือนำไปใช้งานจริง เป็นการตรวจสอบหาความจริงโดยการลงมือใช้ซอฟต์แวร์นั้นซึ่งจำเป็นต้องอาศัยเทคนิคในการตรวจสอบ โดยการทดสอบจะต้องทำตามวัตถุประสงค์ สมมติฐาน ข้อกำหนด ขั้นตอนวิธีที่ได้ระบุเพื่อหาข้อผิดพลาด (error) จุดบกพร่อง ความเสี่ยงของซอฟต์แวร์นั้น ๆ พร้อมทั้งปรับแก้ปัญหานั้นก่อนที่จะส่งมอบงาน เพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามีความถูกต้อง ความสมบูรณ์ ปลอดภัย และคุณภาพที่ดี [9]

2.2.1 ขั้นตอนการทดสอบซอฟต์แวร์

ขั้นตอนการทดสอบซอฟต์แวร์นั้นไม่มีรูปแบบที่ตายตัวส่วนใหญ่แล้วขึ้นอยู่กับระเบียบวิธีการภายในของแต่ละองค์กร ซึ่งแต่ละองค์กรอาจมีขั้นตอนการทดสอบที่แตกต่างกัน ทางผู้วิจัยได้เลือกใช้ขั้นตอนการทดสอบ [10][11] ดังรูปที่ 2.2



รูปที่ 2.2 ขั้นตอนการทดสอบซอฟต์แวร์

1. **วิเคราะห์ความต้องการ** ทีมทดสอบซอฟต์แวร์ต้องทำการศึกษาซอฟต์แวร์ก่อน เพื่อให้ทราบถึงความต้องการของลูกค้าอย่างแท้จริง มีวัตถุประสงค์เพื่อระบุและวิเคราะห์ทั้งข้อกำหนด และความต้องการที่สามารถทดสอบได้ [12]

2. **วางแผนการทดสอบ** เมื่อทำการวิเคราะห์ความต้องการเรียบร้อยแล้ว ทีมทดสอบซอฟต์แวร์ต้องทำการจัดทำแผนการทดสอบ เอกสารกลยุทธ์การทดสอบ เลือกเครื่องมือทดสอบ วางแผนทรัพยากรและกำหนดบทบาทและความรับผิดชอบ กำหนดการฝึกอบรม เพื่อจะใช้เป็นแนวทางในการกำหนดกรณีทดสอบ (Test Case) ในขั้นต่อไป

3. **กำหนดกรณีทดสอบ (Test Case)** นำข้อมูลที่ได้วางแผนการทดสอบ มาสร้างกรณีทดสอบพร้อมสร้างข้อมูลทดสอบ เช่น ทดสอบสภาพแวดล้อม ความพร้อมใช้งาน เป็นต้น เพื่อใช้ทดสอบซอฟต์แวร์

4. **ตั้งค่าสภาพแวดล้อม** กำหนดเงื่อนไขของซอฟต์แวร์ และ ฮาร์ดแวร์ที่ทำการทดสอบ ซึ่งสามารถทำขั้นตอนการตั้งค่าสภาพแวดล้อมพร้อมกับขั้นตอนการกำหนดกรณีทดสอบได้

5. **ขั้นตอนการดำเนินงาน** ขั้นตอนการทดสอบการทำงาน โดยทีมทดสอบทำการทดสอบตามแผนการทดสอบและกรณีทดสอบที่เตรียมไว้ เมื่อพบข้อผิดพลาดก็รายงานไปยังทีมพัฒนาเพื่อทำการแก้ไข และทีมทดสอบจะทำการทดสอบข้อผิดพลาดที่ได้รับการแก้ไขซ้ำ

6. **ปิดวงจรทดสอบ** ดำเนินการเมื่อทำการทดสอบเสร็จสิ้น ทางทีมทดสอบทำการอธิบายและวิเคราะห์รอบการทดสอบ เพื่อปรับปรุงกลยุทธ์ วิธีที่ใช้ในการทดสอบ พร้อมทั้งเตรียมเอกสารรายละเอียดของโครงการ ทำรายงานสรุปผลการทดสอบ

2.2.2 ประเภทของกระบวนการทดสอบ (Testing Process)

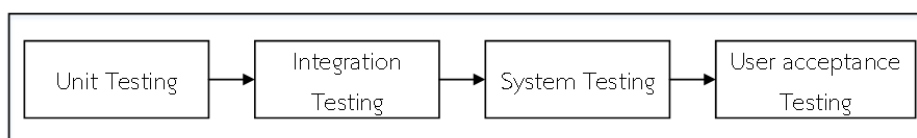
ความต้องการเกี่ยวกับซอฟต์แวร์ของลูกค้า นั้นมีทั้งแบบที่เป็น Functional Requirement และ Non-Functional Requirement ส่งผลให้ในกระบวนการทดสอบจำเป็นต้องแบ่งกระบวนการทดสอบออกเป็น 2 ประเภท [13][14] ดังนี้

1. Functional Testing เป็นการทดสอบตาม Business requirement ที่ได้รับมา เช่น ลูกค้าต้องการให้หน้าสมัครสมาชิกต้องเป็นแค่เขียวเท่านั้นที่สามารถทำการสมัครให้ลูกค้าได้ ซึ่งจะทดสอบโดยให้พนักงานขายมาทำการสมัครสมาชิก ระบบสามารถสมัครได้หรือไม่ ซึ่งในส่วนนี้จะทำการทดสอบแบบ Unit Testing, Integration Testing, Interface and Usability Testing, System Testing, Regression Testing, User Acceptance Testing และ White Box and Black Box Testing

2. Non-functional Testing เป็นการทดสอบเกี่ยวกับความต้องการใช้งานของลูกค้า เช่น ระบบสามารถรองรับลูกค้าจำนวนมากได้หรือไม่ ระยะเวลาในการรันของระบบเป็นเท่าไร ซึ่งในส่วนนี้จะทำการทดสอบในส่วนการทดสอบการโหลด และประสิทธิภาพ (Load and Performance Testing) การทดสอบความปลอดภัยของอุปกรณ์ และสะดวกต่อการใช้งาน (Ergonomics Testing) การทดสอบความกดดันและปริมาณ (Stress & Volume Testing) การทดสอบความเข้ากันได้กับการเคลื่อนย้าย (Compatibility & Migration Testing) การทดสอบการแปลงข้อมูล (Data Conversion Testing) การทดสอบความปลอดภัยและการบุกรุก (Security / Penetration Testing) การทดสอบความพร้อมในการใช้งาน (Operational Readiness Testing) การทดสอบการติดตั้ง (Installation Testing) การทดสอบความปลอดภัยทั้งแอปพลิเคชัน เครือข่าย และระบบ (Security Testing (Application Security, Network Security, System Security))

2.2.3 รูปแบบในการทดสอบซอฟต์แวร์ (Perform the testing of the system)

การพัฒนาซอฟต์แวร์มีความสลับซับซ้อน จึงต้องวางแผนการพัฒนาเป็นขั้นตอน เช่น เริ่มจากพัฒนาเป็นฟังก์ชัน นำฟังก์ชันมารวมกันเป็นโมดูล ซึ่งส่งผลต่อการทดสอบซอฟต์แวร์ที่จำเป็นต้องทดสอบให้ครบถ้วน สมบูรณ์ ดังนั้นการทดสอบต้องแบ่งรูปแบบในการทดสอบซอฟต์แวร์ออกเป็น ส่วน ๆ เพื่อให้สามารถทดสอบได้ครบถ้วน ซึ่งสามารถแบ่งได้ 4 ประเภท ดังรูปที่ 2.3



รูปที่ 2.3 รูปแบบการทดสอบซอฟต์แวร์

1. ทดสอบหน่วยย่อย (Unit testing) เป็นการทดสอบทีละหน่วยย่อย เพื่อเป็นการยืนยันการทำงานระดับย่อยที่สุดว่าทำงานได้ถูกต้อง เป็นการทดสอบโดยผู้เขียนโค้ด

2. ทดสอบการบูรณาการ (Integration testing) เป็นการทดสอบเพื่อตรวจสอบการทำงานร่วมกันของส่วนย่อย ๆ (Component หรือ Module) ที่นำมาประกอบกันเพื่อให้ได้ซอฟต์แวร์ที่สมบูรณ์

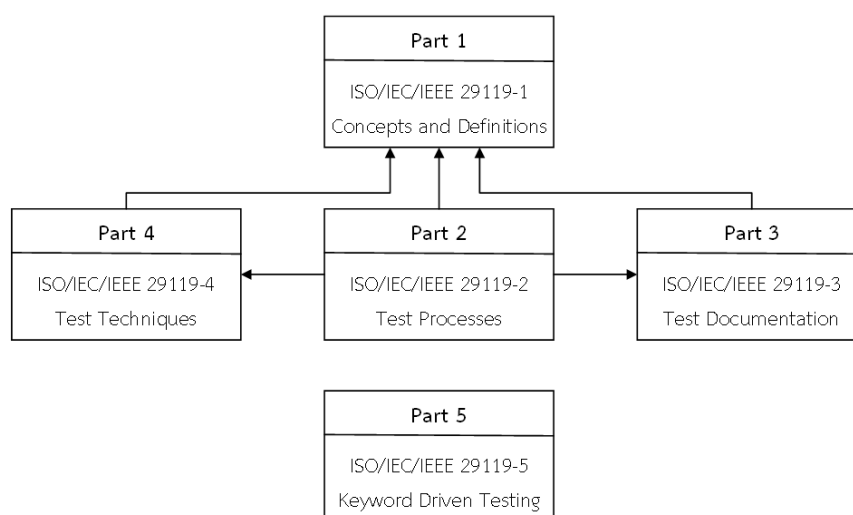
3. ทดสอบระบบ (System testing) เป็นการทดสอบเพื่อตรวจสอบความถูกต้องของการทำงานระบบเมื่อทำงานร่วมกับหน่วยย่อยอื่น ๆ

4. ทดสอบการยอมรับของผู้ใช้งาน (User acceptance testing: UAT) เป็นการทดสอบขั้นสิ้นสุด โดยผู้ใช้งาน (End user) หรือลูกค้า เพื่อตรวจสอบซอฟต์แวร์ว่าสามารถทำงานได้ตามข้อกำหนดและผู้ใช้งานพึงพอใจหรือไม่

2.3 มาตรฐานการทดสอบซอฟต์แวร์ ISO 29119

ISO / IEC / IEEE 29119 คือ การทดสอบซอฟต์แวร์ที่เป็นชุดมาตรฐานที่มีการตกลงกันในระดับสากลเพื่อไว้ใช้เป็นแนวทางในการทดสอบซอฟต์แวร์ที่สามารถนำมาใช้ได้ภายในวงจรชีวิตการพัฒนาซอฟต์แวร์หรือองค์กรใด ๆ ซึ่งมาตรฐานชุดนี้เป็นมาตรฐานที่ได้รับการยอมรับในระดับสากล ซึ่งจะช่วยให้องค์กรที่นำมาใช้นี้ไปใช้จะทำให้องค์กรของคุณมีวิธีการทดสอบที่มีคุณภาพสูง และสามารถนำไปสื่อสารได้ทั่วโลก [15][16]

มาตรฐานสำหรับการทดสอบซอฟต์แวร์สามารถแบ่งออกได้เป็น 5 ส่วน ดังรูปที่ 2.4



รูปที่ 2.4 มาตรฐานในการทดสอบซอฟต์แวร์ (ISO 29119)

2.3.1 ISO/IEC 29119-1, Concepts and Definitions: แนวความคิดและการนิยาม

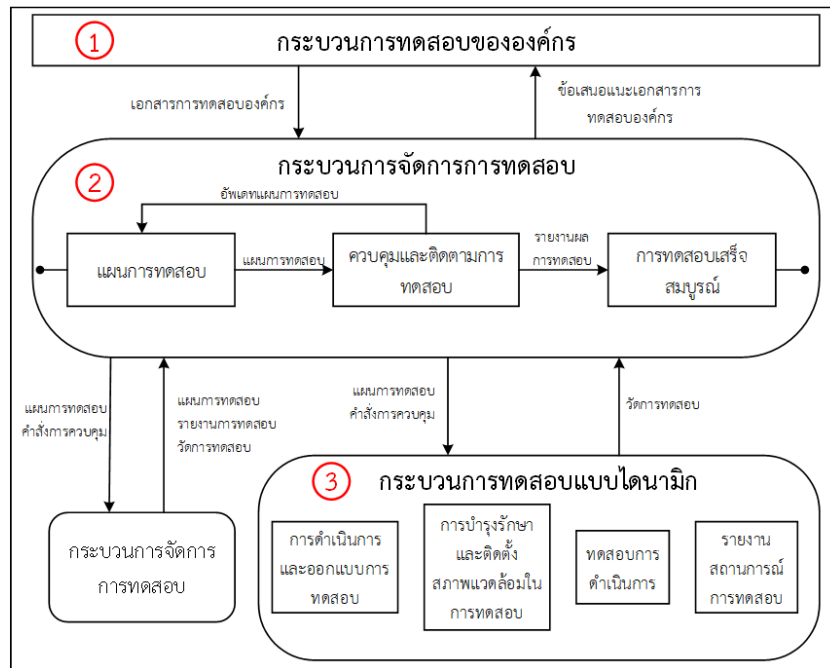
เป็นส่วนที่อำนวยความสะดวกในการใช้ส่วนอื่น ๆ ของมาตรฐานโดยการแนะนำคำศัพท์ที่ใช้สร้างมาตรฐานส่วนอื่น ๆ และมีการยกตัวอย่างของแอปพลิเคชันในแต่ละแนวความคิด

มาตรฐานส่วนที่ 1 จะให้ข้อมูล และให้คำนิยาม คำอธิบายของแนวความคิดในการทดสอบซอฟต์แวร์ และ แนะนำวิธีที่จะประยุกต์ใช้กระบวนการ เอกสาร และเทคนิค ที่ได้กำหนดไว้ในมาตรฐาน 29119 ซึ่งมาตรฐานส่วนที่ 1 ประกอบด้วยหัวข้อต่อไปนี้

1. การทดสอบซอฟต์แวร์เบื้องต้น (Introduction to Software Testing) ประกอบด้วย บทบาทของการทดสอบที่ต้องตรวจสอบความถูกต้องตามความต้องการของลูกค้า (Verification) และ ตรวจสอบความสามารถใช้งานได้ของระบบ (Validation) การทดสอบอย่างละเอียด และการทดสอบแบบฮิวริสติกส์
2. การทดสอบซอฟต์แวร์ในรูปแบบขององค์กร และโปรเจกต์ (Software Testing in an Organizational and Project Context) ที่มุ่งไปในส่วนกระบวนการทดสอบ
3. กระบวนการทดสอบโดยทั่วไปในวงจรการพัฒนาซอฟต์แวร์ (Generic Testing Processes in the Software Life Cycle) ประกอบด้วย การพัฒนากระบวนการย่อยของโครงการ และผลลัพธ์ของโครงการ การบำรุงรักษาอย่างต่อเนื่องและดูแลผล และกระบวนการที่สนับสนุนวงจรการพัฒนาซอฟต์แวร์
4. การทดสอบตามความเสี่ยง (Risk-based Testing) ซึ่งจะนำไปใช้ในกระบวนการทดสอบขององค์กร กระบวนการจัดการทดสอบ และกระบวนการทดสอบแบบไดนามิก
5. การทดสอบกระบวนการย่อย (Test Sub-process) ประกอบด้วย การทดสอบวัตถุประสงค์ รายการทดสอบ (Test Item) การทดสอบลักษณะคุณภาพ การทดสอบพื้นฐาน การทดสอบซ้ำและการถดถอย และเทคนิคการออกแบบทดสอบ
6. การทดสอบการปฏิบัติ (Test Practices) ประกอบด้วย บทนำ การทดสอบตามความต้องการ การทดสอบตามรูปแบบ การทดสอบทางคณิตศาสตร์ การทดสอบตามประสบการณ์ และการทดสอบตามสคริปต์และไม่ได้ตามสคริปต์
7. การทดสอบแบบอัตโนมัติ (Automation in Testing)
8. การกำจัดข้อบกพร่อง (Defect Management)

2.3.2 ISO/IEC 29119-2, Test Process: กระบวนการทดสอบ

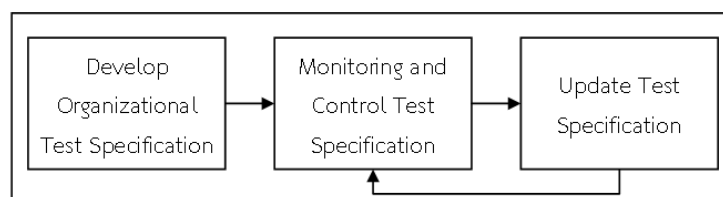
เป็นส่วนที่เกี่ยวข้องกับกระบวนการในการทดสอบ ที่มีการกำหนดแบบจำลองสำหรับ กระบวนการทดสอบซอฟต์แวร์ที่สามารถปรับใช้ได้กับทุกซอฟต์แวร์ที่พัฒนาตามวงจรการพัฒนาซอฟต์แวร์ แบบจำลองที่ระบุกระบวนการทดสอบสามารถนำไปใช้ในการควบคุมการจัดการและการดำเนินการทดสอบซอฟต์แวร์ในทุก ๆ องค์กรหรือกิจกรรมการทดสอบอื่น ๆ กระบวนการทดสอบประกอบด้วย 3 ส่วน แสดงดังรูปที่ 2.5



รูปที่ 2.5 กระบวนการจัดการทดสอบ

1. กระบวนการทดสอบขององค์กร (Organizational Test Process) เป็นส่วนที่เกี่ยวข้องกับนโยบาย และกลยุทธ์ขององค์กร เพื่อบอกถึงคุณลักษณะขององค์กร พร้อมด้วยวิธีแก้ปัญหาของเหตุการณ์ที่จะเกิดขึ้นที่ถูกกำหนดจากองค์กร

ขั้นตอนการดำเนินการของกระบวนการทดสอบขององค์กร เริ่มด้วยการสร้างข้อกำหนดการทดสอบ ติดตามและควบคุมข้อกำหนดการทดสอบ และปรับปรุงข้อกำหนดการทดสอบ ดังรูปที่ 2.6



รูปที่ 2.6 กระบวนการทดสอบขององค์กร

โดยภายในระบบที่พัฒนาขึ้นมานั้น มีการสร้างข้อกำหนดการทดสอบ ขององค์กรซึ่งภายในระบบประกอบด้วย ข้อมูลเบื้องต้นขององค์กร หรือ ลูกค้า ข้อกำหนดด้านนโยบาย และข้อกำหนดด้านกลยุทธ์ขององค์กร รูปแบบการสร้างข้อกำหนดขององค์กรแสดงดังรูปที่ 2.7

```

SettingOrganization.vb* x
TestPlanTemplateDocument (SettingOrganization Events) Load
29 Private Sub SettingOrganization_Load(sender As Object, e As EventArgs) Handles MyBase.Load
30     Form
31     *Connect DataBase
32     con.ConnectionString = "Data Source=CHADATARN\SQLEXPRESS;Initial Catalog=TestPlan;Integrated Security=True"
33     If con.State = ConnectionState.Open Then
34         con.Close()
35     End If
36     con.Open()
37     Random ID
38     OrgId.Text = ran.Next(10001, 99999)
39     *Form Test Specification
40     *Random ID Text
41     OrgId.Text = ran.Next(10001, 99999)
42     ProjectId.Text = OrgId.Text
43     TestSpecification.Text = OrgId.Text
44 End Sub
45
46 Private Sub ToolStripMenuItem2_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem2.Click
47     *Save Btn
48     cmd = con.CreateCommand()
49     cmd.CommandType = CommandType.Text
50     cmd.CommandText = "Insert into SettingOrg ([OrgId],[OrgNo],[OrgType],[ResponBy],[Date],[CreatedBy],
51     [OrgName],[SubOrgName],[CEO],[Coordinator],[Technique],[Note],
52     [Address],[Tel],[Mobile],[Fax],[Email],[Email2]) VALUES
53     ('" & OrgId.Text & "','" & OrgNo.Text & "','" & OrgType.Text & "','" & Respon.Text & "','" & CreateDate.Text & "','" & CreateBy.Text & "','"
54     "' & OrgName.Text & "','" & SubOrgName.Text & "','" & CEO.Text & "','" & Coordinator.Text & "','" & Technique.Text & "','" & Note.Text & "','"
55     "' & Address.Text & "','" & Tel.Text & "','" & Mobile.Text & "','" & Fax.Text & "','" & Mail1.Text & "','" & Mail2.Text & "','" &
56     cmd.ExecuteNonQuery()
57 End Sub
58
59 Private Sub ToolStripMenuItem8_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem8.Click
60     *Save at Test Specification
61     cmd = con.CreateCommand()
62     cmd.CommandType = CommandType.Text
63     cmd.CommandText = "Insert into Project_Org_Policy ([OrgId],[ProjectId],[ProjectName],[TestSpecId],[Version],[Date],[PName],[GroupName],
64     [Object],[Process],[Organization],[Evaluation],[Standards],[Approach],[Policy],
65     [StrategyRisk],[StrategySelection],[StrategyDocument],[StrategyAutomate],[StrategyConfigure],[StrategyIncident],[StrategySubProcess]) VALUES
66     ('" & OrgId.Text & "','" & ProjectId.Text & "','" & ProjectName.Text & "','" & TestSpecification.Text & "','" & Version.Text & "','" & DateTime.Value & "','" & PName
67     "' & Obj.Text & "','" & Process.Text & "','" & Organization.Text & "','" & Evaluate.Text & "','" & Standard.Text & "','" & Approach.Text & "','" & Policy.Text & "','"
68     "' & Risk.Text & "','" & Selection.Text & "','" & Document.Text & "','" & Automate.Text & "','" & Configure.Text & "','" & Incident.Text & "','" & TestSub.Text & "','" &
69     cmd.ExecuteNonQuery()
70 End Sub
71 End Class

```

รูปที่ 2.7 กระบวนการสร้างข้อกำหนดการทดสอบขององค์กร

2. กระบวนการจัดการทดสอบ (Test Management Process) เป็นส่วนที่เกี่ยวข้องกับกระบวนการเตรียมการทดสอบ ควบคุมและติดตามการทดสอบ พร้อมระบุสถานะการทดสอบเมื่อเสร็จสิ้นในแต่ละรอบ

1) กระบวนการเตรียมการทดสอบ (Test Planning Process) เป็นการวางแผนการทดสอบที่ประกอบด้วยการระบุความเสี่ยง วิธีการแก้ปัญหาคือความเสี่ยง การออกแบบกลยุทธ์ในการทดสอบ การกำหนดระยะเวลาและกำหนดสมาชิกในทีมทดสอบ สร้างเอกสารของแผนการทดสอบ เมื่อมีการยืนยันแผนการทดสอบแล้วก็ทำการนำแผนการทดสอบไปดำเนินการทดสอบ โดยภายในระบบที่พัฒนาขึ้นนั้น มีการสร้างแผนการทดสอบดังรูปที่ 2.8

2) กระบวนการติดตาม และควบคุมการทดสอบ (Test Monitoring and Control Processes) เป็นขั้นตอนในการนำแผนการทดสอบมาทำการตั้งค่าข้อมูลตามที่ได้ระบุไว้ มีขั้นตอนการดำเนินงานจากการติดตั้ง ทดสอบการวัด การติดตาม พร้อมทั้งควบคุมและรายงานผลการทดสอบภายในระบบมีการสร้างเอกสารที่ใช้ในการคำนวณประสิทธิภาพ โดยภายในระบบมีการพัฒนาส่วนการวัดประสิทธิภาพการทดสอบซอฟต์แวร์ มีรูปแบบในการวัดประสิทธิภาพดังรูปที่ 2.9

```

TestPlan.vb*
TestPlanTemplateDocument
Button1 Click
140 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
141     'Add
142     cmd = con.CreateCommand()
143     cmd.CommandType = CommandType.Text
144
145     cmd.CommandText = "insert into ItemTest ([TestItemCode],[ModuleName],[FunctionName],
146     [RequirementNo],[Requirement],[Priority]) VALUES
147     ('" & TestItemCode.Text & "','" & ModuleNameItem.Text & "','" & TestFunctionName.Text & "','"
148     & ComboRequireNo.Text & "','" & TestItemRequire.Text & "','" & PriorityBrowse.Text & "'"")
149
150     cmd.ExecuteNonQuery()
151
152     GridView_ItemTestView()
153     MessageBox.Show("Save Successful")
154 End Sub
155
156 Public Sub GridView_ItemTestView()
157     'Require DataView
158     cmd = con.CreateCommand()
159     cmd.CommandType = CommandType.Text
160     cmd.CommandText = "Select * from ItemTest "
161
162     cmd.ExecuteNonQuery()
163     Dim dt As New DataTable()
164     Dim da As New SqlDataAdapter(cmd)
165     da.Fill(dt)
166     ItemTestView.DataSource = dt
167 End Sub
168

```

รูปที่ 2.8 กระบวนการสร้างแผนการทดสอบ (Test Plan)

```

TestStatus.vb*
TestPlanTemplateDocument
TSSearch Click
10 Private Sub TestStatus_Load(sender As Object, e As EventArgs) Handles MyBase.Load
11     'Test Status page
12     con.ConnectionString = "Data Source=CHADATARN\SQLEXPRESS;Initial Catalog=TestPlan;Integrated Security=True"
13     If con.State = ConnectionState.Open Then
14         con.Close()
15     End If
16     con.Open()
17     'End Connection
18 End Sub
19
20
21 Private Sub AddFeature_Click(sender As Object, e As EventArgs) Handles TSSearch.Click
22     'Search data in database for measure test performance
23     cmd = con.CreateCommand()
24     cmd.CommandType = CommandType.Text
25     cmd.CommandText = "SELECT * FROM TestSpecTestCase WHERE TestPlanCode = '" & TSProjCode.Text & "'"
26     cmd.ExecuteNonQuery()
27
28     MessageBox.Show("Save Successful")
29 End Sub
30 End Class

```

รูปที่ 2.9 การวัดประสิทธิภาพการทดสอบซอฟต์แวร์ (Measure Software Test)

3) กระบวนการทดสอบแบบไดนามิก (Dynamic Test Process) เป็นกระบวนการที่เกี่ยวข้องกับการดำเนินการทดสอบที่ทำงานควบคู่กับกระบวนการจัดการทดสอบ ซึ่งขั้นตอนการดำเนินการเริ่มโดยออกแบบการทดสอบและดำเนินการทดสอบ ตั้งค่าสภาพแวดล้อมที่ทดสอบ ดำเนินการทดสอบ และสรุปผลการทดสอบทั้งการทดสอบที่สมบูรณ์ และการทดสอบที่ไม่สมบูรณ์ พร้อมทั้งรายงานเหตุการณ์ที่ทำให้การทดสอบไม่เสร็จสมบูรณ์ โดยภายในระบบมีการสร้างกรณีทดสอบแบบไดนามิกส่วนของกรณีทดสอบ (Test Case) แสดงดังรูปที่ 2.10 และกรณีทดสอบแบบไดนามิกส่วนของขั้นตอนทดสอบ (Test Procedure) แสดงดังรูปที่ 2.11

```

TestSpecification.vb # X
TestPlanTemplateDocument - TestSpecification - GridView_ProcedureView
174
175 Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
176     'Add TestCase
177     cmd = con.CreateCommand()
178     cmd.CommandType = CommandType.Text
179
180     cmd.CommandText = "insert into TestSpecTestCase ([TestPlanCode],[TestCaseCode],[TestCaseName],
181     [Priority],[TestConditionCode],[Purpose],[PreConditions],[Input],[ExpectedResult],[Picture]) V
182     ('" & TestPlanCode.Text & "','" & TestProcedureCode.Text & "','" & TestProcedureCode.Text & "','" &
183     '" & TCPriority.Text & "','" & TCConditionCode.Text & "','" & Purpose.Text & "','" & Preco
184
185     cmd.ExecuteNonQuery()
186
187     GridView_TestCaseView()
188     MessageBox.Show("Save Successful")
189 End Sub
190
191 Public Sub GridView_TestCaseView()
192     'TestCase DataView
193     cmd = con.CreateCommand()
194     cmd.CommandType = CommandType.Text
195     cmd.CommandText = "Select * from TestSpecTestCase "
196
197     cmd.ExecuteNonQuery()
198     Dim dt As New DataTable()
199     Dim da As New SqlDataAdapter(cmd)
200     da.Fill(dt)
201     TestCaseView.DataSource = dt
202 End Sub

```

รูปที่ 2.10 กระบวนการทดสอบแบบไดนามิกส่วนของกรณีทดสอบ (Test Case)

```

TestSpecification.vb # X
TestPlanTemplateDocument - TestSpecification - GridView_ProcedureView
203
204 Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
205     'Test Procedure Head
206     cmd = con.CreateCommand()
207     cmd.CommandType = CommandType.Text
208
209     cmd.CommandText = "insert into TestSpecProcedure ([TestPlanNo],[TestProcedureNo],[ObjectandPriority],[Estimated],
210     [Startup],[RelatewithOtherProcedure],[Date],[Initial],[TestItem],[OkNot],[Comment]) VALUES
211     ('" & TestPlanCode.Text & "','" & TestProcedureCode.Text & "','" & ObjPriority.Text & "','" &
212     '" & StartUp.Text & "','" & Relationship.Text & "','" & DateProcedure.Value & "','" & Init
213
214     cmd.ExecuteNonQuery()
215
216     MessageBox.Show("Save Successful")
217
218     'Test Procedure TestCase
219     cmd = con.CreateCommand()
220     cmd.CommandType = CommandType.Text
221
222     cmd.CommandText = "insert into TestSpecProcedureConcludeTestCase ([TestProcedureNo],[Stepno.TestCase],[Activities]
223     [ExpectedResult],[ActualResult],[TestResult]) VALUES
224     ('" & TestProcedureCode.Text & "','" & StepNo.Text & "','" & ActivitieProce.Text & "','"
225     '" & Expected.Text & "','" & Actual.Text & "','" & Result.Text & "','"
226
227     cmd.ExecuteNonQuery()
228
229     GridView_ProcedureView()
230     MessageBox.Show("Save Successful")
231 End Sub

```

รูปที่ 2.11 กระบวนการทดสอบแบบไดนามิกส่วนของขั้นตอนทดสอบ (Test Procedure)

2.3.3 ISO/IEC 29119-3, Test Documentation: เอกสารการทดสอบ

เป็นต้นแบบสำหรับเอกสารการทดสอบ โดยรูปแบบเอกสารแต่ละประเภทสามารถปรับแต่งให้เหมาะสมตามความต้องการของแต่ละองค์กรได้ ซึ่งเอกสารการทดสอบที่เป็นต้นแบบทั้งหมดนี้เกี่ยวข้องกับกระบวนการทดสอบ (ISO 29119-2) ที่นำทุกขั้นตอนมาสร้างเอกสารการทดสอบ โดยในส่วนมาตรฐานเอกสารการทดสอบประกอบด้วยเอกสาร 3 ส่วน ดังนี้ [7]

1. เอกสารการทดสอบขององค์กร (Organizational Test Process Documentation) ประกอบด้วย เอกสารเกี่ยวกับนโยบายการทดสอบ (Test Policy) และ เอกสารเกี่ยวกับกลยุทธ์การทดสอบ (Organizational Test Strategy)

2. เอกสารกระบวนการจัดการทดสอบ (Test Management Process Documentation) ประกอบด้วย เอกสารเกี่ยวกับแผนการทดสอบ (Test Plan), เอกสารเกี่ยวกับสถานะการทดสอบ (Test Status Report) และ เอกสารเกี่ยวกับการทดสอบที่สมบูรณ์ (Test Completion Report)

3. เอกสารกระบวนการทดสอบแบบไดนามิก (Dynamic Test Process Documentation) ประกอบด้วย เอกสารเกี่ยวกับรายละเอียดการออกแบบการทดสอบ (Test Design Specification), เอกสารเกี่ยวกับรายละเอียดกรณีการทดสอบ (Test Case Specification), เอกสารเกี่ยวกับรายละเอียดขั้นตอนการทดสอบ (Test Procedure Specification), เอกสารเกี่ยวกับความต้องการข้อมูลการทดสอบ (Test Data Requirements), เอกสารเกี่ยวกับวันที่พร้อมทดสอบ (Test Date Readiness Report), เอกสารเกี่ยวกับข้อกำหนดด้านสภาพแวดล้อม (Test Environment Requirements), เอกสารเกี่ยวกับความพร้อมด้านสภาพแวดล้อม (Test Environment Readiness Report), เอกสารเกี่ยวกับผลลัพธ์การทดสอบที่แท้จริง (Actual Results), เอกสารเกี่ยวกับบันทึกการดำเนินการทดสอบ (Test Execution Log) และ เอกสารเกี่ยวกับรายงานเหตุการณ์ทดสอบ (Test Incident Report)

2.3.4 ISO/IEC 29119-4, Test Techniques: เทคนิคการทดสอบ

เป็นการกำหนดเทคนิคในการออกแบบการทดสอบซอฟต์แวร์ เช่น เทคนิคในการออกแบบกรณีทดสอบ และยกตัวอย่างรายละเอียดการใช้งาน ซึ่งเทคนิคการทดสอบสามารถปรับให้เหมาะสมกับความต้องการในแต่ละองค์กรที่ได้นำมาตรฐานมาปรับใช้

เทคนิคการทดสอบ หรือ ISO 29119-4 ครอบคลุมเทคนิคการทดสอบแบบไดนามิกที่นิยมใช้กันทั่วไป ซึ่งภายในมาตรฐานส่วนนี้ประกอบด้วยเทคนิค 3 ประเภท ดังนี้

1. เทคนิคการทดสอบตามข้อมูลจำเพาะ (Specification-Based Testing Techniques) ประกอบด้วย การทดสอบด้วยวิธีต่าง ๆ ดังนี้

- 1) การแบ่งพาร์ทิชันออกเท่า ๆ กัน (Equivalence Partitioning)
- 2) วิธีการจำแนกแบบต้นไม้ (Classification Tree Method)
- 3) การวิเคราะห์ค่าขอบเขต (Boundary Value Analysis)
- 4) การทดสอบการเปลี่ยนสถานะ (State Transition Testing)
- 5) การทดสอบโดยใช้ตารางตัดสินใจ (Decision Table Testing)
- 6) การแสดงเหตุการณ์ที่เกิดขึ้นด้วยกราฟ (Cause-Effect Graphing)
- 7) การทดสอบไวยากรณ์ (Syntax Testing)

8) เทคนิคการทดสอบแบบรวมกัน (Combinatorial Test Techniques) ประกอบด้วย การรวมทั้งหมด (All Combinations) การทดสอบแบบคู่ (Pairwise Testing) การทดสอบแต่ละตัวเลือก (Each Choice Testing) และการทดสอบตัวเลือกพื้นฐาน (Base Choice Testing)

9) แบบแผนการทดสอบ (Scenario Testing) ซึ่งรวมถึงแผนการทดสอบ

10) การทดสอบแบบสุ่ม (Random Testing)

2. เทคนิคการทดสอบตามโครงสร้าง (Structure-Based Testing Techniques) ประกอบด้วยการทดสอบด้วยวิธีต่าง ๆ ดังนี้

1) การทดสอบตามรายงาน (Statement Testing)

2) การทดสอบเป็นก้าน ๆ ตามเงื่อนไข (Branch Testing)

3) การทดสอบการตัดสินใจ (Decision Testing)

4) การทดสอบเงื่อนไข (Condition Testing) ประกอบด้วย เงื่อนไขการทดสอบในแต่ละก้าน ๆ (Branch Condition Testing) การรวมเงื่อนไขการทดสอบแต่ละก้าน (Branch Condition Combination Testing) และ การทดสอบเงื่อนไขการตัดสินใจเงื่อนไขแบบแก้ไข (Modified Condition Decision Condition (MCDC) Testing)

5) การทดสอบการไหลของข้อมูล (Data Flow Testing) ประกอบด้วย คำจำกัดความทั้งหมด การใช้งาน

3. เทคนิคการทดสอบตามประสบการณ์ (Experience-Based Testing Techniques) เป็นการทดสอบด้วยวิธีการคาดเดาข้อผิดพลาด (Error Guessing)

ในมาตรฐานส่วนที่ 4 นี้ยังให้คำนิยามข้อมูลเกี่ยวกับการทดสอบประเภทต่าง ๆ ที่เกี่ยวข้องกับคุณภาพ และการยกตัวอย่างเทคนิคในการออกแบบกรณีทดสอบที่ได้แสดงไว้ด้านบนซึ่งสามารถนำไปปรับใช้ในการทดสอบประเภทต่าง ๆ ประกอบด้วย การทดสอบการเข้าถึง (Accessibility Testing), การทดสอบการสำรองข้อมูล หรือ การกู้คืน (Backup/Recovery Testing), การทดสอบความเข้ากันได้ (Compatibility Testing), การทดสอบการเปลี่ยน (Conversion Testing), การทดสอบการกู้คืนเมื่อเกิดความเสียหาย (Disaster Recovery Testing), การทดสอบการทำงาน (Functional Testing), การทดสอบการติดตั้ง (Install ability Testing), การทดสอบการทำงานร่วมกัน (Interoperability Testing), การทดสอบการปรับตัว (Localization Testing), การทดสอบการบำรุงรักษา (Maintainability Testing), การทดสอบที่เกี่ยวข้องกับประสิทธิภาพ (Performance-Related Testing), การทดสอบการพกพา (Portability Testing), การทดสอบขั้นตอน (Procedure Testing), การทดสอบความน่าเชื่อถือ (Reliability Testing), การทดสอบความปลอดภัย (Security Testing), การทดสอบความเสถียร (Stability Testing) และ การทดสอบการใช้งาน (Usability Testing)

2.3.5 ISO/IEC 29119-5, Keyword Driven Testing: เทคนิคการทดสอบแบบอัตโนมัติ

เป็นการทดสอบด้วยคำสำคัญ ซึ่งคำเหล่านี้เป็นวิธีการอธิบายกรณีทดสอบโดยใช้ชุดคำสำคัญที่กำหนดไว้ คำสำคัญเหล่านี้เป็นชื่อที่เกี่ยวข้องกับการดำเนินการตามขั้นตอนในกรณีทดสอบ ซึ่งใช้คำสำคัญเพื่ออธิบายขั้นตอนการทดสอบแทนภาษาทั่วไป เพื่อให้สามารถทำความเข้าใจกรณีทดสอบและทำให้ดำเนินการอัตโนมัติได้ง่ายขึ้น

มาตรฐานในข้อนี้ครอบคลุมในด้าน คำแนะนำในการทดสอบโดยใช้คำสำคัญ การประยุกต์ใช้คำสำคัญในการทดสอบ กรอบการคำสำคัญที่ใช้ในการทดสอบ การแลกเปลี่ยนข้อมูล ประโยชน์และปัญหาของการทดสอบด้วยคำสำคัญ เริ่มต้นการทดสอบด้วยคำสำคัญ หน้าที่และงานที่เกี่ยวข้องกับการทดสอบ คำสำคัญพื้นฐาน และตัวอย่าง

2.4 มาตรฐานเอกสารทดสอบซอฟต์แวร์ (IEEE Standard 829)

IEEE 829-2008 หรือที่เรียกว่า มาตรฐานสำหรับซอฟต์แวร์และเอกสารประกอบการทดสอบระบบ 829 เป็น มาตรฐาน IEEE ที่ระบุรูปแบบของชุดเอกสารเพื่อใช้ในขั้นตอนที่กำหนดไว้ 8 ขั้นตอนในการทดสอบซอฟต์แวร์และการทดสอบระบบแต่ละขั้นตอนอาจสร้างขึ้นเอง แยกประเภทของเอกสาร มาตรฐานระบุรูปแบบของเอกสารเหล่านี้ แต่ไม่ได้ระบุว่าต้องสร้างขึ้นทั้งหมดหรือไม่ รวมถึงเกณฑ์เกี่ยวกับเนื้อหาที่เพียงพอสำหรับเอกสาร เอกสารสำหรับการทดสอบซอฟต์แวร์เป็นองค์ประกอบที่สำคัญที่เพิ่มกิจกรรมในการทดสอบไปจนถึงระดับของการทดสอบซอฟต์แวร์ ซึ่งองค์กรระหว่างประเทศเช่น IEEE และ ISO ได้เผยแพร่มาตรฐานสำหรับเอกสารการทดสอบซอฟต์แวร์ [17]

2.4.1 เอกสารทดสอบซอฟต์แวร์ประกอบด้วย 5 ประเภท ที่ได้นำมาอ้างอิงและปรับใช้ในงานวิจัย แสดงดังรูปที่ 2.12

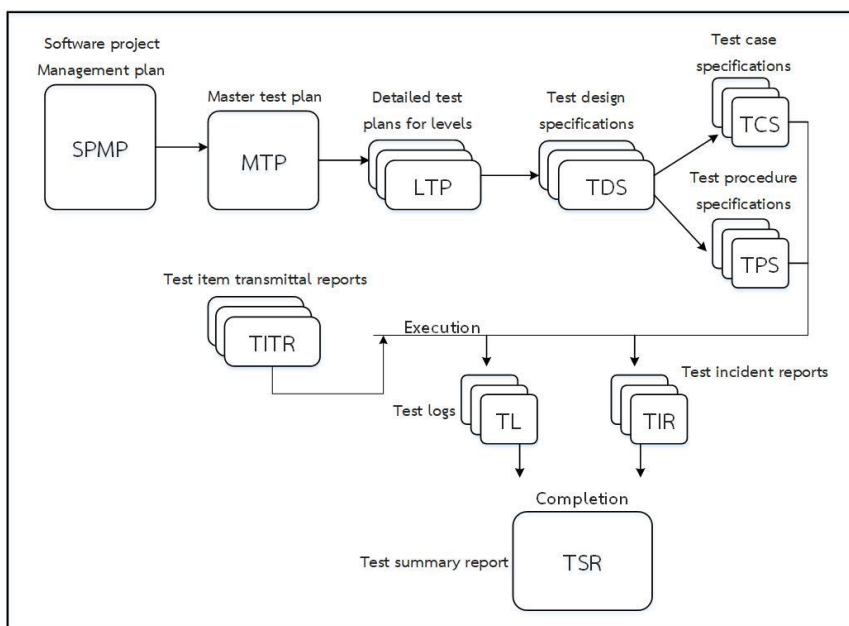
1. บันทึกการทดสอบ หรือ Level Test Log (LTL) เพื่อบันทึกลำดับเหตุการณ์ของรายละเอียดที่เกี่ยวข้องกับการดำเนินการทดสอบ เช่น บันทึกกรณีทดสอบที่ถูกเรียกใช้ เพื่อดูว่าทดสอบผ่านหรือไม่

2. รายงานสิ่งผิดปกติ หรือ Anomaly Report (AR) เพื่อบันทึกเหตุการณ์ใด ๆ ที่เกิดขึ้นระหว่างขั้นตอนการทดสอบที่ต้องการตรวจสอบ ปัญหานี้อาจเรียกได้ว่าเป็น ปัญหา ปัญหาการทดสอบ ข้อบกพร่อง ปัญหาการแก้ไข ความผิดปกติหรือข้อผิดพลาด เอกสารฉบับนี้ถูกตั้งชื่อโดยเจตนาเป็นรายงานความผิดปกติ ไม่ใช่รายงานข้อบกพร่อง รายงานประกอบด้วยรายละเอียดทั้งหมดของเหตุการณ์ เช่นผลที่เกิดขึ้นจริงและที่คาดหวัง เมื่อมันล้มเหลวและหลักฐานสนับสนุนใด ๆ ที่จะช่วยในการแก้ปัญหาของ รายงานนี้ จะรวมถึงการประเมินผลกระทบของเหตุการณ์เมื่อมีการทดสอบที่เป็นไปได้

3. รายงานสถานการณ์ทดสอบระหว่างระดับ หรือ Level Interim Test Status Report (LITSR) เพื่อสรุปผลลัพธ์ระหว่างกิจกรรมการทดสอบที่ระบุและเลือกที่จะให้การประเมินผลและคำแนะนำตามผลการทดสอบสำหรับระดับการทดสอบที่เฉพาะ

4. รายงานการทดสอบ หรือ Level Test Report (LTR) เพื่อสรุปผลของกิจกรรมการทดสอบที่กำหนดไว้และเพื่อให้การประเมินผลและข้อเสนอแนะจากผลการทดสอบหลังจากทดสอบเสร็จสิ้นแล้วสำหรับระดับการทดสอบที่เฉพาะเจาะจง

5. รายงานการทดสอบหลัก หรือ Master Test Report (MTR) เพื่อสรุปผลของกิจกรรมการทดสอบที่กำหนดไว้และเพื่อให้การประเมินเป็นไปตามผลที่วางไว้ รายงานนี้สามารถใช้ในองค์กรใด ๆ ที่ใช้ MTP รายงานการจัดการจะให้ข้อมูลที่สำคัญที่ค้นพบโดยการทดสอบที่ประสบความสำเร็จและรวมถึงการประเมินคุณภาพของความพยายามในการทดสอบคุณภาพของระบบซอฟต์แวร์ภายใต้การทดสอบและสถิติที่ได้จากรายงานความผิดปกติ รายงานนี้จะบันทึกว่าการทดสอบทำอะไรและใช้เวลานานเท่าใดเพื่อปรับปรุงการวางแผนการทดสอบในอนาคต เอกสารขั้นสุดท้ายนี้ใช้เพื่อระบุว่าระบบซอฟต์แวร์ที่ทดสอบอยู่ในเกณฑ์ที่เหมาะสมหรือไม่ ตามเกณฑ์ที่กำหนดโดยผู้มีส่วนได้เสียโครงการ



รูปที่ 2.12 กระบวนการทำงานภายใน IEEE 829

2.5 สิทธิการใช้งานในระบบ (Participation Organizations)

การเข้าใช้งานในระบบนั้นจำเป็นต้องมีการกำหนดสิทธิการใช้งานเพื่อควบคุมและดูแลการเพิ่ม/ลบ เปลี่ยนแปลง ข้อมูล เพื่อป้องกันข้อมูลที่เป็นข้อมูลส่วนบุคคล (Sensitive) และต้อง

คำนึงถึงการรักษาความลับของข้อมูลลูกค้าเพื่อป้องกันข้อมูลรั่วไหล ซึ่งสิทธิการใช้งานสามารถแบ่งได้ตามตารางด้านล่างนี้

ตารางที่ 2.1 สิทธิการใช้งานภายในระบบ

ลำดับ	บทบาท	หน้าที่	รายงานต่อ
1	Planning Manager (PM)	ดูแลและจัดการโครงการ	CEO
2	Team Leader (TL)	ดูแล ควบคุมและการดำเนินงานของโปรเจกต์ในส่วนการควบคุมทีม	PM
3	Quality/Process Manager (QM)	ตรวจสอบคุณภาพของกระบวนการ โปรแกรม	PM, CEO
4	Support Management (SM)	ตรวจสอบมาตรฐาน และเก็บโปรแกรมเข้าสู่โพลเดอร์มาตรฐาน	PM, CEO
5	Development Manager (DM)	เก็บข้อมูลความต้องการของลูกค้า, วิเคราะห์และออกแบบระบบ, เขียนโปรแกรม, ทดสอบโปรแกรม, ติดตั้งโปรแกรม	TL, PM
6	Business Analyst (BA)	เก็บข้อมูลความต้องการของลูกค้า	DM
7	System Analyst (SA)	วิเคราะห์และออกแบบระบบ	DM
8	Developer (DV)	เขียนโปรแกรม	DM
9	Tester (TT)	ทดสอบโปรแกรม	DM
10	System Engineer (SE)	ติดตั้งระบบก่อนนำขึ้นงานขึ้นทดสอบ	DM

2.6 ตัวชี้วัดการทดสอบซอฟต์แวร์ (Software Testing Metrics)

ในกระบวนการพัฒนาซอฟต์แวร์ สิ่งที่สำคัญที่สุดคือการวัดคุณภาพของซอฟต์แวร์และกระบวนการภายในซอฟต์แวร์ หากซอฟต์แวร์ปราศจากการวัดคุณภาพ กล่าวนี้ว่าเป็นซอฟต์แวร์ที่ไม่สมบูรณ์ ซึ่งอาจมีความเสี่ยงเมื่อนำไปใช้งานจริง

ตัวชี้วัดในการทดสอบภายในงานวิจัยนี้ ใช้สำหรับการวัดความถูกต้องระหว่างการพัฒนา และการทดสอบกรณีทดสอบ (Test Case) ซึ่งทำให้สามารถที่จะทำให้ทราบถึงโครงการและสถานะของซอฟต์แวร์ที่ทำการทดสอบ [18]

โดยภายในงานวิจัยประกอบด้วยตัวชี้วัดการทดสอบซอฟต์แวร์ ดังนี้

1. Test Case Executed ใช้สำหรับการตรวจสอบสถานะของกรณีทดสอบ (Test Case) ที่ได้นำไปทดสอบ โดยคำนวณจากข้อมูลของกรณีทดสอบที่ได้ดำเนินการทดสอบ และ จำนวนกรณีทดสอบที่ถูกสร้างทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Executed} = (\text{No. of Test cases executed} / \text{Total no. of Test cases written}) * 100$$

2. Test Case Not Executed ใช้สำหรับการตรวจสอบสถานะของกรณีทดสอบ (Test Case) ที่รอดำเนินการทดสอบ โดยคำนวณจากข้อมูลของกรณีทดสอบที่ยังไม่ได้ดำเนินการทดสอบ และ จำนวนกรณีทดสอบที่ถูกสร้างทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases not executed} = (\text{No. of Test cases not executed} / \text{Total no. of Test cases written}) * 100$$

3. Test Cases Passed ใช้สำหรับการคำนวณกรณีทดสอบ (Test Case) ที่ดำเนินการทดสอบผ่าน โดยคำนวณจากข้อมูลของจำนวนกรณีทดสอบที่ทดสอบผ่าน และ จำนวนกรณีทดสอบที่ดำเนินการทดสอบทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Passed} = (\text{No. of Test cases Passed} / \text{Total no. of Test cases Executed}) * 100$$

4. Test Cases Failed ใช้สำหรับการคำนวณกรณีทดสอบ (Test Case) ที่ดำเนินการทดสอบไม่ผ่าน โดยคำนวณจากข้อมูลของจำนวนกรณีทดสอบที่ทดสอบไม่ผ่าน และ จำนวนของกรณีทดสอบที่ดำเนินการทดสอบทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Failed} = (\text{No. of Test cases Failed} / \text{Total no. of Test cases Executed}) * 10$$

5. Test Cases Blocked ใช้สำหรับการคำนวณกรณีทดสอบ (Test Case) ที่ถูกบล็อก ไม่ให้ทดสอบ โดยคำนวณจากข้อมูลกรณีทดสอบที่ถูกบล็อก และ จำนวนของกรณีทดสอบที่ดำเนินการทดสอบทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Blocked} = (\text{No. of Test cases Blocked} / \text{Total no. of Test cases Executed}) * 100$$

2.7 ฐานข้อมูล (Database)

Database หรือ ฐานข้อมูล คือส่วนที่ใช้ในการจัดเก็บข้อมูลอย่างเป็นระบบ ซึ่งข้อมูลจะถูกเก็บในรูปแบบของ rows, columns และ tables โดยอาศัยดัชนี หรือ indexed เพื่อให้สามารถเข้าถึงและค้นหาข้อมูลได้อย่างสะดวกรวดเร็ว และไม่ให้เกิดความซ้ำซ้อนของข้อมูล ซึ่งสามารถทำการแก้ไขข้อมูลในฐานข้อมูลได้โดยการเพิ่ม (insert), ลด (delete) หรือเปลี่ยนแปลง (update) การจัดเก็บข้อมูลในฐานข้อมูลนั้นสามารถหลีกเลี่ยงความขัดแย้งของข้อมูล ข้อมูลในระบบมีความ

ถูกต้องเชื่อถือได้ และมีรูปแบบเป็นมาตรฐานเดียวกัน โดยมีการกำหนดระบบความปลอดภัยของข้อมูล

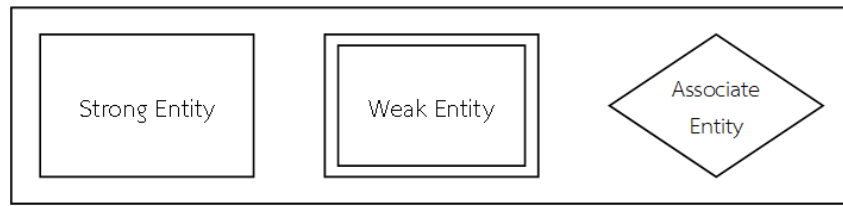
ในปัจจุบันนี้เป็นยุคสารสนเทศที่มีข้อมูลจำนวนมาก ซึ่งจำเป็นต้องได้รับการกลั่นกรองอย่างเหมาะสม เพื่อให้เป็นสารสนเทศที่สามารถนำไปใช้ประโยชน์อย่างมากมาย ทั้งงานด้านธุรกิจ การบริหาร หรือธุรกิจอื่น ๆ ซึ่งในธุรกิจที่มีข้อมูลประมาณมาก ๆ จะพบกับความยุ่งยากในการจัดเก็บข้อมูล การนำข้อมูลออกมาใช้งานได้อย่างถูกต้อง รวดเร็ว ดังนั้นในปัจจุบันนี้จึงมีการใช้ computer database มาเพื่อเป็นเครื่องมือในการช่วยจัดเก็บข้อมูล ประมวลผลข้อมูล ทำให้ระบบการจัดเก็บข้อมูลสะดวก สามารถทำงานได้อย่างเป็นระบบ รวดเร็ว และมีการจัดการความถูกต้อง ข้อมูลที่จัดเก็บใน computer database จะถูกเก็บในรูปแบบไฟล์ (files) เช่น จำนวนสินค้า ยอดขายสินค้า ข้อมูลค้า ข้อมูลสมาชิก หรือ คลังสินค้า

2.7.1 นิยามและคำศัพท์พื้นฐานเกี่ยวกับระบบฐานข้อมูล

1. บิต (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด
2. ไบท์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำบิตมารวมกันเป็นตัวอักขระ (Character)
3. เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลที่ประกอบจากตัวอักขระตั้งแต่หนึ่งตัวขึ้นไปมารวมกัน แล้วได้ความหมายของสิ่งใดสิ่งหนึ่ง เช่น ชื่อ ที่อยู่ ตำแหน่ง เป็นต้น
4. ระเบียบ (Record) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำเขตข้อมูล (field) มารวมกัน เพื่อให้เกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่ง เช่น ข้อมูลของสมาชิก 1 ระเบียบ (1 คน) ประกอบด้วย
 - รหัสสมาชิก 1 เขตข้อมูล (field)
 - ชื่อสมาชิก 1 เขตข้อมูล (field)
 - ที่อยู่ 1 เขตข้อมูล (field)
5. แฟ้มข้อมูล (File) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำข้อมูลหลาย ๆ ระเบียบที่เป็นเรื่องเดียวกันมารวมกัน เช่น แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลพนักงาน

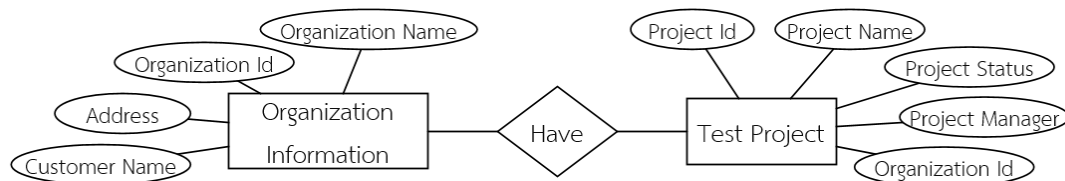
2.7.2 คำศัพท์ในฐานข้อมูลที่เกี่ยวข้องมีดังนี้

1. เอนทิตี (Entity) หมายถึง ชื่อของสิ่งที่ต้องการจัดเก็บข้อมูลไว้ เช่น เอนทิตีลูกค้า เอนทิตีพนักงาน ซึ่งเอนทิตีประกอบด้วย 3 ประเภท ดังรูปที่ 2.13



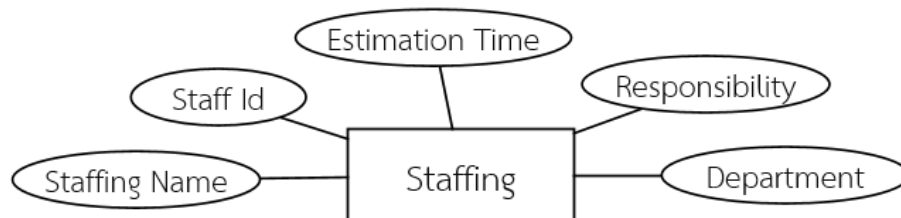
รูปที่ 2.13 ประเภทของเอนทิตี

- 1) เอนทิตีปกติ (Strong Entity หรือ Regular Entity) เป็นเอนทิตีที่อยู่ได้โดยไม่ต้องอาศัยเอนทิตีอื่นในการคงอยู่ ไม่ต้องพึ่งพา หรือขึ้นอยู่กับเอนทิตีอื่น
- 2) เอนทิตีอ่อนแอ (Weak Entity) เป็นเอนทิตีที่อยู่ได้โดยอาศัยเอนทิตีอื่นในการคงอยู่
- 3) เอนทิตีเชิงความสัมพันธ์ (Associate Entity) เป็นเอนทิตีที่ถูกสร้างเพื่อเชื่อมความสัมพันธ์ระหว่างเอนทิตีที่เข้าด้วยกัน ตัวอย่างเช่น ข้อมูลองค์กร (Organization Information) อยู่ในส่วนโปรเจกทดสอบ (Test Project) หมายความว่าข้อมูลองค์กรมีความสัมพันธ์กับโปรเจกทดสอบ แสดงดังรูปที่ 2.14



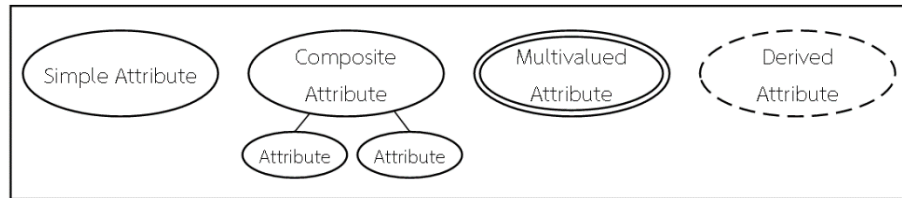
รูปที่ 2.14 ตัวอย่างการใช้เอนทิตีเชิงความสัมพันธ์

2. แอตทริบิวต์ (Attribute) หรือคุณสมบัติ (Property) ซึ่งแอตทริบิวต์จะใช้ในการอธิบายถึงคุณสมบัติหรือคุณลักษณะของเอนทิตี เช่น เอนทิตีของพนักงาน (Staffing) ประกอบด้วยรหัสพนักงาน แผนก ชื่อ-สกุล หน้าที่ เป็นต้น ซึ่งส่วนประกอบเหล่านั้นเรียกว่า แอตทริบิวต์ ซึ่งหนึ่งคุณลักษณะถูกเก็บในหนึ่งแอตทริบิวต์ เช่น แอตทริบิวต์รหัสประจำตัวพนักงาน แอตทริบิวต์ชื่อ แสดงดังรูปที่ 2.15



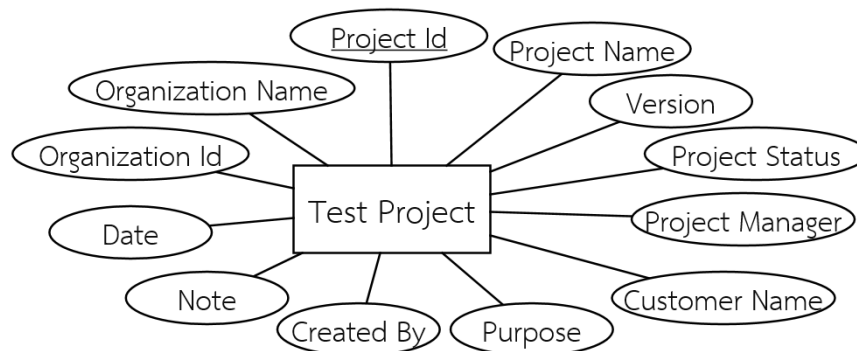
รูปที่ 2.15 ตัวอย่างแอตทริบิวต์

แอตทริบิวต์ที่ใช้อธิบายถึงคุณลักษณะต่าง ๆ สามารถแบ่งออกได้เป็น 4 ประเภทได้แก่ แอตทริบิวต์แบบธรรมดา (Simple Attribute) คอมโพสิตแอตทริบิวต์ (Composite Attribute) แอตทริบิวต์ที่มีหลายค่า (Multivalued Attribute) และ ดีไรฟ์แอตทริบิวต์ (Derived Attribute) แสดงดังรูปที่ 2.16



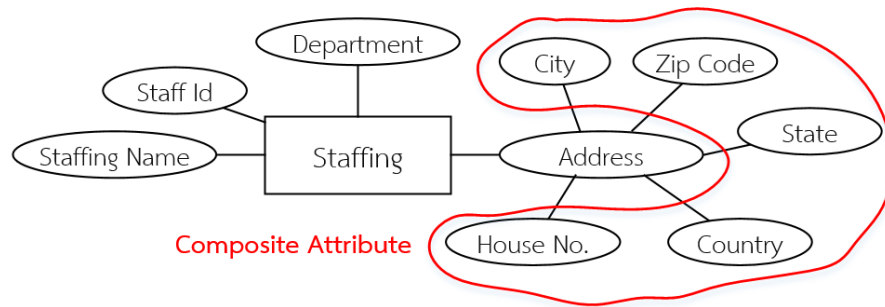
รูปที่ 2.16 ประเภทของแอตทริบิวต์

1) แอตทริบิวต์แบบธรรมดา (Simple Attribute) เป็นแอตทริบิวต์ไม่สามารถแบ่งย่อยได้อีก ซึ่งมีรูปแบบ ชื่อตาราง ที่ประกอบด้วยแอตทริบิวต์หลัก, แอตทริบิวต์ที่ 2 เป็นต้น ตัวอย่าง ตารางชื่อโปรเจกต์ทดสอบ (Test Project) ประกอบด้วย แอตทริบิวต์รหัสโปรเจกต์ (Project Id Attribute) แอตทริบิวต์ชื่อโปรเจกต์ แอตทริบิวต์เวอร์ชัน แอตทริบิวต์สถานะโปรเจกต์ แอตทริบิวต์ผู้จัดการโครงการ แอตทริบิวต์วันที่สร้างโปรเจกต์ ดังรูปที่ 2.17 โดยมีแอตทริบิวต์รหัสโปรเจกต์เป็นคีย์แอตทริบิวต์ (Key Attribute) หรือ คีย์หลัก (Primary Key: PK)



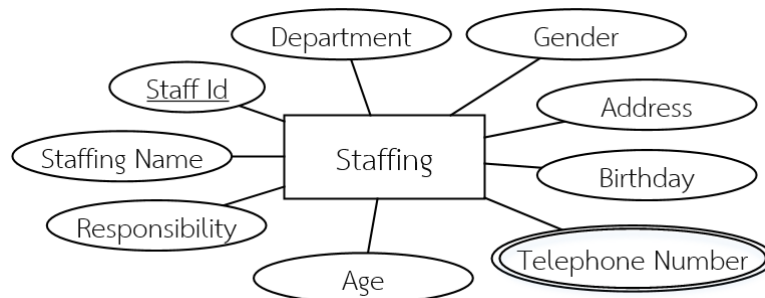
รูปที่ 2.17 ตัวอย่างแอตทริบิวต์แบบธรรมดา

2) คอมโพสิตแอตทริบิวต์ (Composite Attribute) เป็นแอตทริบิวต์แบ่งย่อยซึ่งสามารถแบ่งย่อยคุณลักษณะได้อีก เช่น แอตทริบิวต์ที่อยู่ (Address Attribute) สามารถแบ่งแอตทริบิวต์ย่อยออกได้เป็น City, Zip Code, State, Country และ House No. แสดงตัวอย่างดังรูปที่ 2.18



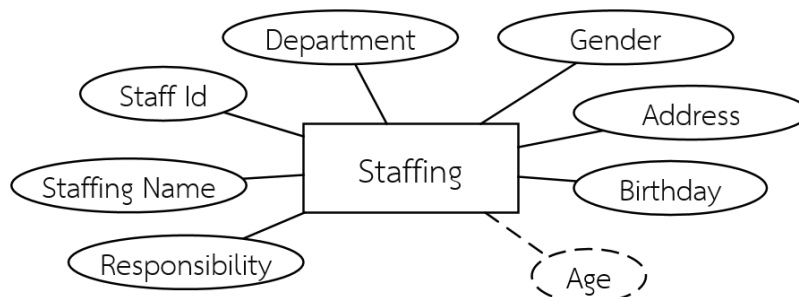
รูปที่ 2.18 ตัวอย่างคอมโพสิตแอตทริบิวต์

3) แอตทริบิวต์ที่มีหลายค่า (Multivalued Attribute) เป็นแอตทริบิวต์ที่สามารถมีคุณลักษณะนั้นได้หลาย เช่น แอตทริบิวต์เบอร์โทรศัพท์ (Telephone Number Attribute) ในบุคคลหนึ่งสามารถมีเบอร์โทรศัพท์ได้หลายเบอร์ สัญลักษณ์ของแอตทริบิวต์ที่มีหลายค่า จะใช้สัญลักษณ์วงรีสองวงซ้อนกัน ดังรูปที่ 2.19



รูปที่ 2.19 ตัวอย่างแอตทริบิวต์ที่มีหลายค่า

4) ดีไรฟ์แอตทริบิวต์ (Derived Attribute) เป็นแอตทริบิวต์ที่ไม่ต้องทำการจัดเก็บ ซึ่งจะเป็นแอตทริบิวต์ที่ได้มาจากการคำนวณแอตทริบิวต์อื่น เช่น แอตทริบิวต์อายุ (Age Attribute) สามารถคำนวณได้จากแอตทริบิวต์วันเดือนปีเกิด ซึ่งสัญลักษณ์ของดีไรฟ์แอตทริบิวต์ จะใช้สัญลักษณ์วงรีที่ใส่เส้นเป็นเส้นประ ดังรูปที่ 2.20



รูปที่ 2.20 ตัวอย่างดีไรฟ์แอตทริบิวต์

3. ความสัมพันธ์ (Relationships) เป็นคำกริยาที่ใช้ในการเชื่อมความสัมพันธ์ระหว่างเอนทิตี (Entity) เข้าด้วยกัน โดยชื่อความสัมพันธ์จะใช้คำกริยาที่แสดงการกระทำ เช่น อยู่ใน มี เรียน เป็นต้น ซึ่งสัญลักษณ์ของความสัมพันธ์จะอยู่ในลักษณะที่เป็นสี่เหลี่ยมข้าวหลามตัด (Diamond) เพื่อใช้แทนการกระทำที่มีคำกริยาของความสัมพันธ์อยู่ภายในสี่เหลี่ยม

การจำแนกประเภทของความสัมพันธ์ตามความสัมพันธ์ระหว่างเอนทิตี เป็นการพิจารณาถึงสัดส่วนความสัมพันธ์ระหว่างสมาชิกในความสัมพันธ์ระหว่างเอนทิตีที่มี 3 ประเภท ดังนี้

1) ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one Relationships) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตีหนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่ง ในลักษณะหนึ่งต่อหนึ่ง (1 : 1) ตัวอย่างเช่น โปรเจกทดสอบ (Test Project) 1 โปรเจกมีการอนุมัติที่สมบูรณ์ได้เพียง 1 ครั้งเท่านั้น ดังรูปที่ 2.21



รูปที่ 2.21 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-many Relationships) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตีหนึ่งที่มีความสัมพันธ์กับข้อมูลหลายๆ ข้อมูลในอีกเอนทิตีหนึ่ง ในลักษณะ (1:m) ตัวอย่างเช่น ภายใน 1 องค์กร (Organization Information) สามารถมีได้หลายโปรเจกทดสอบ (Test Project) ดังรูปที่ 2.22



รูปที่ 2.22 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม

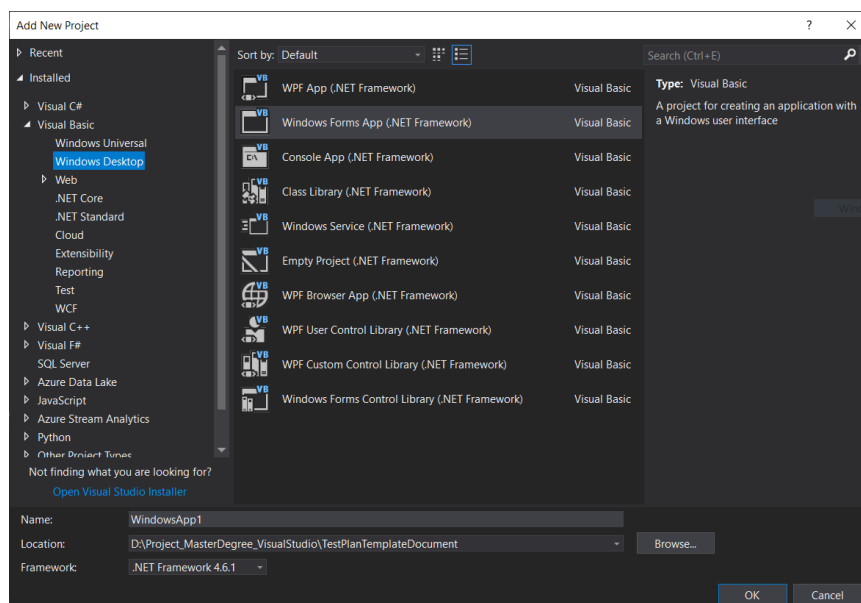
3) ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-many Relationships) เป็นการแสดงความสัมพันธ์ของข้อมูลสองเอนทิตีในลักษณะกลุ่มต่อกลุ่ม (m:n) ตัวอย่างเช่น กรณีทดสอบ (Test Case) 1 กรณี มีขั้นตอนการทดสอบได้หลายขั้นตอน (Test Procedure) ดังรูปที่ 2.23



รูปที่ 2.23 ตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม

2.8 โปรแกรม Visual Basic Studio

Visual Basic Studio เป็นโปรแกรมที่ใช้ในการพัฒนาซอฟต์แวร์ โปรแกรมประยุกต์และระบบต่าง ๆ ซึ่งเป็นโปรแกรมที่รองรับการพัฒนาในรูปแบบแพลตฟอร์มที่หลากหลาย เช่น windows desktop, cross platform, web และสามารถใช้ภาษาที่หลากหลาย เช่น C#, Basic, C++, JavaScript, Python แสดงดังรูปที่ 2.24 ซึ่งภายในงานวิจัยนี้ได้เลือกรูปแบบการพัฒนาโดยใช้ภาษา Visual Basic (VB) และเลือกรูปแบบเป็น windows forms app (.Net Framework)



รูปที่ 2.24 โปรแกรม Visual Basic Studio

จากข้างต้นที่กล่าวมานั้นภายใน Visual Basic Studio (VB) มีฟังก์ชันการทำงานในรูปแบบที่หลากหลาย ซึ่งในการพัฒนาระบบภายในงานวิจัยนี้ได้ใช้ฟังก์ชัน ส่วนคอนโทรล ส่วนโค้ด ส่วนฐานข้อมูล และส่วนรายงาน ในการดำเนินงาน แสดงรายละเอียดดังต่อไปนี้

2.8.1 ส่วนคอนโทรล (Controls) เป็นส่วนที่ช่วยในการออกแบบโปรแกรม ซึ่งเน้นการออกแบบหน้าจอกากราฟิก หรือ Graphic User Interface (GUI) ช่วยในการออกแบบหน้าจอของซอฟต์แวร์ดังรูปที่ 2.25 ซึ่งประกอบด้วยเครื่องมือ หรือ คอนโทรลต่าง ๆ เช่น Form, Textbox, Label เป็นต้น โดยจะเรียกว่าเป็นวัตถุ หรือ ออบเจกต์ (Object) ทุกออบเจกต์จะมีคุณสมบัติ (Properties) และเมธอด (Methods) ประจำตัว โดยจะมีคุณสมบัติที่ต่างกันในแต่ละออบเจกต์

รูปที่ 2.25 หน้าออกแบบ (Design)

2.8.2 ส่วนโค้ด (Coding) เป็นส่วนของชุดคำสั่งที่ใช้ในการควบคุมการทำงานของแต่ละออบเจกต์ดังรูปที่ 2.26 โดยจะแบ่งเป็นส่วน ๆ เรียกว่า โพรซีเจอร์ (procedure) แต่ละโพรซีเจอร์จะประกอบไปด้วย ชุดคำสั่งที่ใช้ในการควบคุมการทำงานของแต่ละออบเจกต์เพื่อตอบสนองการกระทำของผู้ใช้

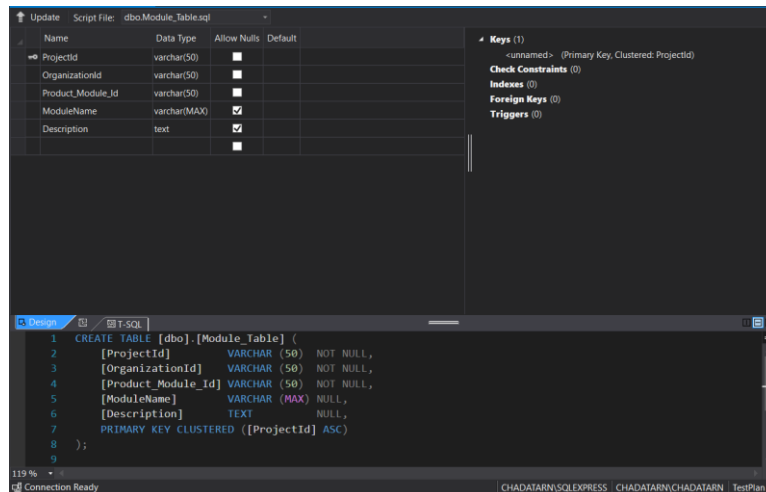
```

1 Imports System.Data.SqlClient
2
3
4 Public Class CreateNew
5
6     Dim con As New SqlConnection
7     Dim cmd As New SqlCommand
8     Dim i As Integer
9
10
11     'For Create Project Test Plan
12     Private Sub ToolStripMenuItem1_Click(sender As Object, e As EventArgs)
13         'Menu
14         Form1.Show()
15         Me.Close()
16     End Sub
17
18     Private Sub ToolStripMenuItem2_Click(sender As Object, e As EventArgs)
19         'Exit
20         Application.Exit()
21     End Sub
22
23     Private Sub Button6_Click(sender As Object, e As EventArgs)
24         TestPlan.Show()
25         Me.Close()
26     End Sub
27
28     Private Sub CreateNew_Load(sender As Object, e As EventArgs) Handles MyBase.Load
29         con.ConnectionString = "Data Source=CHADATARN\SQLEXPRESS;Initial Catalog=TestPlan;Integrated Security=SSPI"
30         If con.State = ConnectionState.Open Then

```

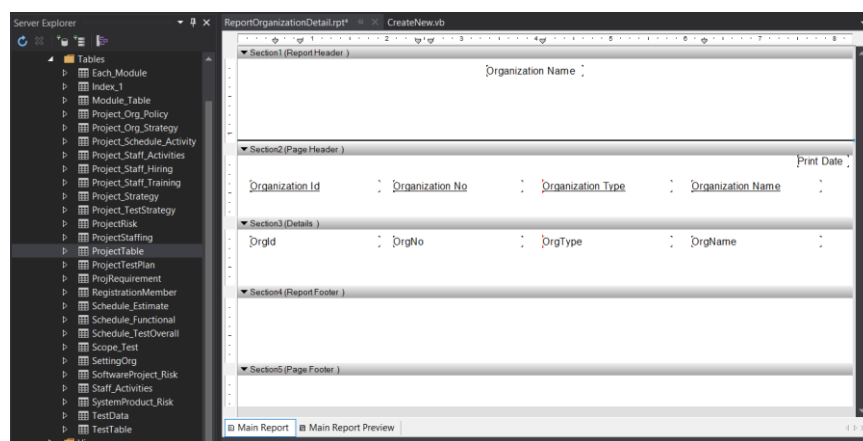
รูปที่ 2.26 ส่วนโค้ด (Coding)

2.8.3 ส่วนฐานข้อมูล (Database) เป็นส่วนของผู้ใช้ในการเก็บข้อมูลที่เชื่อมกับโปรแกรม MS SQL Server Management Studio ซึ่งภายใน Visual Studio สามารถสร้างตาราง (Table) จากในโปรแกรมได้ ดังรูปที่ 2.27 โดยสามารถกำหนดคีย์หลัก (PK) ของแต่ละตาราง (Table) ได้ ทั้งยังสามารถสร้างชุดคำสั่งเพื่อดึงข้อมูลออกมาแสดงได้



รูปที่ 2.27 ฐานข้อมูล (Database) ใน visual basic studio

2.8.4 ส่วนรายงาน (Report) เป็นส่วนที่ใช้ในการสร้างรายงานเพื่อแสดงข้อมูลการทดสอบ ซึ่งทำงานร่วมกับ crystal report โดยอาศัยการดึงข้อมูลมาจากฐานข้อมูลที่ได้สร้างไว้ ดังรูปที่ 2.28



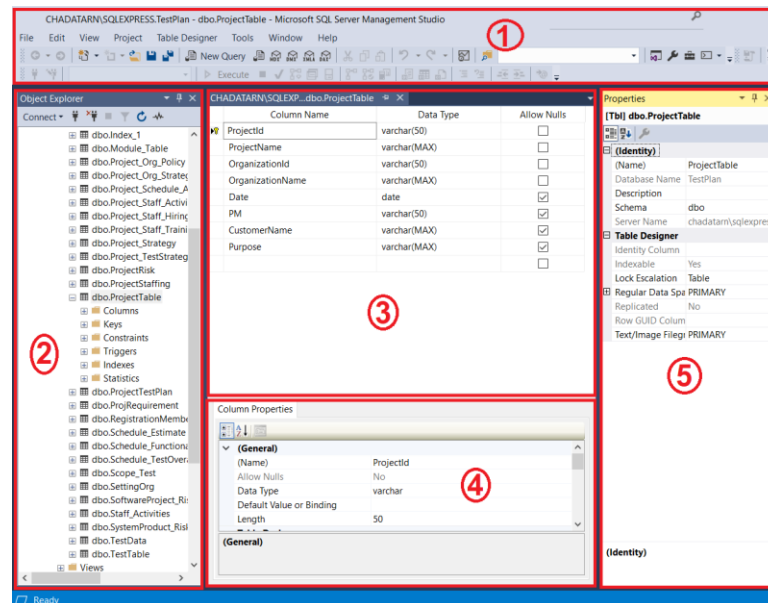
รูปที่ 2.28 ส่วนหน้ารายงานโดยใช้ Crystal Report

2.9 โปรแกรม Microsoft SQL Server Management Studio

SQL Server หรือ Microsoft SQL Server คือระบบฐานข้อมูลที่ใช้ในการสร้าง จัดเก็บ ดึง เรียกใช้ และจัดการข้อมูล ตัวอย่างของ SQL Server ซึ่งระบบ SQL เป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ด้วย SQL Server เป็นฐานข้อมูลที่มีฟังก์ชันการทำงานที่ครอบคลุม ใช้งานง่าย และราคาถูก ทางผู้วิจัยจึงได้เลือกใช้ SQL Server เป็นระบบฐานข้อมูลของงานวิจัยนี้

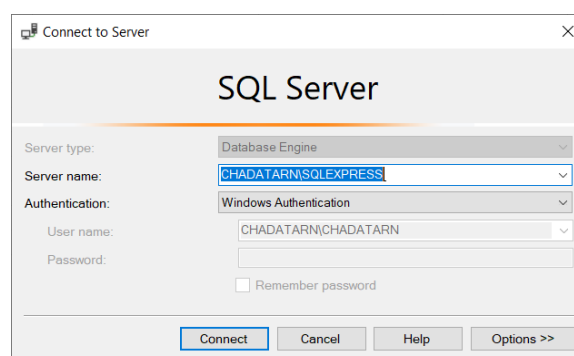
2.9.1 ส่วนประกอบของ SQL Server

ภายใน SQL Server ประกอบด้วยส่วนที่ใช้ในการดำเนินงานทั้ง 5 ส่วน ดังรูปที่ 2.29



รูปที่ 2.29 โปรแกรม SQL Server

1. Menu เป็นส่วนที่แสดงคำสั่งพื้นฐานของโปรแกรมที่ประกอบด้วยเมนู File ซึ่งใช้ในการเชื่อมต่อหรือยกเลิกกับฐานข้อมูล สร้างโปรเจกใหม่ บันทึกข้อมูล เมนู Edit เป็นส่วนที่ใช้ในการแก้ไขข้อมูล ค้นหาหรือแทนค่าข้อมูล เมนู View ใช้ในการดูผลลัพธ์ (output) ดูรายการ error ต่าง ๆ หรือการเรียกใช้เครื่องมือ (toolbox) เป็นต้น เมนู Project ใช้ในการตั้งค่าพื้นฐานแก่โปรเจก เมนู Table Designer ใช้ในการออกแบบข้อมูลซึ่งประกอบไปด้วย การตั้งค่าคีย์หลัก (Primary Key) การเพิ่ม/ลบ คอลัมน์ การเชื่อมความสัมพันธ์ระหว่างตาราง การกำหนด Index หรือ Keys ต่าง ๆ เป็นต้น เมนู Tools ใช้ในการตั้งค่าแก่โปรแกรม เมนู Window ใช้ในการกำหนดลักษณะการแสดงผลของหน้าโปรแกรมที่ใช้งาน และ เมนู Help ส่วนที่ใช้ติดต่อกับนักเทคนิค
2. Object Explorer เป็นส่วนที่ใช้ในการเชื่อมต่อกับเซิร์ฟเวอร์ฐานข้อมูล (Database Server) ดังรูปที่ 2.30 และแสดงฐานข้อมูล (Database) ทั้งหมดที่ได้สร้าง พร้อมทั้งเป็นส่วนที่ใช้ในการควบคุมการดำเนินการเปลี่ยนแปลง แก้ไข ลบ ตารางที่ได้สร้างขึ้นมา



รูปที่ 2.30 การเชื่อมต่อฐานข้อมูลกับเซิร์ฟเวอร์

3. Design เป็นส่วนที่ใช้ในการสร้าง และอัพเดทข้อมูลของแต่ละตาราง (Table) ซึ่งประกอบด้วย Column Name การตั้งชื่อให้คอลัมน์, Data Types ประเภทของข้อมูล และ Allows Nulls การกำหนดค่าข้อมูลที่ว่าง พร้อมทั้งการกำหนดคีย์หลัก (Primary Key) ให้แต่ละตาราง

4. Column Properties เป็นส่วนที่ใช้ในการแทนข้อมูลในแต่ละคอลัมน์ ซึ่งแบ่งออกเป็น 2 ส่วน คือ ส่วน General ประกอบด้วย Name ชื่อของคอลัมน์, Allows Nulls สถานะการกำหนดค่าข้อมูลที่ว่าง, Data Types ประเภทของข้อมูล, Default Value or Binding ค่าเริ่มต้นของข้อมูลหรือการซ่อนข้อมูล และ ส่วน Table Designer

5. Properties ส่วนที่แสดงรายละเอียดของตาราง ซึ่งแบ่งออกเป็น 2 ส่วน คือ ส่วน Identity ประกอบด้วย Name ชื่อของตาราง, Database Name ชื่อฐานข้อมูล, Description คำอธิบาย, Schema โครงสร้างข้อมูล, Server Name ชื่อเซิร์ฟเวอร์ และ ส่วน Table Designer

2.9.2 ประเภทของข้อมูลที่จัดเก็บใน SQL Server

กลุ่มของประเภทข้อมูลที่จัดเก็บใน SQL Server แบ่งออกเป็น 3 กลุ่ม คือ

1. กลุ่มข้อมูลตัวหนังสือ ประกอบด้วย

1) varchar ใช้ในการเก็บข้อมูลที่เป็นตัวหนังสือ โดยจะใช้พื้นที่ในการจัดเก็บเท่ากับข้อมูลจริง ขนาด 1-byte ต่อหนึ่งตัวอักษร มีขนาด 4,000 ตัวอักษรต่อฟิลด์ เหมาะสำหรับจัดเก็บข้อมูลที่เป็น ชื่อ นามสกุล

2) nvarchar ใช้ในการเก็บข้อมูลที่เป็นตัวหนังสือ โดยจะใช้พื้นที่ในการจัดเก็บเท่ากับข้อมูลจริง ขนาด 2-byte ต่อหนึ่งตัวอักษร มีขนาด 4,000 ตัวอักษรต่อฟิลด์ เหมาะสำหรับจัดเก็บข้อมูลที่เป็นตัวอักษรหลายภาษา

3) char ใช้ในการเก็บข้อมูลที่เป็นตัวหนังสือ โดยจะใช้พื้นที่ในการจัดเก็บเท่ากับที่ประกาศจองพื้นที่ไว้ โดยไม่สนขนาดของข้อมูลจริงว่ามีเท่าใด รองรับการเก็บข้อมูลเพียงภาษาเดียว

4) nchar ใช้ในการเก็บข้อมูลที่เป็นตัวหนังสือ โดยจะใช้พื้นที่ในการจัดเก็บเท่ากับที่ประกาศจองพื้นที่ไว้ โดยไม่สนขนาดของข้อมูลจริงว่ามีเท่าใด มีขนาด 4,000 ตัวอักษรต่อฟิลด์

5) ntext ใช้ในการเก็บข้อมูลที่เป็นตัวหนังสือที่ต้องใช้ขนาดพื้นที่มาก มีขนาด 1 ล้านตัวอักษรต่อหนึ่งฟิลด์

2. กลุ่มข้อมูลตัวเลข ประกอบด้วย

1) Int ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่เป็นจำนวนเต็มปกติ มีขนาด

2) Integer ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่เป็นจำนวนเต็มปกติ

3) Decimal ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่มีหลักทศนิยม ซึ่งทศนิยมที่จัดเก็บจะไม่มีการปัดเศษ

4) Numeric ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่มีหลักทศนิยม ต้องกำหนดขนาดของตัวเลขและทศนิยม

5) Real ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่มีหลักทศนิยม และมีค่าจำนวนมาก แต่มีการปัดเศษทศนิยม

6) Float ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่มีหลักทศนิยม และมีค่าจำนวนมาก ซึ่งทศนิยมที่จัดเก็บจะไม่มีการปัดเศษ

3. กลุ่มข้อมูลอื่น ๆ ประกอบด้วย

1) Date ใช้ในการเก็บข้อมูลที่เป็นวันที่

2) Time ใช้ในการเก็บข้อมูลที่เป็นเวลา

3) Data Time ใช้ในการเก็บข้อมูลที่เป็นวันที่และเวลา

4) Money ใช้ในการเก็บข้อมูลที่เป็นตัวเลขที่เกี่ยวกับเงิน

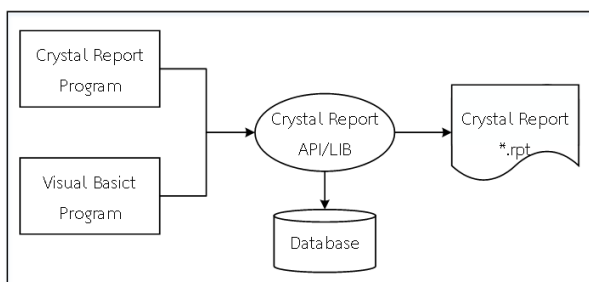
5) Year ใช้ในการเก็บข้อมูลปีซึ่งสามารถระบุได้แบบ 2 ตำแหน่งหรือ 4 ตำแหน่ง เช่น 40-59 (1940-2059) หรือ 2000-2019

2.10 Crystal Report

Crystal Report เป็นเครื่องมือที่ใช้ในการสร้างรายงานที่มีหลากหลายรูปแบบด้วยความสะดวกและง่ายต่อการใช้งาน และสามารถเชื่อมกับฐานข้อมูลได้ เช่น MS SQL Server, MS Access, XML เป็นต้น ซึ่งภายในงานวิจัยนี้อาศัยโปรแกรม Visual Basic Studio ในการเรียกใช้โปรแกรม crystal report มาเพื่อใช้ในการสร้างเอกสารประกอบการทดสอบซอฟต์แวร์ของระบบ เพื่อแสดงรายงานแก่ผู้ใช้งาน

2.10.1 การทำงานของ Crystal report

ในงานวิจัยนี้มีการใช้ Crystal report จากโปรแกรม Visual Basic Studio เพื่อทำการสร้างฟิลด์รายการที่ต้องการใช้ และเชื่อมฐานข้อมูลเพื่อดึงข้อมูลขึ้นมาตามฟิลด์ที่ต้องการ ซึ่งภายในงานวิจัยใช้งานข้อมูลในรูปแบบ MS SQL Server เมื่อดึงข้อมูลจากฐานข้อมูลเรียบร้อยแล้วจะได้รายงาน ซึ่งภาพรวมการทำงานของ crystal report แสดงดังรูปที่ 2.30



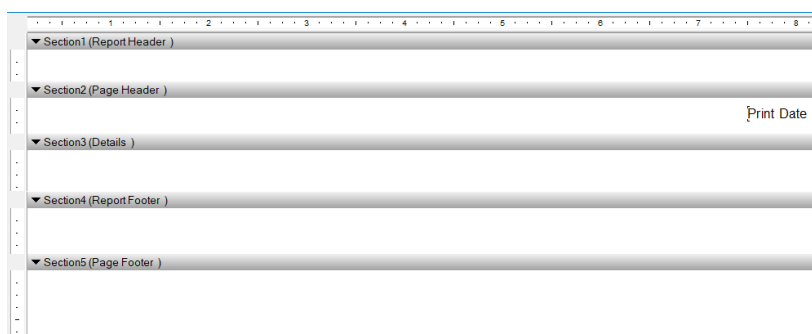
รูปที่ 2.31 การทำงานของ crystal report

2.10.2 ส่วนประกอบของ Crystal report

ภายใน crystal report ประกอบด้วย 5 ส่วน คือ พื้นที่ในการทำงาน ส่วนการแสดงผล การจัดกลุ่ม การแสดงกราฟ และการสร้างรายงานย่อย

1. พื้นที่ในการทำงาน หรือ section

พื้นที่ในการทำงานเป็นส่วนที่ใช้ในการสร้างรูปแบบรายงาน ใช้ในการวางฟิลด์ วางกราฟ หรือ ข้อมูลต่าง ๆ ที่ต้องการนำมาสร้างเป็นรายงาน ซึ่งภายในส่วนพื้นที่การทำงานนั้น crystal report แบ่งออกเป็น 5 ส่วน แสดงดังรูปที่ 2.32



รูปที่ 2.32 พื้นที่การทำงานภายใน crystal report

ส่วนที่ 1 ส่วนหัวรายงาน (Report Header) เป็นส่วนหัวเรื่องของรายงาน ซึ่งการออกแบบในส่วนนี้จะปรากฏที่หน้าแรกของรายงานเพียงหน้าเดียวโดยปรากฏที่ส่วนบนของหน้าแรก เช่น ชื่อโปรเจค ชื่อบริษัท ที่อยู่บริษัท ชื่อลูกค้า เป็นต้น

ส่วนที่ 2 ส่วนหัวของหน้า (Page Header) ในส่วนนี้จะอยู่ส่วนบนของทุกหน้ารายงาน

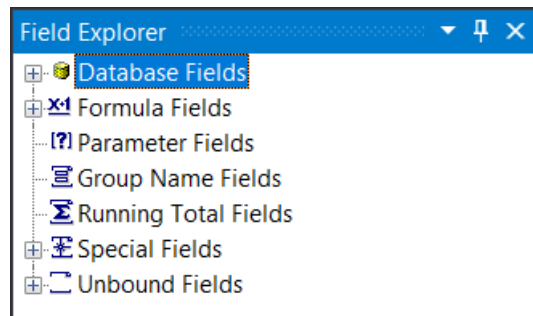
ส่วนที่ 3 รายละเอียด (Details) เป็นส่วนที่แสดงรายละเอียดของข้อมูลแต่ละรายการ ซึ่งเป็นส่วนหลักและส่วนสำคัญของการออกแบบรายงาน โดยที่ข้อมูลที่แสดงนั้นจะนำมาจากฐานข้อมูล เช่น รหัสพนักงาน ชื่อพนักงาน หรือเบอร์โทรศัพท์ เป็นต้น

ส่วนที่ 4 ส่วนท้ายของหน้า (Report Footer) เป็นส่วนที่ใช้ในการสรุปเรื่องรายงาน ซึ่งจะปรากฏในส่วนสุดท้ายของหน้ารายงานเพียงหน้าเดียว ซึ่งจะปรากฏในหน้าสุดท้าย

ส่วนที่ 5 ส่วนท้ายของรายงาน (Page Footer) เป็นส่วนที่แสดงเป็นลำดับสุดท้ายของแต่ละกลุ่มเมื่อแสดงรายละเอียดของข้อมูลในกลุ่มนั้นแล้ว ซึ่งจะแสดงที่ส่วนล่างของหน้ารายงานในแต่ละหน้า

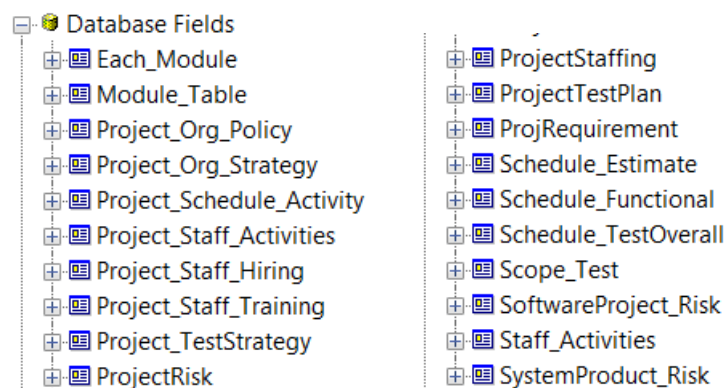
2. ส่วนที่ใช้ในการแสดงข้อมูล หรือ fields

ส่วนที่ใช้ในการแสดงผลข้อมูล (Fields) ที่ประกอบด้วย ฐานข้อมูล สูตรการคำนวณ การแบ่งกลุ่ม ขอบเขตของข้อมูล แบ่งออกเป็นส่วนต่าง ๆ แสดงดังรูปที่ 2.33



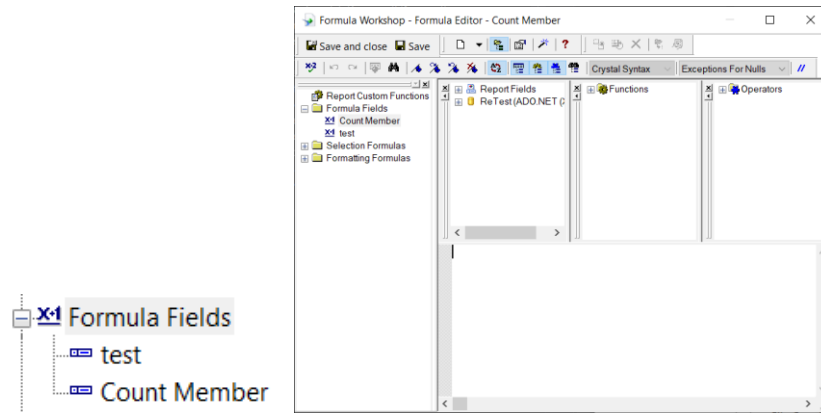
รูปที่ 2.33 ส่วนประกอบของ crystal report

1) Database Fields เป็นส่วนในการดึงข้อมูลจากฐานข้อมูล (database) ซึ่งแสดงตารางข้อมูลทั้งหมดที่ใช้ในการสร้างรายงาน ตัวอย่างฐานข้อมูลที่แสดงในฟิลด์ ดังรูปที่ 2.34



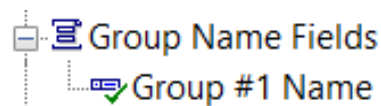
รูปที่ 2.34 Database Fields ใน crystal report

2) Formula Fields เป็นส่วนที่ใช้ในการใส่สูตรเพื่อคำนวณในรายงาน โดยเริ่มที่สร้างกลุ่มฟิลด์ขึ้นมาก่อน จากนั้นทำการระบุฟิลด์ที่ต้องการคำนวณ ซึ่งตัวอย่างแสดงดังรูปที่ 2.35



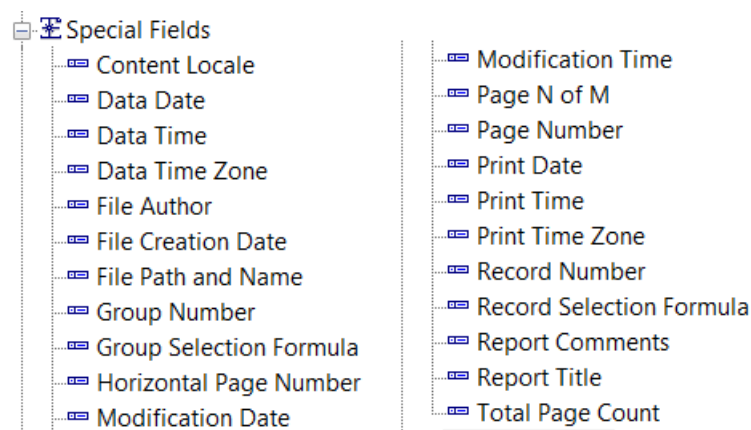
รูปที่ 2.35 รูปแบบของ Formula Fields

- 3) Parameter Fields เป็นส่วนที่ใช้ในการรับค่าจากภายนอก
- 4) Group Name Fields เป็นส่วนที่แสดงชื่อกลุ่มต่าง ๆ ที่ได้แบ่งไว้ ซึ่งจำเป็นส่วนเดียวกับ “การจัดกลุ่ม หรือ group” ในข้อที่ 3 ซึ่งตัวอย่างแสดงดังรูปที่ 2.36



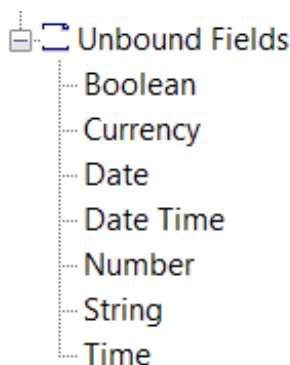
รูปที่ 2.36 รูปแบบ Group Name Fields

- 5) Running Total Fields เป็นส่วนที่เก็บค่าตัวเลข และ ผลรวมต่าง ๆ
- 6) Special Fields เป็นส่วนฟังก์ชันพิเศษต่าง ๆ ที่ใช้ในการสร้างรายงาน เช่น เวลาในการพิมพ์ เวลาในการพิมพ์ เลขหน้ารายงาน เป็นต้น ซึ่งตัวอย่างฟิลด์แสดงดังรูปที่ 2.37



รูปที่ 2.37 รูปแบบ Special Fields

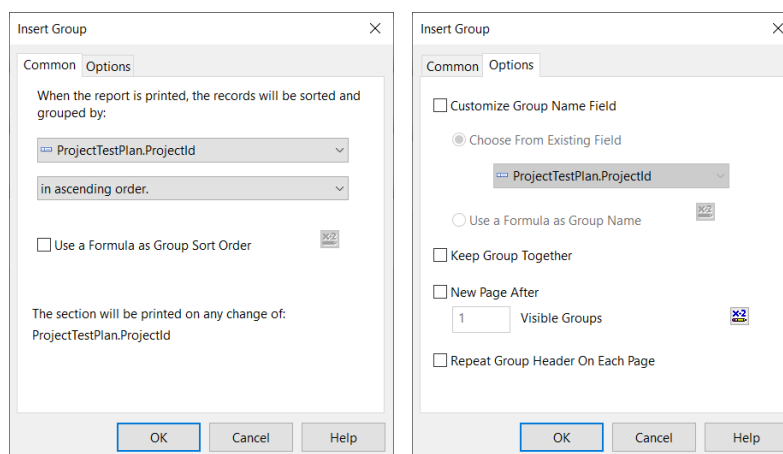
- 7) Unbound Fields เป็นส่วนที่แสดงประเภทของข้อมูล เช่น ค่าเงิน เวลา วัน-เวลา ตัวเลข เป็นต้น ซึ่งตัวอย่างฟิลด์แสดงดังรูปที่ 2.38



รูปที่ 2.38 รูปแบบ Unbound Fields

3. การจัดกลุ่ม หรือ group

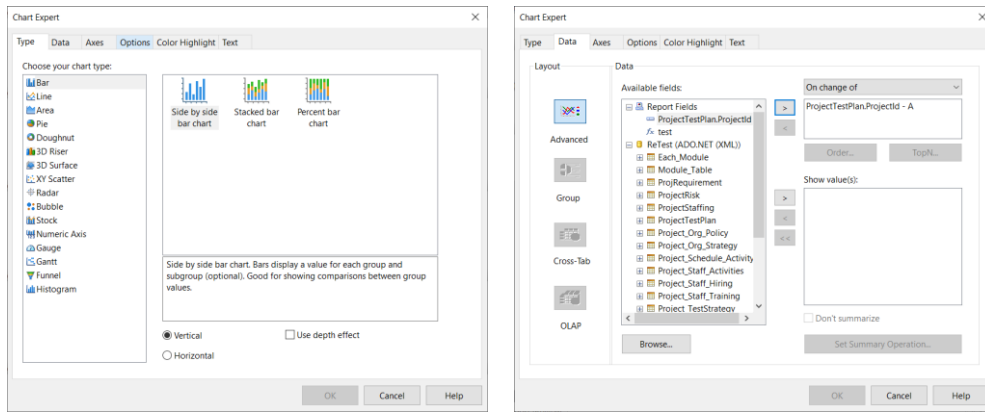
ส่วนในการนำข้อมูลจากฐานข้อมูลมาสร้างกลุ่ม เพื่อให้สะดวกต่อการสร้างรายงานในแต่ละหน้า และทำให้สามารถจัดลำดับความสัมพันธ์ได้ง่ายขึ้น โดยต้องเลือกเรคคอร์ด (record) ที่ใช้ในการจัดกลุ่ม เช่น เลือก projected จากนั้นทำการจัดกลุ่มที่ต้องการตามเรคคอร์ดที่เลือกไว้ ซึ่งตัวอย่างแสดงดังรูปที่ 2.39



รูปที่ 2.39 การจัดกลุ่มข้อมูลเพื่อทำรายงาน

4. การแสดงกราฟ หรือ chart

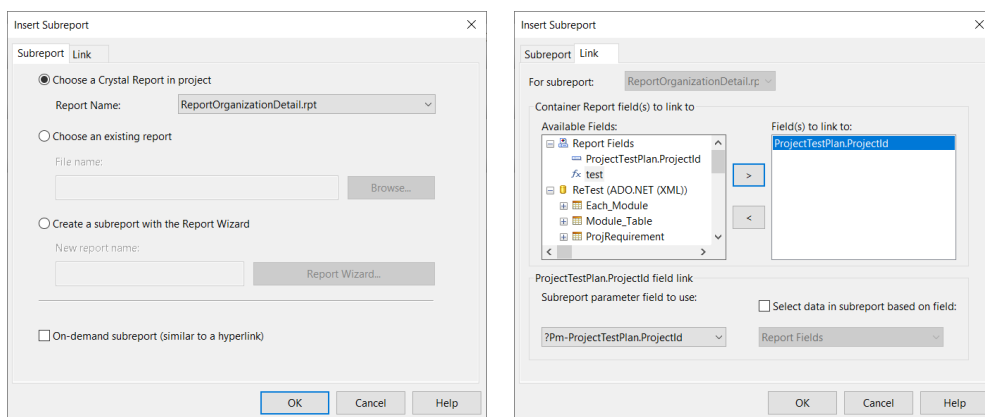
ส่วนในการนำแสดงข้อมูลในรูปแบบกราฟ ซึ่งภายใน crystal report ประกอบด้วยกราฟหลายประเภท เช่น การเส้น กราฟแท่ง เป็นต้น เมื่อเลือกประเภทกราฟแล้วก็ทำการเลือกฟิลด์ที่ต้องการสร้าง ซึ่งตัวอย่างแสดงดังรูปที่ 2.40



รูปที่ 2.40 รายงานที่แสดงข้อมูลในรูปแบบกราฟ

5. การสร้างรายงานย่อย หรือ SubReport

ส่วนที่สร้างรายงานย่อยภายใต้รายงานหลักที่ได้สร้างขึ้นมา โดยต้องเลือกไฟล์รายงานที่ต้องการนำมาสร้าง จากนั้นเลือกฟิลด์ที่ต้องการ ซึ่งแสดงตัวอย่างดังรูปที่ 2.41



รูปที่ 2.41 การสร้างรายงานส่วนย่อย

2.11 งานวิจัยที่เกี่ยวข้อง

2.11.1 Tu Honglei¹, Sun Wei¹ และ Zhang Yanan การวิจัยเกี่ยวกับเมตริกซอฟต์แวร์และการวัดความซับซ้อนของซอฟต์แวร์ [19]

จากงานวิจัยเกี่ยวกับเมตริกซอฟต์แวร์และการวัดความซับซ้อนของซอฟต์แวร์ ได้กล่าวว่า การกำหนดความหมายของเมตริกซอฟต์แวร์และประวัติความเป็นมาและประเภทของเมตริกซอฟต์แวร์ถูกมองเป็นภาพรวม การวัดความซับซ้อนของซอฟต์แวร์เป็นองค์ประกอบที่สำคัญของเมตริกซอฟต์แวร์และเกี่ยวข้องกับค่าใช้จ่าย การบำรุงรักษาในการพัฒนาซอฟต์แวร์เพื่อปรับปรุงคุณภาพซอฟต์แวร์และความสามารถในการควบคุมโครงการจำเป็นต้องควบคุมความซับซ้อนของซอฟต์แวร์ด้วยการวัดผลที่เกี่ยวข้อง ซึ่งงานวิจัยนี้ได้กล่าวถึงวิธี McCabe และเมตริก C & K สำหรับตัวอย่างของเมตริกความซับซ้อน

จากการศึกษางานวิจัยพบว่าเมตริกซอฟต์แวร์กลายเป็นพื้นฐานของการจัดการซอฟต์แวร์และเป็นสิ่งจำเป็นต่อการพัฒนาซอฟต์แวร์ให้สำเร็จ การขยายตัวของความซับซ้อนของซอฟต์แวร์ช่วยกระตุ้นการเกิดเมตริกที่วัดความซับซ้อนของซอฟต์แวร์ ความซับซ้อนของซอฟต์แวร์จะส่งผลโดยตรงต่อความน่าเชื่อถือของซอฟต์แวร์ ถึงแม้ผู้คนจะตระหนักถึงความสำคัญของเมตริกซอฟต์แวร์ ในส่วนเมตริกก็ยังคงต้องการที่จะเติบโตอย่างรวดเร็วเพื่อตอบสนองความต้องการในการพัฒนาซอฟต์แวร์ การขาดทฤษฎีพื้นฐานและการกลเม็ดของวิธีการวัดซอฟต์แวร์ยังคงเป็นเรื่องเล็กเมื่อเทียบกับทฤษฎีซอฟต์แวร์และแนวปฏิบัติอื่น ๆ

2.11.2 Tamer Abdou, Peter Grogono และ Pankaj Kamthan กรอบแนวคิดสำหรับกระบวนการทดสอบซอฟต์แวร์จากแหล่งที่มาแบบเปิด [20]

จากงานวิจัยเกี่ยวกับกรอบแนวคิดสำหรับกระบวนการทดสอบซอฟต์แวร์จากแหล่งที่มาแบบเปิด พบว่าการยอมรับและใช้ Open Source Software (OSS) ในวงกว้างได้เน้นย้ำถึงความจำเป็นในการตรวจสอบวิธีการรักษาคุณภาพ ด้วยวัตถุประสงค์ในการระบุกระบวนการทดสอบ OSS ได้มีการศึกษาโครงการ OSS ที่รู้จักกันดีสามโครงการ ได้แก่ เซิร์ฟเวอร์ Apache HTTP, Mozilla Web browser และ NetBeans IDE ในการศึกษาเหล่านี้พบว่ามีการผสมผสานอย่างใกล้ชิดกับกิจกรรมของมาตรฐานการทดสอบตามมาตรฐาน ISO / IEC อย่างไรก็ตามความแตกต่างที่สำคัญเกิดขึ้นในงานที่เกี่ยวข้องกับแต่ละกิจกรรมของกระบวนการทดสอบ เพื่อจัดระบบกระบวนการทดสอบ OSS ให้เสนอกรอบการทดสอบกระบวนการซอฟต์แวร์โอเพนซอร์ส (OSS-TPF) การจัดตำแหน่ง OSS-TPF กับมาตรฐานการทดสอบ ISO / IEC

จากการศึกษางานวิจัยพบว่า การขยายตัวของ OSS จำเป็นต้องให้ความสนใจกับคุณภาพและวิธีการทดสอบอย่างเป็นระบบเป็นวิธีการที่ต้องตระหนักถึง การศึกษาเชิงประจักษ์และความครอบคลุมของ OSS-TPF ในงานวิจัยแสดงให้เห็นว่าการทดสอบมีบทบาทสำคัญในการพัฒนา OSS การเปรียบเทียบกระบวนการทดสอบ ISO / IEC และ OSS-TPF แสดงให้เห็นถึงความคล้ายคลึงกันระหว่างสอง อย่างไรก็ตามยังแสดงให้เห็นถึงความแตกต่างที่โดดเด่นและแสดงให้เห็นถึงศักยภาพในการปรับปรุงกระบวนการทดสอบใน OSS

2.11.3 Paulo Eira, Ricardo J.Machado และ António Silva การปรับปรุงตามมาตรฐาน ISO / IEC / IEEE 29119-3 สำหรับวิสาหกิจขนาดกลางและขนาดย่อม [2]

จากงานวิจัยเกี่ยวกับการปรับปรุงตามมาตรฐาน ISO / IEC / IEEE 29119-3 สำหรับวิสาหกิจขนาดกลางและขนาดย่อม พบว่าการประยุกต์ใช้มาตรฐาน ISO 29119-3 ในวิสาหกิจขนาดกลางและขนาดย่อม (SMEs) โดยนำเสนอกระบวนการทำซ้ำและได้ตอบสำหรับการปรับปรุง โดยมีเป้าหมายคือเพื่อตอบสนองความต้องการของกระบวนการทดสอบของ SMEs เหล่านี้ซึ่งรวมกระบวนการพัฒนาซอฟต์แวร์กับจุดสำคัญของมาตรฐาน ISO 29119-3 โดยไม่ละเมิดความสมบูรณ์ กระบวนการปรับปรุง

จะถูกนำมาใช้และได้รับการกล่าวถึงในการตั้งค่า ซึ่งผลลัพธ์จะแสดงให้เห็นถึงการปรับปรุงกระบวนการทดสอบของ SME โดยใช้ ISO 29119-3 เป็นแนวทางในการลดทรัพยากรจำนวนมาก

จากการศึกษางานวิจัยพบว่าการปรับใช้มาตรฐาน ISO 29119-3 (ส่วนที่สามของมาตรฐาน) สำหรับ SMEs ที่ดำเนินการวิจัยเพื่อเสนอกระบวนการที่สอดคล้องกับมาตรฐานเดิมที่ประกอบด้วยผลลัพธ์ 4 ค่า ซึ่งการรวมมาตรฐานในองค์กรไม่ใช่เรื่องง่าย เนื่องจากความซับซ้อนและการเพิ่มข้อมูลเมื่อใช้การปรับแต่ง 4 ข้อกับ ISO 29119-3 ก็สามารถกำหนดชุดเอกสารที่ลดลงให้เพียงพอสำหรับใช้ในองค์กร SME โดยมีที่พัฒนาประกอบด้วย 5 องค์ประกอบ ซึ่งในงานวิจัยนี้ได้นำเสนอแนวทางการปรับปรุงสำหรับการทดสอบเอกสารซึ่งแม้ว่าจะลดลงอย่างมากเมื่อเทียบกับมาตรฐาน ISO 29119-3 มาตรฐานฉบับเดิมยังคงสอดคล้องกับเอกสารของมาตรฐานตั้งต้นองค์กร SME จึงสามารถใช้คู่มือแนวทางการปฏิบัติในการทดสอบซอฟต์แวร์ได้

2.11.4 LJUBOMIR LAZIĆ คุณภาพและการวัดซอฟต์แวร์ [21]

จากงานวิจัยเกี่ยวกับคุณภาพและการวัดซอฟต์แวร์ ที่มุ่งเน้นไปที่การทดสอบซอฟต์แวร์และการวัดซึ่งสามารถประเมินผลเชิงปริมาณของกระบวนการพัฒนาซอฟต์แวร์ที่สำคัญได้ ซึ่งได้อธิบายถึงรูปแบบตัววัดคุณภาพซอฟต์แวร์และการทดสอบซึ่งเป็นกรอบสำหรับการสร้างและตีความเมตริกในกระบวนการในการพัฒนาซอฟต์แวร์ โมเดลนี้ได้รับการตรวจสอบและใช้ในโครงการซอฟต์แวร์ขนาดใหญ่ในองค์กรพัฒนาซอฟต์แวร์ ประเด็นสำคัญสำหรับเมตริกในกระบวนการแนวคิดและความหมายของรูปแบบและการใช้งาน

จากการศึกษางานวิจัยพบว่า เมตริกกำลังได้รับความสำคัญและเป็นที่ยอมรับในภาคธุรกิจในองค์กรที่เติบโตและมุ่งมั่นที่จะปรับปรุงคุณภาพขององค์กร การวัดกระบวนการทดสอบเป็นกระบวนการที่จำเป็นสำหรับผู้จัดการการทดสอบ เพื่อให้ซอฟต์แวร์มีประสิทธิภาพในการออกแบบและประเมินกลยุทธ์การทดสอบที่คุ้มค่า การจัดการที่มีประสิทธิภาพของกระบวนการ ใด ๆ ต้องใช้การวัดปริมาณและการสร้างแบบจำลอง ซึ่งได้อธิบายถึงรูปแบบความพยายามหรือผลลัพธ์สำหรับเมตริกการทดสอบซอฟต์แวร์ในกระบวนการและการจัดการด้านคุณภาพ โมเดลนี้ใช้วิธีการตีความเมตริกแบบเดิม ๆ และประเมินคุณภาพเมื่อโครงการพัฒนาซอฟต์แวร์กำลังอยู่ระหว่างการดำเนินการได้แสดงให้เห็นถึงการใช้แบบจำลองสำหรับขั้นตอนการทดสอบรวมทั้งส่วนหน้าของกระบวนการพัฒนาในเรื่องการออกแบบและการตรวจสอบโค้ด แนวทางการปรับปรุงและความสามารถในการใช้ตัวชี้วัดผลลัพธ์ที่สามารถนำมาใช้ได้ตลอดช่วงวัฏจักรการพัฒนาซอฟต์แวร์ ความพยายามในการทดสอบแบบจำลอง ผลลัพธ์และเมตริกที่สร้างขึ้นเป็นวิธีการที่มีประสิทธิภาพในการจัดการคุณภาพ

2.11.5 Ning Chen, Ethan W. Chen และ Ian. Chen การรวบรวมมาตรฐานการทดสอบซอฟต์แวร์ ISO / IEC / IEEE 29119 กับ การพัฒนาแบบอาร์จาย [22]

จากงานวิจัยเกี่ยวกับการรวบรวมมาตรฐานการทดสอบซอฟต์แวร์ ISO / IEC / IEEE 29119 กับ การพัฒนาแบบอาร์จาย พบว่า ความต้องการใช้มาตรฐานการทดสอบเป็นกลไกที่รวมมาตรฐานการทดสอบเข้ากับการพัฒนาซอฟต์แวร์แบบอาร์จาย และมีวิธีการปรับอย่างไรให้สอดคล้องกับระดับที่เหมาะสมซึ่งเกี่ยวข้องกับระดับวุฒิภาวะของการทดสอบการรวมตัวของรูปแบบการชำระเงิน (TMMi)

จากการศึกษางานวิจัยพบว่า การรวมแบบทดสอบการทดสอบแบบครบกำหนด (TMMi) ที่ได้แนะนำระดับวุฒิภาวะ 5 ระดับได้แก่ ระดับเริ่มต้น ระดับการจัดการ ระดับการกำหนด ระดับการวัด และระดับความเหมาะสม โดยที่จำนวนขั้นตอนของเอกสารและเครื่องมือในการตั้งค่าอาร์จายสามารถวัดได้ตามระดับที่กำหนดตามที่แนะนำโดย TMMi ซึ่งในงานวิจัยนี้ได้นำเสนอตัวอย่างการปฏิบัติตามข้อกำหนดของ IEEE 29119 ที่กำหนดเป้าหมายระดับการกำหนดวุฒิภาวะ TMMi 2 (เช่น Managed) ซึ่งเหมาะสำหรับโปรเจกแบบอาร์จายเป็นส่วนใหญ่

2.11.6 โปรแกรมสำหรับการทดสอบซอฟต์แวร์ Modern Test Case Management Software for QA and Development Teams [23]

จากโปรแกรมที่เกี่ยวข้องกับการทดสอบซอฟต์แวร์ที่มุ่งเน้นไปในส่วนการสร้างกรณีทดสอบ (Test Cases) ซึ่งภายในโปรแกรมประกอบฟังก์ชันต่าง ๆ ดังนี้ การจัดการบทความ การทดสอบแบบอัตโนมัติ การทดสอบแบบฟังก์ชัน การติดตามผลการทดสอบ รายงานและเมตริกซ์ โดยมีลักษณะการทำงานของฟังก์ชันในการสร้างกรณีทดสอบ แสดงดังรูปที่ 2.42

จากการศึกษาโปรแกรมการจัดการกรณีทดสอบซอฟต์แวร์ ทำให้พบว่า โปรแกรมได้ถูกพัฒนาขึ้นมาในรูปแบบ web-based ซึ่งในบางกรณีอาจจะไม่สะดวกหากเป็นกลุ่มบริษัทขนาดเล็ก หรือ สำหรับผู้เริ่มต้นในการพัฒนาและทดสอบซอฟต์แวร์ โดยต้องมีการเพิ่มค่าใช้จ่าย ทางผู้วิจัยจึงเลือกใช้การพัฒนาแบบ windows application ที่ไม่จำเป็นต้องอาศัยอินเทอร์เน็ตในการใช้งาน และฟังก์ชันการสร้างกรณีทดสอบของโปรแกรมการจัดการกรณีทดสอบซอฟต์แวร์ มีคุณลักษณะที่ไม่ครอบคลุมตามมาตรฐานสากล และไม่ครอบคลุมในบางคุณลักษณะที่จำเป็นสำหรับการทำงาน เช่น สภาพแวดล้อมที่ใช้สำหรับการทดสอบกรณีทดสอบที่ได้สร้างขึ้น หรือการจัดการกับความต้องการของลูกค้า (Requirement) ที่ต้องครบถ้วน พร้อมทั้งจับคู่ความสัมพันธ์กับฟังก์ชันที่นำมาทดสอบ ทางผู้วิจัยจึงได้ทำมาตรฐานการทดสอบสากลมาช่วยในการจัดการกับปัญหาเหล่านี้ พร้อมทั้งได้สร้างฟังก์ชันอื่น ๆ ที่เกี่ยวข้องกับการทดสอบ เพื่อให้ได้กระบวนการ และเอกสารสำหรับการทดสอบที่ครบถ้วน และครอบคลุมที่สุด

C1 Edit Test Case

Title *
Format table with built-in style

Section * Prerequisites | **Template *** Test Case (Text) | **Type *** Other | **Priority *** Medium

Estimate 5m | **References** | **Automation Type** None

Preconditions
In vulputate libero at http://www.example.com/.
* Donec placerat, dui vitae
* Ut auctor mi erat ac dolor.
* Lorem ipsum dolor sit amet, consectetur adipiscing elit.

The preconditions of this test case. Reference other test cases with [C#](e.g. [C17]).

Steps
In vulputate libero ****in nulla feugiat tincidunt****:
* Felis libero varius orci, in vulputate
* Massa turpis scelerisque diam.
* Nunc et felis est. Phasellus laoreet nibh vel augue
* Faucibus at varius est pretium.

The required steps to execute the test case.

Expected Result

Countrylist.csv
Other File, 2B

Drop files here to attach, or click to browse.

Actions
Delete a test case to remove it from its test suite. This also deletes all related running tests.

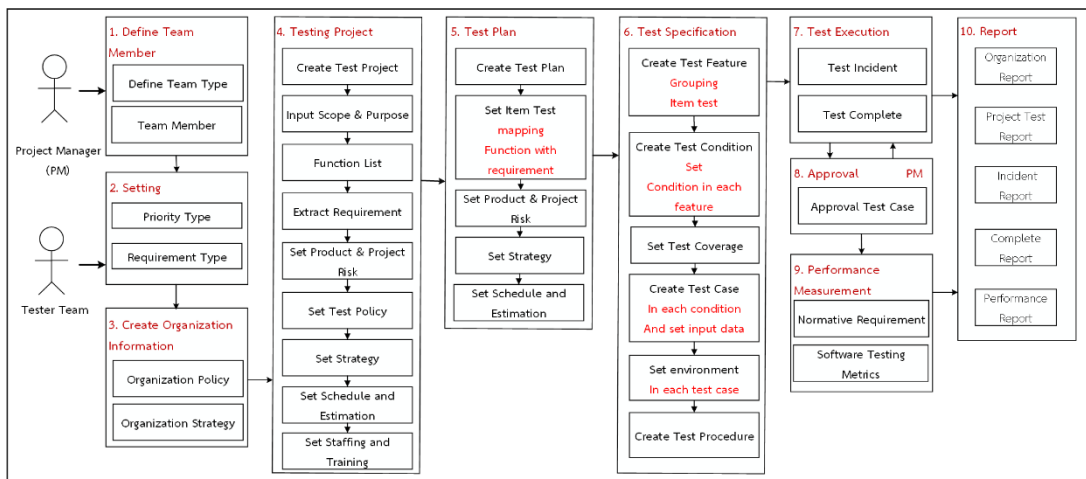
Delete this test case

รูปที่ 2.42 โปรแกรมสร้างกรณีทดสอบ (Modern Test Case Management Software)

บทที่ 3

วิธีการดำเนินงานวิจัย

งานวิจัยนี้มีจุดประสงค์เพื่อสร้างระบบทดสอบซอฟต์แวร์ตามวงจรการทดสอบ และมาตรฐานสากลในการทดสอบซอฟต์แวร์และมาตรฐานสากลของเอกสารทดสอบ พร้อมทั้งวัดประสิทธิภาพของการทดสอบซอฟต์แวร์และประสิทธิภาพของซอฟต์แวร์ที่ได้รับการทดสอบ จากการศึกษาวิจัยได้ทำการสร้างความสัมพันธ์ระหว่างขั้นตอนการทดสอบซอฟต์แวร์กับมาตรฐานการทดสอบซอฟต์แวร์ทั้ง ISO 29119 และ IEEE 829 เพื่อสร้างระบบที่มีฟังก์ชันการทำงานครอบคลุมทุกขั้นตอนของการทดสอบ เพื่อให้ได้เอกสารประกอบการทดสอบซอฟต์แวร์ที่ครบถ้วนสมบูรณ์ตามขั้นตอนการทดสอบและมาตรฐานในการทดสอบซอฟต์แวร์ที่สามารถนำไปใช้ในกระบวนการทดสอบได้



รูปที่ 3.1 แผนภาพการทำงานของระบบ

ภาพรวมของระบบทดสอบซอฟต์แวร์มีการแบ่งส่วนการทำงานออกเป็น 10 ส่วน ดังรูปที่ 3.1

ส่วนที่ 1 Define Team Member ส่วนในการกำหนดผู้ใช้ระบบ โดยมีการแบ่งออกเป็น 2 กระบวนการย่อย คือ กำหนดประเภทของกลุ่มสมาชิก (Define Team Type) และ กำหนดสมาชิกในกลุ่ม (Team Member)

ส่วนที่ 2 Setting ส่วนในการตั้งค่าการใช้งานเริ่มต้น โดยมีการแบ่งออกเป็น 2 การตั้งค่า คือ การตั้งค่าลำดับความสำคัญ (Priority Type) และ การตั้งค่าประเภทความต้องการ (Requirement Type)

ส่วนที่ 3 Create Organization Information การระบุข้อมูลของบริษัท หรือ องค์กรที่เข้าไปพัฒนาระบบ และทดสอบระบบ ซึ่งประกอบด้วยข้อมูลเบื้องต้น (เช่น ผู้บริหารองค์กร ประเภท

ผลิตภัณฑ์ขององค์กร บุคคลที่เกี่ยวข้องกับโปรเจกต์ทดสอบ) ขององค์กรที่จำเป็นสำหรับการดำเนินการ ประกอบด้วย ข้อมูลองค์กรเบื้องต้น การกำหนดนโยบาย (Policy) และกำหนดกลยุทธ์ในการแก้ไข (Strategy)

ส่วนที่ 4 Testing Project ขั้นตอนในการสร้างโปรเจกต์ทดสอบซอฟต์แวร์ (Test Project) ซึ่งจะเชื่อมโยงกับข้อมูลในส่วนที่ 1 และ ส่วนที่ 2 เพื่อใช้ในการกำหนดข้อมูลเบื้องต้นของโปรเจกต์การทดสอบ ในส่วนที่เกี่ยวข้องกับองค์กร (เช่น แผนนโยบาย แผนกลยุทธ์) สร้างโปรเจกต์การทดสอบที่ประกอบด้วย การสกัดความต้องการของลูกค้า แล้วแบ่งประเภทกับฟังก์ชันที่มีอยู่ในระบบ เพื่อสร้างแผนการทดสอบ และกรณีทดสอบในลำดับต่อไป ซึ่งประกอบด้วย การกำหนดขอบเขตและวัตถุประสงค์ การกำหนดฟังก์ชันของระบบ การแยกความต้องการ (Requirement) การตั้งค่าความเสี่ยงของโปรเจกต์และผลิตภัณฑ์ การตั้งค่านโยบายการทดสอบ การตั้งค่ากลยุทธ์การทดสอบ การตั้งค่าแผนการดำเนินการ พร้อมประเมินค่าใช้จ่ายในกระบวนการทดสอบ และการตั้งค่าการฝึกอบรมหากลูกค้าต้องการ

ส่วนที่ 5 Test Plan ส่วนที่ใช้ในการวางแผนการทดสอบ ประกอบด้วย การตั้งค่าสิ่งที่จะทดสอบ โดยการจับคู่ระหว่างฟังก์ชันและความต้องการของลูกค้า การตั้งค่าความเสี่ยงของโปรเจกต์และผลิตภัณฑ์ การตั้งค่ากลยุทธ์การทดสอบ และการตั้งค่าแผนการดำเนินการ พร้อมประเมินค่าใช้จ่ายในกระบวนการทดสอบ

ส่วนที่ 6 Test Specification ส่วนในการกำหนดกรณีทดสอบ ประกอบด้วย การจัดกลุ่มความต้องการและสิ่งที่ต้องการทดสอบ ที่มีความคล้ายกันเข้าด้วยกัน การตั้งค่าเงื่อนไขให้กลุ่มที่ต้องการทดสอบ สร้างกรณีทดสอบโดยแยกกลุ่มที่เลือกไว้เป็นกรณีไป พร้อมตั้งค่าสภาพแวดล้อมที่ใช้สำหรับการทดสอบ และสร้างขั้นตอนการทดสอบโดยรวมกรณีทดสอบเข้าด้วยกัน

ส่วนที่ 7 Test Execution ขั้นตอนในการนำผลลัพธ์ส่วนที่ 4, 5 และ 6 มาดำเนินการทดสอบตามแผน เมื่อทดสอบแล้วจะได้ผลการทดสอบซึ่งแบ่งออกเป็น 2 ประเภทคือ การทดสอบที่มีปัญหาหรือทดสอบไม่สมบูรณ์ (Test Incident) และการทดสอบที่สมบูรณ์ (Test Complete)

ส่วนที่ 8 Approval ส่วนที่ใช้ในการอนุมัติกรณีทดสอบที่ผ่านการทดสอบสมบูรณ์เพื่อที่จะส่งกลับไปยังฟังก์ชัน Test Complete เพื่อสร้างเอกสารการทดสอบที่สมบูรณ์ โดยในส่วนนี้ผู้จัดการโครงการมีหน้าที่ในการดำเนินการเพียงผู้เดียวเท่านั้น

ส่วนที่ 9 Performance Measurement กระบวนการวัดประสิทธิภาพของซอฟต์แวร์ที่ประกอบด้วย 2 ส่วน คือ Normative Requirement, Software Testing Metrics

ส่วนที่ 10 Report เป็นส่วนสุดท้ายของระบบที่แสดงรายงานต่าง ๆ ที่ได้สร้างขึ้นจากระบบ ประกอบด้วย เอกสารที่เกี่ยวกับข้อมูลขององค์กรที่ต้องการทดสอบ (Organization Report) เอกสารที่เกี่ยวข้องกับโปรเจกต์ทดสอบ (Project Test Report) ที่ประกอบด้วยแผนการทดสอบ (Test Plan) และ กรณีทดสอบ (Test Case) เอกสารรายงานเหตุการณ์การทดสอบ (Incident Report)

เอกสารรายงานการทดสอบที่สมบูรณ์ (Complete Report) และ เอกสารรายงานประสิทธิภาพสร้างกรณีทดสอบ (Performance Report)

3.1 เครื่องมือที่ใช้ในการวิจัย

การวิจัยในครั้งนี้เครื่องมือที่ผู้วิจัยเลือกใช้ประกอบด้วย Visual Basic Studio, Microsoft SQL Server และแบบสอบถาม โดยแสดงรายละเอียดดังนี้

1. Visual Basic Studio ซึ่งเป็นโปรแกรม Open Source ที่ใช้สำหรับการพัฒนาระบบ โดยใช้ภาษา Visual basic (vb.net) สามารถใช้งานได้ฟรี
2. Microsoft SQL Server ซึ่งเป็นโปรแกรม Open Source ที่ใช้สำหรับเก็บข้อมูลภายในระบบ สามารถใช้งานได้ฟรี
3. แบบสอบถาม ใช้ในการสำรวจความพึงพอใจของผู้ใช้งานระบบ โดยภายในแบบสอบถามประกอบด้วยส่วนประกอบ 3 ส่วน ดังนี้

ส่วนที่ 1 แบบสอบถามสร้างในแบบมาตราส่วนประมาณค่า (Rating Scale) ที่มี 5 ระดับ ให้นำหน้าหน้าของการประเมินค่าดังนี้

ระดับคะแนน	5	หมายถึง	พึงพอใจมากที่สุด
ระดับคะแนน	4	หมายถึง	พึงพอใจมาก
ระดับคะแนน	3	หมายถึง	พึงพอใจปานกลาง
ระดับคะแนน	2	หมายถึง	พึงพอใจน้อย
ระดับคะแนน	1	หมายถึง	พึงพอใจน้อยที่สุด

ส่วนที่ 2 การเก็บรวบรวมข้อมูล แบบประเมินการใช้งานระบบผู้วิจัยทำการเก็บรวบรวมข้อมูลจากผู้ที่มีหน้าที่ในการพัฒนาโปรแกรม และผู้ที่ทำหน้าที่ที่เกี่ยวข้องทางคอมพิวเตอร์ ประกอบด้วย ผู้มีหน้าที่ในการตรวจสอบคุณภาพงานภายในองค์กร รวมทั้งหมด 20 คน

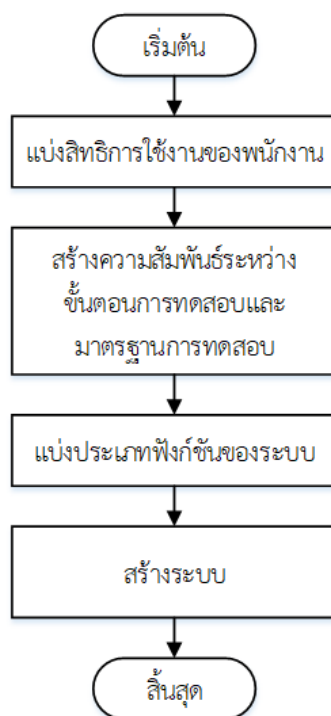
ส่วนที่ 3 ส่วนประกอบของแบบสอบถามประกอบด้วยหัวข้อคำถาม ดังนี้

ความปลอดภัย (security) ความพร้อมใช้งาน (availability) ประสิทธิภาพ (efficiency) ความซื่อตรง (integrity) ความน่าเชื่อถือ (reliability) ความถูกต้อง (accuracy) การใช้งาน (usability) ความอยู่รอดได้ (survivability) การตรวจสอบได้ (testability) และความยืดหยุ่นของระบบ (flexibility)

3.2 ขั้นตอนการพัฒนาการพัฒนาระบบ

การพัฒนาการพัฒนาระบบทดสอบซอฟต์แวร์ประกอบด้วยขั้นตอนการวางแผนพัฒนาระบบ ศึกษาถึงสิทธิการเข้าใช้งานภายในระบบ เพื่อป้องกันข้อมูลที่มีความเสี่ยง และเป็นการรักษาความปลอดภัยทางบริษัท พัฒนาระบบโดยสร้างฟังก์ชันจากการหาความสัมพันธ์ระหว่างมาตรฐานสากลรวมกับ

วิธีการทดสอบที่ปฏิบัติโดยทั่วไป แบ่งประเภทฟังก์ชันตามการใช้งาน ซึ่งมีรายละเอียด และขั้นตอนการดำเนินการ ดังรูปที่ 3.2



รูปที่ 3.2 ภาพรวมขั้นตอนการดำเนินงานวิจัย

3.2.1 สิทธิการใช้งานระบบ

การกำหนดสิทธิการเข้าถึงการใช้งานของระบบถูกกำหนดขึ้นเพื่อควบคุมความปลอดภัยในการใช้งาน โดยแบ่งสิทธิออกเป็น 4 สิทธิ ดังนี้

1. ผู้จัดการโปรเจค (Project Manager)

เป็นผู้ควบคุมทุกขั้นตอนในการทำโปรเจครวมถึงขั้นตอนการทดสอบ ซึ่งจะเป็นผู้ที่กำหนดขอบเขตของการทดสอบโดยรับคำสั่งจากลูกค้า ส่งผลให้สามารถเข้าตรวจสอบ แก้ไข เปลี่ยนแปลง ได้ในทุกฟังก์ชันของระบบ

2. ผู้จัดการการทดสอบ (Test Manager)

เป็นหัวหน้าทีมในการทดสอบซอฟต์แวร์ ซึ่งจะมีสิทธิการใช้งานเหมือนกับผู้จัดการโปรเจค ส่งผลให้สามารถเข้าตรวจสอบ แก้ไข เปลี่ยนแปลง ได้ในทุกฟังก์ชันของระบบ แต่จะไม่สามารถเปลี่ยนแปลงขอบเขตคำสั่งของลูกค้าได้หากต้องการแก้ไขจะต้องให้ผู้จัดการโปรเจคแก้ไข

3. สำหรับทีมการทดสอบ (Test Team)

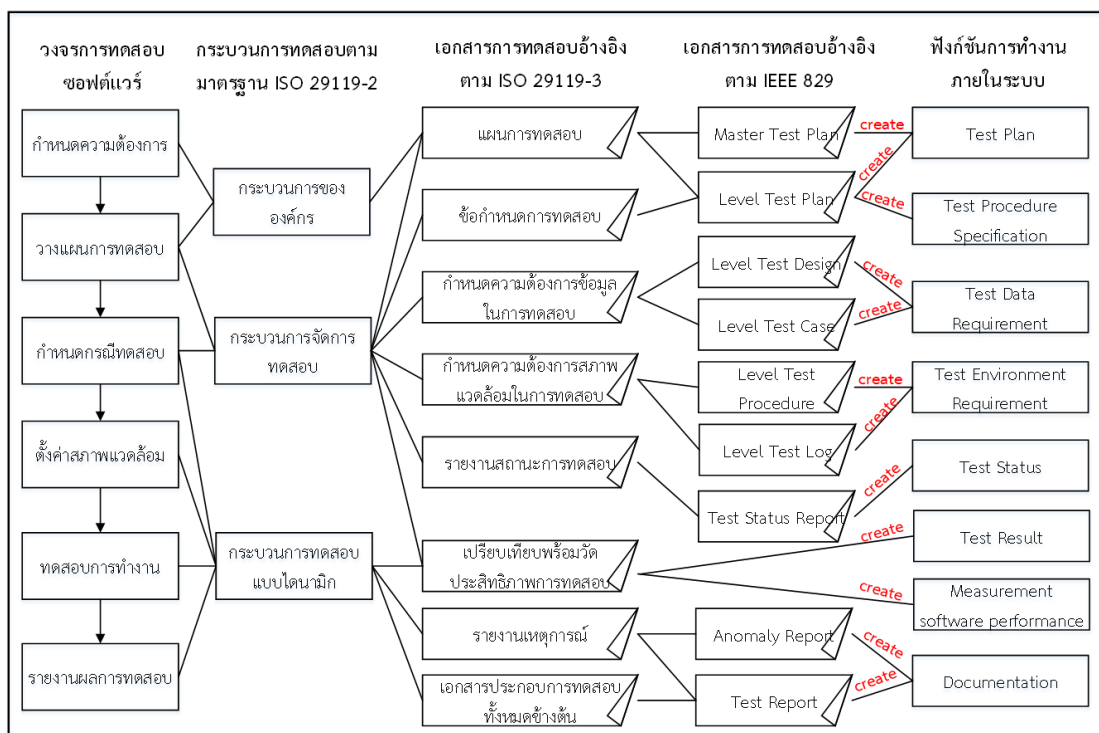
เป็นทีมในการทดสอบซอฟต์แวร์ มีหน้าที่ในการทดสอบตามขั้นตอนของระบบ ส่งผลให้สามารถแก้ไขได้เฉพาะบางฟังก์ชัน โดยไม่สามารถเปลี่ยนแปลงหรือแก้ไขขอบเขตคำสั่งของลูกค้าได้

4. สำหรับทีมพัฒนาระบบ (Development Team)

เป็นทีมในการพัฒนาซอฟต์แวร์ มีหน้าที่หลักคือพัฒนาซอฟต์แวร์ แต่จำเป็นต้องมีส่วนในการทดสอบซอฟต์แวร์บางส่วน ส่งผลให้สามารถแก้ไขได้เฉพาะบางฟังก์ชัน โดยไม่สามารถเปลี่ยนแปลงหรือแก้ไขขอบเขตคำสั่งของลูกค้าได้

3.2.2 สร้างความสัมพันธ์ระหว่างขั้นตอนการทดสอบและมาตรฐานเอกสารการทดสอบ

การสร้างระบบเตรียมแผนการทดสอบนั้นจำเป็นต้องสร้างจากมาตรฐานการทดสอบเพื่อควบคุมมาตรฐานของเอกสารที่สร้างขึ้น และต้องอ้างอิงจากกระบวนการทดสอบที่เป็นสากลที่ผู้คนนิยมใช้งานเพื่อให้สามารถนำไปปรับใช้ได้ง่ายเป็นพื้นฐานของเอกสารการทดสอบ ซึ่งขั้นตอนการสร้างความสัมพันธ์ระหว่างขั้นตอนในการทดสอบซอฟต์แวร์และมาตรฐานในการทดสอบซอฟต์แวร์ประกอบด้วย ขั้นตอนการทดสอบซอฟต์แวร์ มาตรฐานสากลในการทดสอบซอฟต์แวร์ ISO 29119 และมาตรฐานเอกสารการทดสอบ IEEE 829 แสดงดังรูปที่ 3.3



รูปที่ 3.3 แผนภาพความสัมพันธ์ระหว่างขั้นตอนการทดสอบ มาตรฐานการทดสอบ และมาตรฐานเอกสารการทดสอบ

จากรูปที่ 3.3 สามารถอธิบายรายละเอียดในแต่ละกระบวนการได้ดังนี้

1. วงจรการทดสอบซอฟต์แวร์ (Software Testing Life Cycle)

การศึกษามาตรฐานในการทดสอบซอฟต์แวร์และหาความสัมพันธ์กับขั้นตอนการทดสอบซอฟต์แวร์ที่มีอยู่โดยทั่วไป เพื่อให้ได้กระบวนการในการทดสอบซอฟต์แวร์เป็นไปตามมาตรฐานสากล ครอบคลุมทุกขั้นตอนของการทดสอบและสามารถนำไปใช้งานได้จริง โดยขั้นตอนในวงจรการทดสอบซอฟต์แวร์ประกอบด้วย 6 ขั้นตอน คือ กำหนดความต้องการในการทดสอบ วางแผนในการทดสอบ กำหนดกรณีทดสอบต่าง ๆ ที่ต้องการทดสอบ ตั้งค่าสภาพแวดล้อมที่ต้องการทดสอบ ทดสอบการทำงานของซอฟต์แวร์ และรายงานผลการทดสอบซอฟต์แวร์

2. มาตรฐานสากลในการทดสอบซอฟต์แวร์ (Software Testing Life Cycle)

มาตรฐานในการทดสอบซอฟต์แวร์ที่นำมาปรับใช้ในงานวิจัยคือ ISO 29119 ประกอบด้วย 5 ส่วน คือ แนวความคิดและการนิยาม กระบวนการทดสอบ เอกสารการทดสอบ เทคนิคในการทดสอบ และการทดสอบด้วยคำหลัก ซึ่งจะใช้ส่วนที่ 2 กระบวนการทดสอบ และส่วนที่ 3 เอกสารการทดสอบ โดยที่ส่วนที่ 2 (ISO 29119-2) กระบวนการทดสอบ ประกอบด้วย กระบวนการทดสอบองค์กร กระบวนการจัดการการทดสอบ และกระบวนการทดสอบแบบไดนามิก ส่วนที่ 3 (ISO 29119-3) เอกสารการทดสอบซึ่งประกอบด้วย แผนการทดสอบ ข้อกำหนดการทดสอบ กำหนดความต้องการข้อมูลในการทดสอบ กำหนดความต้องการสภาพแวดล้อมในการทดสอบ รายงานสถานการณ์ทดสอบ เปรียบเทียบพร้อมวัดประสิทธิภาพการทดสอบ รายงานเหตุการณ์ และเอกสารประกอบการทดสอบทั้งหมดที่พร้อมในการพิมพ์

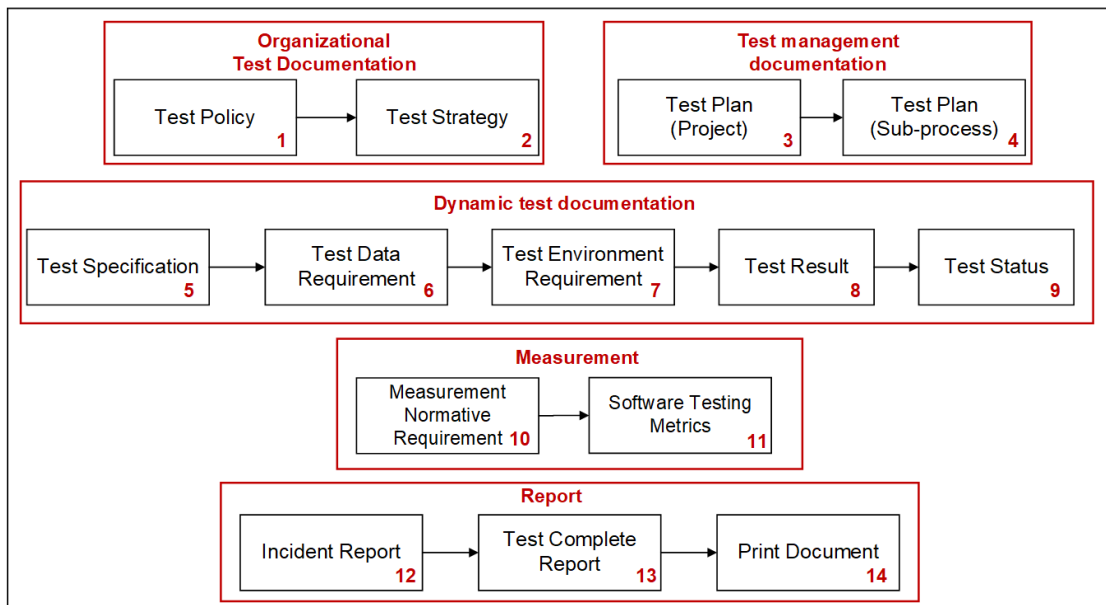
3. มาตรฐานเอกสารทดสอบซอฟต์แวร์ IEEE 829

การอ้างอิงมาตรฐานเอกสาร IEEE 829 เพื่อตรวจสอบเอกสารที่จำเป็นในการทดสอบซอฟต์แวร์ โดยในมาตรฐานประกอบด้วยเอกสาร 8 ประเภท คือ master test plan, level test plan, level test design, level test case, level test procedure, level test log, anomaly report และ test report

3.2.3 การแบ่งกลุ่มการดำเนินงาน และรายละเอียดของแต่ละฟังก์ชัน

จากการหาความสัมพันธ์ระหว่างมาตรฐาน ISO 29119-2, ISO 29119-3, IEEE 829 และวงจรการทดสอบ ในข้างต้นสามารถสร้างฟังก์ชันสำหรับการเตรียมเอกสารการทดสอบให้ครอบคลุมทั้งหมด โดยมีการแบ่งกลุ่มการดำเนินงานออกเป็น 5 กลุ่ม คือ Organizational Test Process Documentation, Test Management Processes Documentation, Dynamic Test Process Documentation, Measurement Performance และ Documentation เพื่อให้ครอบคลุมทุกกระบวนการในการทดสอบ ดังรูปที่ 3.5 ซึ่งประกอบด้วยฟังก์ชัน Test policy, Test strategy, Test plan, Test status, Test complete, Test case, Test procedure specification, Test data

requirement, Test environment requirement, Test result, Measurement software performance และ documentation



รูปที่ 3.4 แผนภาพแสดงการแบ่งกลุ่มตามมาตรฐาน ISO 29119

จากรูปที่ 3.4 สามารถอธิบายรายละเอียดในส่วนต่าง ๆ ใน ดังนี้

1. เอกสารขั้น ตอนการทดสอบขององค์กร (Organizational Test Process Documentation)

ในส่วนนี้เกี่ยวข้องกับการสร้างนโยบายการทดสอบ และกลยุทธ์การทดสอบที่เกี่ยวข้องกับองค์กรซึ่งประกอบด้วย 2 ฟังก์ชันคือ

1. ฟังก์ชัน Test policy เป็นส่วนที่ใช้เพื่อสร้างนโยบายแผนการทดสอบขององค์กร
2. ฟังก์ชัน Test strategy เป็นส่วนที่ใช้เพื่อสร้างแผนกลยุทธ์การทดสอบขององค์กร

2. เอกสารจัดการกระบวนการทดสอบ (Test Management Processes Documentation)

ในส่วนนี้เกี่ยวข้องกับการสร้างแผนการทดสอบซอฟต์แวร์ ประกอบด้วย 3 ฟังก์ชัน คือ

1. ฟังก์ชัน Test plan เป็นส่วนในการสร้างแผนการทดสอบซอฟต์แวร์ ซึ่งแบ่งออกเป็น 2 ประเภท คือ Project test plan และ System test plan แต่ละประเภทจะมีข้อกำหนดการระบุที่ต่างกัน โดย Project test plan จะครอบคลุม System test plan โดยใน 1 Project test plan สามารถมีหลาย system test plan ได้

2. ฟังก์ชัน Test status เป็นส่วนในการกำหนดสถานะของการทดสอบ โดยฟังก์ชันนี้จะอยู่ในฟังก์ชัน Test Result แต่จัดอยู่ในกลุ่มของการจัดการกระบวนการทดสอบ

3. ฟังก์ชัน Test complete specification เป็นส่วนในการระบุการทดสอบที่ผ่านการทดสอบสมบูรณ์ ซึ่งฟังก์ชันนี้จะสัมพันธ์กับฟังก์ชัน Test complete documentation ในส่วน Documentation

3. เอกสารกระบวนการทดสอบแบบไดนามิก (Dynamic Test Processes Documentation)

ในส่วนนี้เกี่ยวข้องกับกระบวนการทดสอบแบบไดนามิกประกอบด้วย 5 ฟังก์ชัน คือ

1. ฟังก์ชัน Test procedure specification เกี่ยวข้องกับข้อกำหนดการทดสอบรายละเอียดในการทดสอบ

2. ฟังก์ชัน Test case specification เกี่ยวข้องกับการกำหนดกรณีการทดสอบ

3. ฟังก์ชัน Test data requirement เกี่ยวข้องกับการกำหนดความต้องการของข้อมูลในการทดสอบ

4. ฟังก์ชัน Test environment requirement เกี่ยวข้องกับการกำหนดความต้องการของสภาพแวดล้อมที่ใช้ในการทดสอบ

5. ฟังก์ชัน Test result เกี่ยวข้องกับการเปรียบเทียบผลการทดสอบ การเลือกเมทริกซ์ที่ใช้ในการวัดประสิทธิภาพซึ่งสัมพันธ์กับฟังก์ชัน Measurement การระบุสถานะการทดสอบซึ่งสัมพันธ์กับฟังก์ชัน Test status

6. ฟังก์ชัน Test incident specification เป็นส่วนในการระบุปัญหาเมื่อทดสอบซอฟต์แวร์ไม่สมบูรณ์

4. การวัดประสิทธิภาพ (Measurement Performance)

เป็นขั้นตอนในการวัดประสิทธิภาพของซอฟต์แวร์เมื่อทำการทดสอบซอฟต์แวร์เสร็จสิ้นในแต่ละรอบ ซึ่งภายในงานวิจัยแบ่งส่วนการวัดประสิทธิภาพออกเป็น 2 ส่วน คือ การวัดประสิทธิภาพตามข้อกำหนดเชิงบรรทัดฐาน และการวัดประสิทธิภาพการทดสอบซอฟต์แวร์ ดังนี้

1) การวัดประสิทธิภาพข้อกำหนดเชิงบรรทัดฐาน คือ องค์ประกอบที่ได้ถูกกำหนดไว้ล่วงหน้าซึ่งจะต้องปฏิบัติตามเพื่อให้เป็นไปตามความต้องการของลูกค้า หรือ องค์กรณ์นั้น ๆ ซึ่งสามารถปรับใช้ได้ขององค์กรที่ต้องได้รับการแก้ไข โดยภายในงานวิจัยมีการวัดประสิทธิภาพนี้เพื่อให้ทราบถึงระดับความต้องการขององค์กรเพื่อจะได้แก้ไขให้มีประสิทธิภาพได้ องค์ประกอบของข้อกำหนดเชิงบรรทัดฐานแบ่งออกเป็น 3 ส่วนดังนี้

Shall เป็นการระบุว่าจะต้องปฏิบัติตามข้อกำหนดนั้น ซึ่งภายในงานวิจัยกำหนดค่าความสำคัญอยู่ที่ร้อยละ 80

Should เป็นการระบุที่เป็นคำแนะนำว่าควรปฏิบัติข้อกำหนดนั้น ซึ่งภายในงานวิจัย กำหนดค่าความสำคัญอยู่ที่ร้อยละ 60

May เป็นการระบุที่เกี่ยวกับการอนุญาต ให้สิทธิ์ในการเลือกปฏิบัติตามข้อกำหนด หรือไม่ ซึ่งภายในงานวิจัยกำหนดค่าความสำคัญอยู่ที่ร้อยละ 40

2) การวัดประสิทธิภาพในการทดสอบซอฟต์แวร์ (Software testing metrics) การวัดประสิทธิภาพในการทดสอบซอฟต์แวร์จะทำการวัดทั้งหมด 8 ข้อ โดยมีสูตรการคำนวณ คือ

2.1 Test Case Executed ใช้สำหรับการตรวจสอบสถานะของกรณีทดสอบ (Test Case) ที่ได้นำไปทดสอบ โดยคำนวณจากข้อมูลของกรณีทดสอบที่ได้ดำเนินการทดสอบ และ จำนวนกรณีทดสอบที่ถูกสร้างทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Executed} = (\text{No. of Test cases executed} / \text{Total no. of Test cases written}) * 100$$

2.2 Test Case Not Executed ใช้สำหรับการตรวจสอบสถานะของกรณีทดสอบ (Test Case) ที่รอดำเนินการทดสอบ โดยคำนวณจากข้อมูลของกรณีทดสอบที่ยังไม่ได้ดำเนินการทดสอบ และ จำนวนกรณีทดสอบที่ถูกสร้างทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases not executed} = (\text{No. of Test cases not executed} / \text{Total no. of Test cases written}) * 100$$

2.3 Test Cases Passed ใช้สำหรับการคำนวณกรณีทดสอบ (Test Case) ที่ดำเนินการทดสอบผ่าน โดยคำนวณจากข้อมูลของจำนวนกรณีทดสอบที่ทดสอบผ่าน และ จำนวนกรณีทดสอบที่ดำเนินการทดสอบทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Passed} = (\text{No. of Test cases Passed} / \text{Total no. of Test cases Executed}) * 100$$

2.4 Test Cases Failed ใช้สำหรับการคำนวณกรณีทดสอบ (Test Case) ที่ดำเนินการทดสอบไม่ผ่าน โดยคำนวณจากข้อมูลของจำนวนกรณีทดสอบที่ทดสอบไม่ผ่าน และ จำนวนของกรณีทดสอบที่ดำเนินการทดสอบทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Failed} = (\text{No. of Test cases Failed} / \text{Total no. of Test cases Executed}) * 10$$

2.5 Test Cases Blocked ใช้สำหรับการคำนวณกรณีทดสอบ (Test Case) ที่ถูกบล็อก ไม่ให้ทดสอบ โดยคำนวณจากข้อมูลกรณีทดสอบที่ถูกบล็อก และ จำนวนของกรณีทดสอบที่ดำเนินการทดสอบทั้งหมด แสดงดังสูตรต่อไปนี้

$$\text{Test cases Blocked} = (\text{No. of Test cases Blocked} / \text{Total no. of Test cases Executed}) * 100$$

2.6 จำนวนที่นำกรณีทดสอบมาทดสอบซ้ำ (number of re-executed test cases) เมื่อการทดสอบมีปัญหาจำเป็นต้องทำการทดสอบซ้ำ การคำนวณปริมาณกรณีทดสอบที่นำมาทดสอบซ้ำ แสดงให้เห็นถึงประสิทธิภาพของกรณีทดสอบ แสดงดังสูตรต่อไปนี้

number of re-executed test cases = (จำนวนกรณีทดสอบที่มีจำนวนครั้งในการทดสอบมากกว่า 1 ครั้ง/จำนวนกรณีทดสอบทั้งหมด)

2.7 จำนวนเหตุการณ์ที่เกิดขึ้น (number of incidents per category) แสดงให้เห็นถึงกระบวนการทดสอบที่ไม่ครอบคลุมในกรณีทดสอบ แสดงดังสูตรต่อไปนี้

number of incidents per category = (จำนวนเหตุการณ์ที่พบในกรณีทดสอบ/จำนวนกรณีทดสอบที่ถูกทดสอบ)

2.8 จำนวนปัญหาที่ได้รับการแก้ไขอย่างสมบูรณ์ (number of solved incidents per category) แสดงถึงประสิทธิภาพของแผนการทดสอบ ด้วยการคำนวณจำนวนกรณีทดสอบที่ได้ทำการแก้ไข แสดงดังสูตรต่อไปนี้

number of solved incidents per category = (จำนวนกรณีทดสอบที่แก้ไข/จำนวนกรณีทดสอบที่พบปัญหา)

5. เอกสารการทดสอบ (Documentation)

เป็นขั้นตอนสุดท้ายของระบบที่เกี่ยวข้องกับเอกสารการทดสอบซอฟต์แวร์ ซึ่งประกอบด้วย เอกสารที่รายงานเหตุการณ์ที่เกิดขึ้นระหว่างการทดสอบ คือ ฟังก์ชัน test incident report, ฟังก์ชัน test complete report ฟังก์ชัน test status คือ รายงานสถานการณ์ทดสอบ ประกอบด้วย 3 ฟังก์ชัน คือ

1. ฟังก์ชัน Test documentation เป็นส่วนที่แสดงเอกสารการทดสอบ ประกอบด้วย Test Plan (Organization Specification และ Test Case)

2. ฟังก์ชัน Test complete report เป็นเอกสารรายงานถึงสถานการณ์ทดสอบที่สมบูรณ์ โดยอาศัยข้อมูลจากฟังก์ชัน Test complete specification

3. ฟังก์ชัน Test incident report เป็นเอกสารรายงานถึงเหตุการณ์ระหว่างการทดสอบ ในกรณีที่การทดสอบไม่สมบูรณ์ ซึ่งจำเป็นต้องได้รับการแก้ไขโดยอาศัยเอกสารนี้เพื่อรายงานสิ่งที่เกิดขึ้น

3.3 การออกแบบระบบทดสอบซอฟต์แวร์

ในขั้นตอนการพัฒนาโปรแกรมประกอบด้วย การออกแบบระบบที่ให้ทราบถึงภาพรวมการทำงานของระบบ ตั้งแต่สร้างความสัมพันธ์ระหว่างโมดูลเพื่อหาทิศทางการทำงาน และการส่งออกของข้อมูลในแต่ละโมดูล การระบุผู้ที่มีส่วนเกี่ยวข้องในแต่ละฟังก์ชันของระบบ และความสัมพันธ์ของข้อมูลที่แสดงด้วยแผนงานฐานข้อมูล แสดงดังนี้

3.3.1 แผนภาพการไหลของข้อมูล (Data Flow Diagram)

แสดงทิศทางการไหลของข้อมูล สิทธิการเข้าถึงในแต่ละฟังก์ชัน ความสัมพันธ์ของข้อมูล (Data Store) โดยแบ่งระดับ (Level) การแสดงออกเป็น 3 ระดับ คือ Context Diagram, Data Flow Diagram Level 1 และ Data Flow Diagram Level 2 แสดงดังนี้

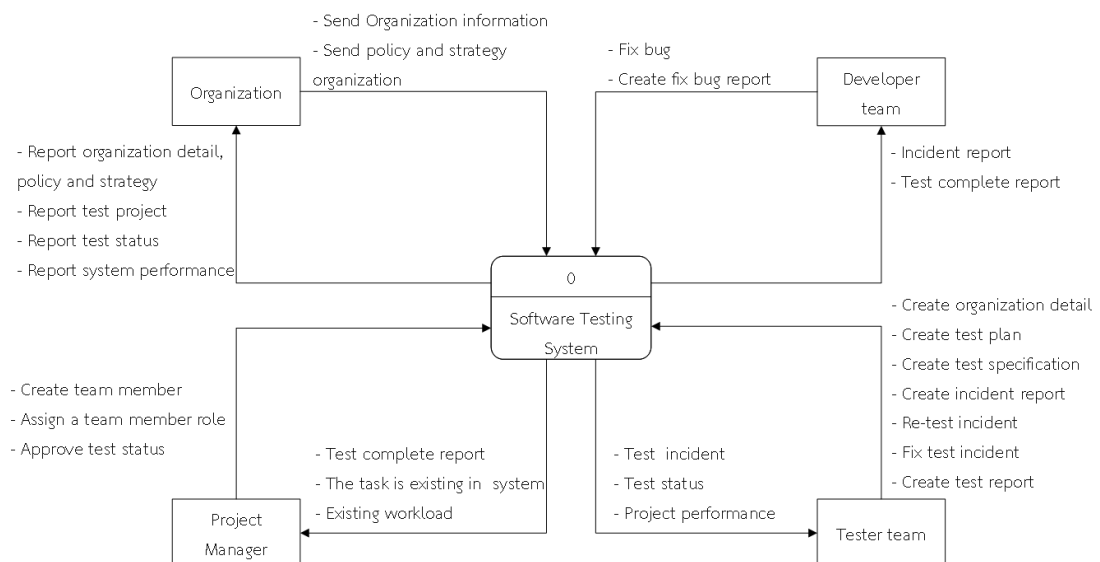
1. แผนภาพการไหลของข้อมูล (Context Diagram) หรือ Data Flow Diagram Level 0 แสดงภาพรวมการไหลของข้อมูลที่เกี่ยวข้องกับบุคคล ซึ่งระบบมีผู้เกี่ยวข้อง 4 ส่วน แสดงดังรูปที่ 3.5

Organization ที่เป็นตัวแทนขององค์กร มีหน้าที่ในการส่งข้อมูล ตรวจสอบข้อมูลต่าง ๆ ขององค์กร ควบคุมถึงขอบเขต และความต้องการขององค์กร

Project Manager (PM) ผู้จัดการโครงการเป็นผู้ที่มีอำนาจสูงสุดในแต่ละโปรเจกต์ที่ได้รับมอบหมาย ซึ่งมีหน้าที่ คือ กำหนดสมาชิกที่มีส่วนเกี่ยวข้องในแต่ละโปรเจกต์ พร้อมกำหนดหน้าที่ในโปรเจกต์ และเป็นผู้ที่ทำหน้าที่อนุมัติ ตรวจสอบแผนการทดสอบ แผนกรณีทดสอบ หรือเหตุการณ์ที่เกิดขึ้นระหว่างการทดสอบโปรเจกต์

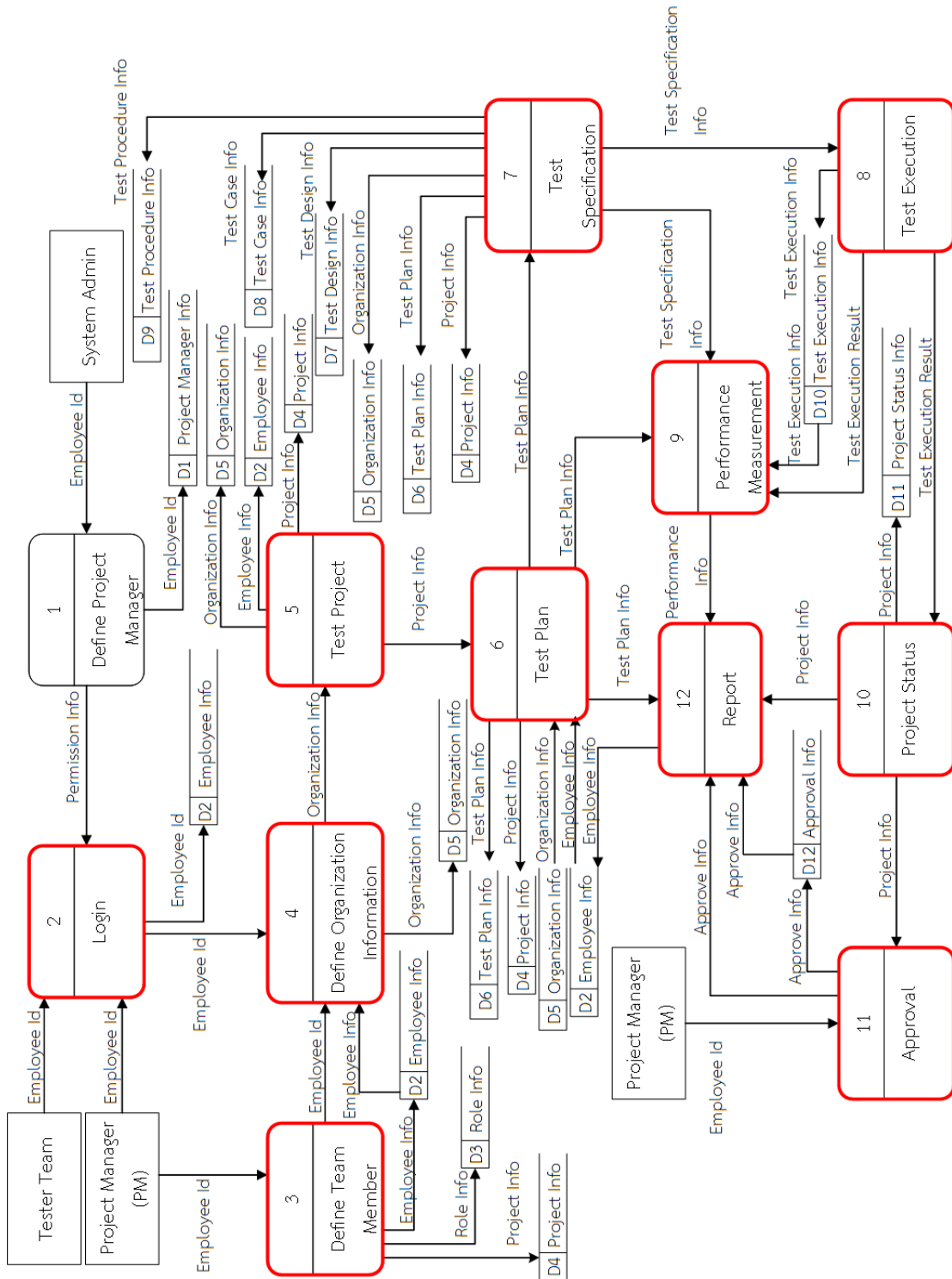
Tester Team ทีมทดสอบ ทำหน้าที่ทดสอบโปรเจกต์ที่ได้รับมอบหมาย พร้อมตรวจสอบและคำนึงถึงคุณภาพของการทดสอบ

Developer Team ทีมพัฒนาระบบ มีส่วนเกี่ยวข้องในกระบวนการทดสอบโดยรับคำสั่งในการแก้ไขข้อผิดพลาดที่ทีมทดสอบพบระหว่างการทดสอบ



รูปที่ 3.5 แผนภาพการไหลของข้อมูล Context Diagram

2. แผนภาพการไหลของข้อมูลด้วย Data Flow Diagram Level 1: DFD



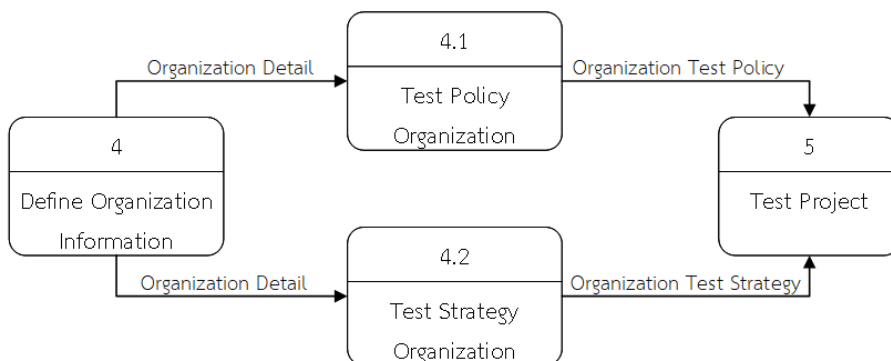
รูปที่ 3.6 แผนภาพการไหลของข้อมูล DFD Level 1

การออกแบบแผนภาพกระแสข้อมูล DFD Level 1 ประกอบด้วย Define Project Manager Process, Login Process, Define Team Member Process, Define Organization Information Process, Test Project Process, Test Plan Process, Test Specification Process, Test Execution Process, Performance Measurement Process, Project Status Process, Approval Process และ Report Process โดยภายในงานวิจัยนี้ได้ดำเนินในกระบวนการที่วงสีแดงไว้เท่านั้น ซึ่งมีกระบวนการดำเนินการดังนี้ Define Project Manager Process เป็นกระบวนการในการสร้างบัญชีผู้ใช้งานระบบ เพื่อกำหนดสิทธิในชั้นผู้จัดการโปรเจกต์ ซึ่งกระบวนการนี้ถูกสร้างโดยผู้ดูแลระบบ ส่งข้อมูลไปยังกระบวนการต่อไป คือ Login Process เป็นกระบวนการเข้าสู่ระบบ ซึ่งอาศัยสิทธิการใช้งานจากกระบวนการเข้าสู่ระบบ และกระบวนการกำหนดสมาชิกในทีมทดสอบ โดยผู้จัดการโครงการต้องเข้าสู่ระบบก่อนเพื่อกำหนดทีม และสมาชิกในทีมทดสอบ เพื่อให้สมาชิกสามารถเข้าใช้งานระบบได้ โดยต้องใช้กระบวนการ Define Team Member Process ในการกำหนดทีม และสมาชิกในทีม เมื่อกำหนดสิทธิ และทีมในการดำเนินการเสร็จเรียบร้อยแล้ว ในกระบวนการ Define Organization Information Process เป็นกระบวนการสร้างองค์กร หรือ ลูกค้า ที่ต้องการทดสอบระบบ เมื่อสร้างข้อมูลเรียบร้อยแล้ว ส่งข้อมูลไปยังกระบวนการ Test Project Process เป็นกระบวนการโปรเจกต์ทดสอบ เพื่อกำหนดข้อมูลโปรเจกต์ สกัดความต้องการ กำหนดโมดูล กระบวนการต่อไป คือ Test Plan Process เป็นกระบวนการแผนการทดสอบ กำหนดฟังก์ชันในการทดสอบ กระบวนการต่อไป คือ Test Specification Process เป็นกระบวนการในการระบุกรณีทดสอบ เงื่อนไขการทดสอบ เมื่อสร้างแผนการทดสอบ กรณีทดสอบเสร็จสิ้นแล้ว กระบวนการต่อไป คือ Test Execution Process เป็นกระบวนการดำเนินการทดสอบ โดยนำแผนการทดสอบ กรณีทดสอบ ที่สร้างไว้ไปทดสอบตามแผนที่ได้สร้างไว้ กระบวนการต่อไป คือ Performance Measurement Process เป็นกระบวนการวัดประสิทธิภาพของโปรเจกต์ที่ได้สร้างไว้ Project Status Process เป็นกระบวนการแสดงสถานะของโปรเจกต์ที่ได้ทดสอบมา Approval Process เป็นกระบวนการอนุมัติแผนการทดสอบ กรณีทดสอบ และตรวจสอบผลการทดสอบ และกระบวนการสุดท้าย คือ Report Process เป็นกระบวนการสร้างรายงาน เอกสารต่าง ๆ

3. การออกแบบแผนภาพกระแสข้อมูล Data Flow Diagram: DFD Level 2 ประกอบด้วยกระบวนการในระดับที่ 2 ดังนี้

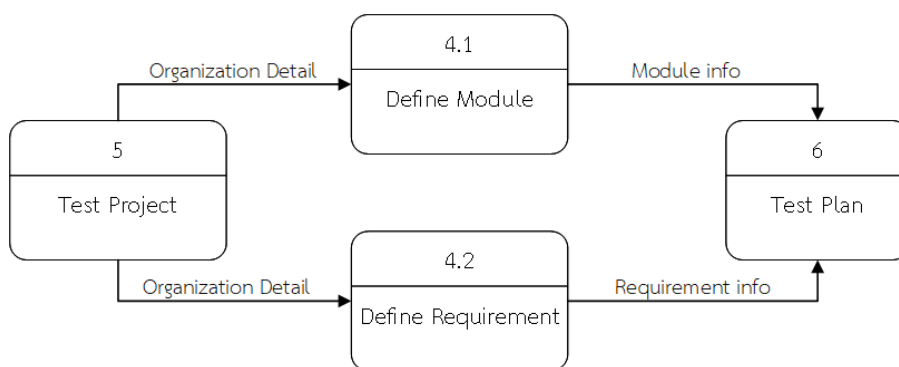
1) การออกแบบแผนภาพกระแสข้อมูล DFD Level 2 : ในกระบวนการ Define Organization Information ประกอบด้วยกระบวนการ Test Policy Organization กระบวนการในการกำหนดนโยบายการทดสอบองค์กรที่องค์กรได้กำหนดไว้ และทีมทดสอบต้องอ้างอิงการดำเนินการตามนโยบายที่ได้กำหนดไว้ และ Test Strategy Organization กระบวนการกำหนดกลยุทธ์การทดสอบองค์กรได้กำหนดวิธีในการแก้ปัญหา เงื่อนไข ต่าง ๆ ในการทดสอบ ซึ่งทีมทดสอบ

หรือโปรเจค ต้องดำเนินการตามที่ได้ระบุ ซึ่งกระบวนการ Define Organization Information ในระดับที่ 2 แสดงดังรูปที่ 3.7



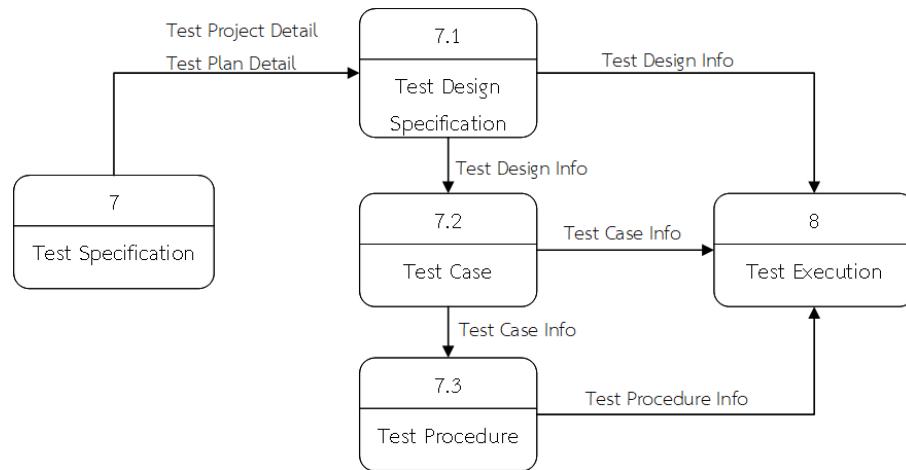
รูปที่ 3.7 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการกำหนดข้อมูลองค์กร

2) การออกแบบแผนภาพกระแสข้อมูล DFD Level 2 : ในกระบวนการ Test Project Process ประกอบด้วยกระบวนการย่อย 2 กระบวนการ คือ Define Module กำหนดโมดูล พร้อมฟังก์ชันที่ต้องการทดสอบ และ Define Requirement กำหนดความต้องการที่ได้รับมาจากลูกค้า นำมาสกัดโดยจับคู่กับฟังก์ชันที่ต้องการทดสอบ แล้วส่งข้อมูลไปยังกระบวนการแผนการทดสอบในลำดับต่อไป ซึ่งกระบวนการ Test Project Process ในระดับที่ 2 แสดงดังรูปที่ 3.8



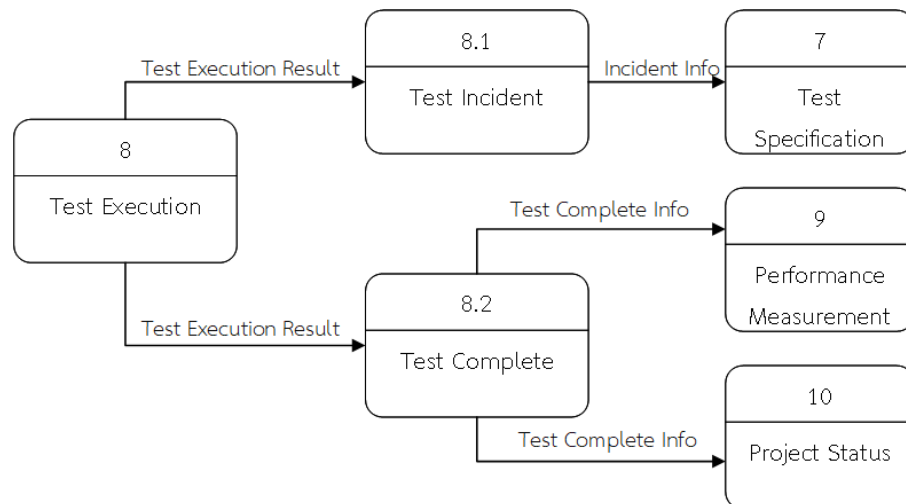
รูปที่ 3.8 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการโปรเจคทดสอบ

3) การออกแบบแผนภาพกระแสข้อมูล DFD Level 2 : ในกระบวนการ Test Specification ประกอบด้วยกระบวนการย่อย 3 กระบวนการ คือ Test Design Specification กำหนดเงื่อนไขการทดสอบ (Test Condition) Test Case กำหนดกรณีต่าง ๆ ที่สกัดออกมาจากเงื่อนไขการทดสอบ และ Test Procedure ขั้นตอนการทดสอบ โดยนำกรณีทดสอบมาสร้างเป็นขั้นตอนในการทดสอบ พร้อมทั้งระบุผลลัพธ์ที่คาดหวัง (Expected Result) และ ผลลัพธ์ที่แท้จริง (Actual Result) ซึ่งกระบวนการ Test Specification ในระดับที่ 2 แสดงดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการข้อกำหนดการทดสอบ

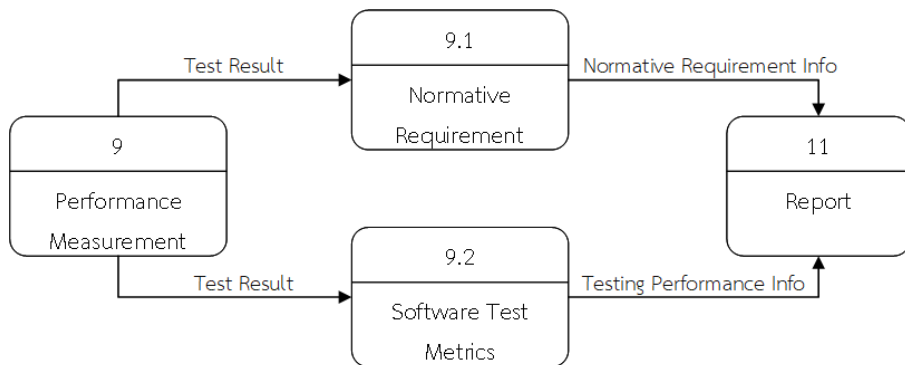
4) การออกแบบแผนภาพกระแสข้อมูล DFD Level 2 : ในกระบวนการ Test Execution ประกอบด้วยกระบวนการย่อย 2 กระบวนการ คือ Test Incident กระบวนการที่แสดงการทดสอบที่ไม่สมบูรณ์ เพื่อทำการรายงานสถานการณ์ที่เกิดขึ้นระหว่างขั้นตอนการทดสอบ พร้อมทั้งสร้างแผนเพื่อการทดสอบซ้ำ และ Test Complete เป็นกระบวนการที่ทดสอบสมบูรณ์ แต่ก่อนที่จะส่งไปยังกระบวนการอื่น ๆ เพื่อนำไปใช้งาน จำเป็นต้องรอการอนุมัติผลการทดสอบจากผู้จัดการโปรเจกต์ก่อน (Project Manager) ซึ่งกระบวนการ Test Execution ในระดับที่ 2 แสดงดังรูปที่ 3.10



รูปที่ 3.10 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการดำเนินการทดสอบ

5) การออกแบบแผนภาพกระแสข้อมูล DFD Level 2 : ในกระบวนการ Performance Measurement ประกอบด้วยกระบวนการย่อย 2 กระบวนการ คือ Normative Requirement เป็นกระบวนการคำนวณประสิทธิภาพที่จำเป็น ซึ่งอ้างอิงข้อมูลตามมาตรการทดสอบสากลที่ได้ระบุค่าความจำเป็นไว้ และ Software Test Performance กระบวนการคำนวณประสิทธิภาพของ

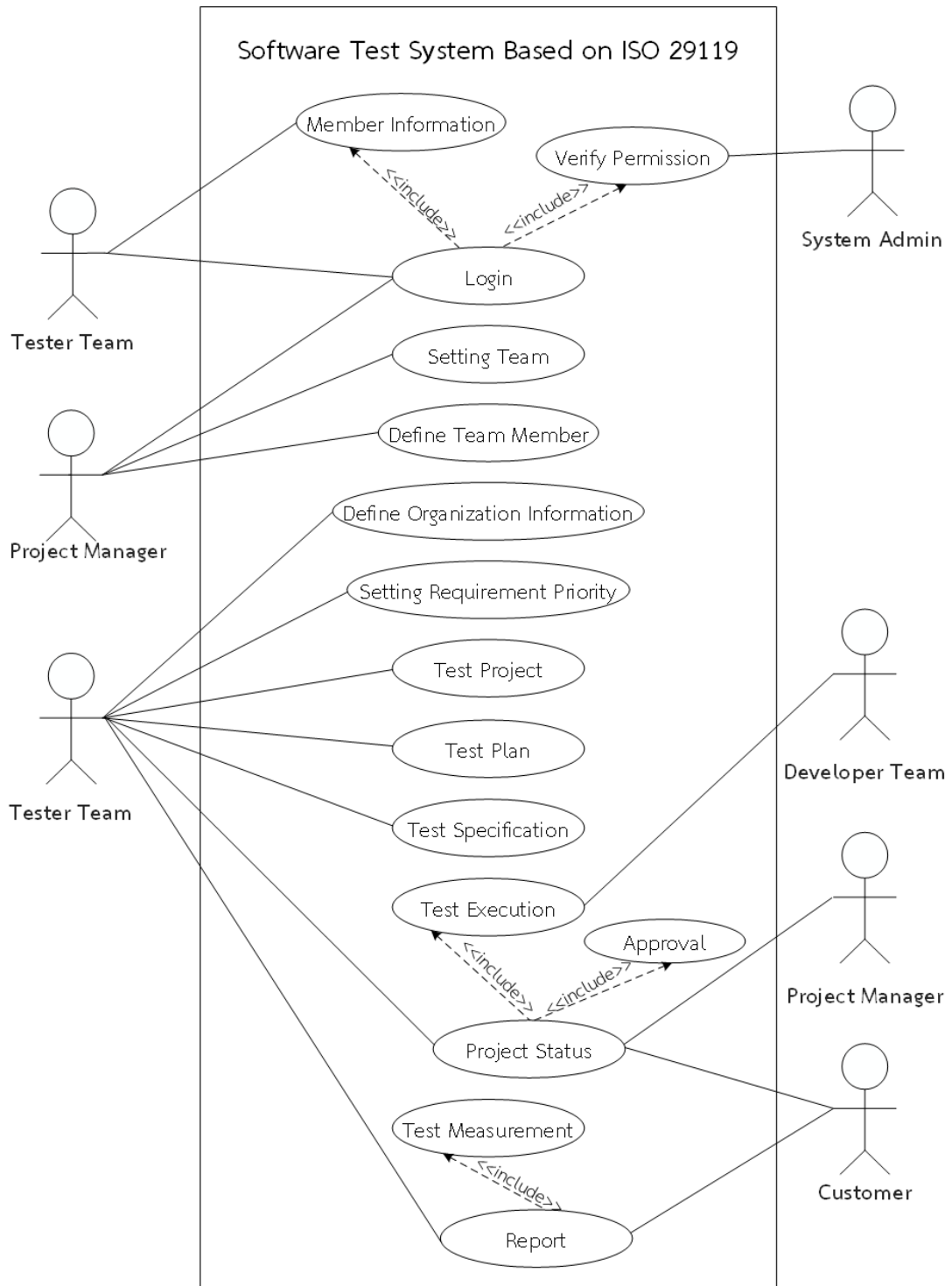
ซอฟต์แวร์ หรือ โปรเจคที่ทำการทดสอบ เพื่อให้ทราบถึงประสิทธิภาพในการทดสอบ พร้อมได้ทำการปรับปรุงแก้ไขได้ทันที ซึ่งกระบวนการ Performance Measurement ในระดับที่ 2 แสดงดังรูปที่ 3.11



รูปที่ 3.11 แผนภาพการไหลของข้อมูล DFD Level 2 ของกระบวนการวัดประสิทธิภาพ

3.3.2 แผนภาพยูสเคส (Use Case Diagram)

จากแผนภาพแสดงการไหลของข้อมูลข้างต้น ผู้จัดทำได้เขียนแผนภาพยูสเคสเพื่อแสดงการทำงานของระบบ ภายในระบบทดสอบซอฟต์แวร์ตามมาตรฐานการทดสอบนั้น ประกอบด้วยฟังก์ชันดังนี้ Login, Define Team Member, Define Organizational Information, Test Project, Test Plan, Test Specification, Test Execution, Project Status, Performance Measurement และ Report มีผู้เกี่ยวข้องซึ่งกับกระบวนการทำงานของระบบประกอบด้วย Tester Team, Project Manager และ Customer แสดงดังรูปที่ 3.12



รูปที่ 3.12 แผนภาพยูสเคส

1. คำอธิบายยูสเคส

ตารางที่ 3.1 คำอธิบายการกำหนดข้อมูลสมาชิก (Member Information)

หัวข้อ	คำอธิบาย
Use Case Name	Member Information
Scenario	เก็บข้อมูลสมาชิก ผู้มีส่วนในการเข้าใช้ระบบเท่านั้น
Description	เป็นฐานข้อมูลของสมาชิกในระบบ
Trigger	ต้องการเก็บข้อมูลสมาชิก
Actor	ทีมทดสอบ (Tester Team) ผู้จัดการโปรเจค (Project Manager: PM)
Pre-Condition	เก็บข้อมูลสมาชิก ผู้มีส่วนเข้าใช้ระบบ
Post-Condition	เก็บข้อมูลได้
Flow	1. บันทึกข้อมูลสมาชิกลงฐานข้อมูลได้ 2. ดูข้อมูลสมาชิกได้

ตารางที่ 3.2 คำอธิบายการตรวจสอบสิทธิการใช้งานระบบ (Verity Permission)

หัวข้อ	คำอธิบาย
Use Case Name	Verify Permission
Scenario	ผู้ดูแลระบบทำการสร้างผู้ใช้งานในระดับผู้จัดการโปรเจคเท่านั้น (Project Manager)
Description	เป็นขั้นตอนที่ทำโดยวิธี Back-end โดยผู้ดูแลระบบ ทำการสร้าง username, password ให้กับผู้ใช้งาน
Trigger	ผู้ใช้ต้องการสร้างผู้ใช้งานระบบ
Actor	ผู้ดูแลระบบ (System Admin)
Pre-Condition	ผู้ไม่เกี่ยวข้องจะไม่สามารถเข้าสู่ระบบได้
Post-Condition	ตรวจจับผู้ที่ไม่มสิทธิในการเข้าระบบ
Flow	1. กำหนดสิทธิการใช้งานในแต่ละฟังก์ชัน 2. สร้าง username, password

ตารางที่ 3.3 คำอธิบายการเข้าสู่ระบบ (Login)

หัวข้อ	คำอธิบาย
Use Case Name	Login
Scenario	ต้องการเข้าสู่ระบบ โดยมีการควบคุมสิทธิ์ในการเข้าใช้ระบบ
Description	ต้องการเข้าสู่ระบบ โดยสุ่มรหัส
Trigger	ผู้ใช้งานต้องการสร้างหน้าการเข้าสู่ระบบที่มีการตรวจสอบสิทธิ์การใช้งาน
Actor	ผู้จัดการโปรเจก (Project Manager: PM) ทีมทดสอบ (Tester Team)
Pre-Condition	เข้าในสิทธิ์การใช้งานที่ต่างกัน ฟังก์ชันการใช้งานต่างกัน
Post-Condition	แสดงฟังก์ชันการใช้งานที่ต่างกัน
Flow	<ol style="list-style-type: none"> 1. ได้รับ username, password 2. ทำการเข้าสู่ระบบโดยการใส่ username, password 3. หากข้อมูลถูกต้อง เข้าสู่ระบบสำเร็จ 4. หากข้อมูลไม่ถูกต้อง ไม่สามารถเข้าสู่ระบบได้

ตารางที่ 3.4 คำอธิบายการตั้งค่าทีม (Setting Team)

หัวข้อ	คำอธิบาย
Use Case Name	การตั้งค่าทีม (Setting Team)
Scenario	กำหนดทีม
Description	กำหนดทีมเพื่อใช้ในการทำโปรเจก
Trigger	ต้องการกำหนดทีม ในการดำเนินงานโปรเจก
Actor	ผู้จัดการโปรเจก (Project Manager: PM)
Pre-Condition	กำหนดทีมในการดำเนินงานโปรเจก
Post-Condition	สร้างทีมในการดำเนินงานโปรเจก
Flow	<ol style="list-style-type: none"> 1. กำหนดทีมในการดำเนินงานโปรเจก 2. กำหนดหน้าที่ของทีม 3. กำหนดสมาชิกในทีม

ตารางที่ 3.5 คำอธิบายการกำหนดทีมในการทดสอบ (Define Team Member)

หัวข้อ	คำอธิบาย
Use Case Name	Define Team Member
Scenario	กำหนดกลุ่มของสมาชิก และสมาชิกในกลุ่ม
Description	เพื่อแบ่งกลุ่มการทำงานให้ชัดเจน เพื่อควบคุมการจ่ายงาน
Trigger	ต้องการกำหนดกลุ่ม ต้องการระบุสมาชิกในแต่ละกลุ่ม
Actor	ผู้จัดการโปรเจค (Project Manager: PM)
Pre-Condition	สร้างกลุ่มที่ทดสอบ ระบุสมาชิกในแต่ละกลุ่ม อย่างเท่าเทียม
Post-Condition	แบ่งกลุ่มได้อย่างเท่าเทียม
Flow	1. PM สร้างกลุ่มทีมทดสอบ 2. ระบุสมาชิกในแต่ละกลุ่ม 3. กำหนดโปรเจคงานให้แต่ละกลุ่ม

ตารางที่ 3.6 คำอธิบายการสร้างข้อมูลองค์กร (Define Organization Information)

หัวข้อ	คำอธิบาย
Use Case Name	Define Organization Information
Scenario	ทำการสร้างข้อมูลองค์กรที่เป็นลูกค้า
Description	สร้างข้อมูลองค์กร ระบุนโยบายขององค์กร ระบุกลยุทธ์ในการทดสอบขององค์กร
Trigger	ต้องการสร้างข้อมูลองค์กร
Actor	ทีมทดสอบ (Tester Team)
Pre-Condition	เก็บข้อมูลองค์กร เก็บนโยบายของแต่ละองค์กร เก็บกลยุทธ์ในการทดสอบของแต่ละองค์กร
Post-Condition	สร้างข้อมูลองค์กร กำหนดเงื่อนไขด้านนโยบาย และกลยุทธ์

หัวข้อ	คำอธิบาย
Flow	<ol style="list-style-type: none"> 1. สร้างข้อมูลลูกค้า หรือ องค์กร 2. สร้างนโยบาย และกลวิธีในการทดสอบ 3. นำไปปรับใช้กับโครงการทดสอบ

ตารางที่ 3.7 คำอธิบายการตั้งค่าลำดับความสำคัญของความต้องการ (Setting Requirement Priority)

หัวข้อ	คำอธิบาย
Use Case Name	ตั้งค่าลำดับความสำคัญของการกำหนดความต้องการ (Setting Requirement Priority)
Scenario	ต้องการกำหนดลำดับความต้องการ
Description	กำหนดลำดับความต้องการ โดยในแต่ละโปรเจกมีลำดับที่ต่างกัน
Trigger	ผู้ใช้งานระบบต้องกำหนดลำดับความต้องการ ในแต่ละโปรเจก
Actor	ทีมทดสอบ (Tester Team)
Pre-Condition	กำหนดลำดับความต้องการ
Post-Condition	นำลำดับที่ได้มาใช้ในโปรเจก
Flow	<ol style="list-style-type: none"> 1. กำหนดลำดับความต้องการ 2. นำลำดับที่กำหนดไว้มาใช้ในโปรเจก

ตารางที่ 3.8 คำอธิบายการสร้างโปรเจกการทดสอบ (Define Test Project)

หัวข้อ	คำอธิบาย
Use Case Name	Define Test Project
Scenario	เปิดโปรเจกใหม่เพื่อสร้างแผนการทดสอบ
Description	สร้างโปรเจกเพื่อทำการทดสอบระบบ
Trigger	สร้างโปรเจกเพื่อทดสอบระบบ เก็บรวบรวมความต้องการจากลูกค้า
Actor	ทีมทดสอบ (Tester Team)
Pre-Condition	สร้างโปรเจกการทดสอบ ตามความต้องการของลูกค้า
Post-Condition	สร้างโปรเจกได้ มีข้อมูลตรงตามความต้องการของลูกค้า

หัวข้อ	คำอธิบาย
Flow	<ol style="list-style-type: none"> 1. สร้างโปรเจกต์การทดสอบ 2. ระบุวัตถุประสงค์ และขอบเขตการทดสอบ 3. กำหนดโมดูลที่ต้องการทดสอบ 4. สกัดความต้องการทดสอบ 5. สร้างข้อกำหนดความเสี่ยง 6. สร้างกลวิธีในการจัดการปัญหาที่เกิดขึ้น 7. ระบุตารางเวลา ในการดำเนินการ

ตารางที่ 3.9 คำอธิบายการสร้างแผนการทดสอบ (Define Test Plan)

หัวข้อ	คำอธิบาย
Use Case Name	Define Test Plan
Scenario	กำหนดแผนการทดสอบ
Description	วางแผนการทดสอบ
Trigger	สร้างแผนการทดสอบที่ครอบคลุม
Actor	ทีมทดสอบ (Tester Team)
Pre-Condition	ทดสอบครอบคลุมทุกโมดูลที่ต้องการ
Post-Condition	ทุกโมดูลถูกทดสอบ
Flow	<ol style="list-style-type: none"> 1. สร้างแผนการทดสอบ 2. ระบุวัตถุประสงค์ และขอบเขตการทดสอบ 3. กำหนดโมดูลที่ต้องการทดสอบ 4. สกัดความต้องการทดสอบ 5. สร้างเงื่อนไขการทดสอบ 6. สร้างข้อกำหนดความเสี่ยง 7. สร้างกลวิธีในการจัดการปัญหาที่เกิดขึ้น 8. ระบุตารางเวลา ในการดำเนินการ

ตารางที่ 3.10 คำอธิบายการกำหนดการทดสอบ (Define Test Specification)

หัวข้อ	คำอธิบาย
Use Case Name	Define Test Specification
Scenario	สร้างแผนกรณีทดสอบ
Description	กรณีทดสอบพร้อม ขั้นตอนการทดสอบ
Trigger	กรณีทดสอบ
Actor	ทีมทดสอบ (Tester Team)
Pre-Condition	ระบุกรณีทดสอบครอบคลุมทุกความต้องการ
Post-Condition	ทดสอบกรณีทดสอบครบถ้วน
Flow	<ol style="list-style-type: none"> 1. สร้างกรณีการทดสอบ 2. ระบุวัตถุประสงค์ และขอบเขตการทดสอบ 3. กำหนดโมดูลที่ต้องการทดสอบ 4. สกัดความต้องการทดสอบ 5. สร้างขั้นตอนการทดสอบ 6. สร้างข้อกำหนดความเสี่ยง 7. สร้างกลวิธีในการจัดการปัญหาที่เกิดขึ้น 8. ระบุตารางเวลา ในการดำเนินการ

ตารางที่ 3.11 คำอธิบายการดำเนินการทดสอบ (Test Execution)

หัวข้อ	คำอธิบาย
Use Case Name	Test Execution
Scenario	ดำเนินการทดสอบ
Description	ทดสอบตามแผนที่ได้วางไว้
Trigger	ทดสอบซอฟต์แวร์
Actor	ทีมทดสอบ (Tester Team) ทีมพัฒนาระบบ (Developer Team)
Pre-Condition	ทดสอบตามกรณี และแผนการทดสอบ
Post-Condition	ทดสอบครบถ้วน
Flow	<ol style="list-style-type: none"> 1. นำแผนการทดสอบพร้อมกรณีทดสอบ ไปทำการทดสอบ 2. รายงานผลการทดสอบ

ตารางที่ 3.12 คำอธิบายการกำหนดสถานะการทดสอบ (Approval)

หัวข้อ	คำอธิบาย
Use Case Name	การอนุมัติ (Approval)
Scenario	อนุมัติสถานะการทดสอบ
Description	ผู้จัดการโครงการทำการตรวจสอบ พร้อมทั้งอนุมัติสถานะการทดสอบ
Trigger	แสดงสถานะการอนุมัติ
Actor	ผู้จัดการโครงการ (Project Manager)
Pre-Condition	อนุมัติสถานะโครงการเมื่อทดสอบเสร็จสิ้น
Post-Condition	สถานะเปลี่ยน เมื่อผู้จัดการโครงการ อนุมัติการทดสอบที่เสร็จสิ้น
Flow	1. เมื่อทดสอบเสร็จสิ้น 2. ผู้จัดการเปลี่ยนสถานะการทดสอบ

ตารางที่ 3.13 คำอธิบายการกำหนดสถานะการทดสอบ (Project Status)

หัวข้อ	คำอธิบาย
Use Case Name	Project Status
Scenario	สถานะโครงการ
Description	แสดงสถานะล่าสุดของโครงการ
Trigger	แสดงสถานะ
Actor	ผู้จัดการโครงการ (Project Manager) ทีมทดสอบ (Tester Team) ลูกค้า (Customer)
Pre-Condition	เปลี่ยนสถานะของโครงการ เปลี่ยนตามขั้นตอนการทดสอบ
Post-Condition	สถานะเปลี่ยน
Flow	1. เปลี่ยนตามผลการทดสอบ

ตารางที่ 3.14 คำอธิบายการวัดประสิทธิภาพของการทดสอบ (Test Measurement)

หัวข้อ	คำอธิบาย
Use Case Name	Test Measurement
Scenario	วัดประสิทธิภาพ
Description	วัดประสิทธิภาพความต้องการ วัดประสิทธิภาพของซอฟต์แวร์ที่ทดสอบ
Trigger	วัดประสิทธิภาพมีความถูกต้อง
Actor	ผู้ทดสอบ (Tester Team)
Pre-Condition	วัดประสิทธิภาพความต้องการ วัดประสิทธิภาพของซอฟต์แวร์ที่ทดสอบ
Post-Condition	วัดประสิทธิภาพ
Flow	1. วัดประสิทธิภาพความต้องการ 2. วัดประสิทธิภาพของซอฟต์แวร์ที่ทดสอบ 3. แสดงผลในรูปกราฟ

ตารางที่ 3.15 คำอธิบายการสร้างรายงาน (Report)

หัวข้อ	คำอธิบาย
Use Case Name	Report
Scenario	สร้างรายงานประกอบการทดสอบ
Description	รายงานการทดสอบ เช่น Test Plan, Test Case
Trigger	สร้างรายงานแผนการทดสอบที่ถูกต้อง
Actor	ทีมทดสอบ (Tester Team) ลูกค้า (Customer)
Pre-Condition	สร้างรายงานตามความต้องการ สร้างรายงานด้วยความถูกต้อง
Post-Condition	สร้างรายงาน
Flow	1. เลือกโปรเจกต์ที่ต้องการดูข้อมูล 2. สร้างรายงาน

3.3.3 ER-Diagram

1. ฐานข้อมูลของระบบ

ในส่วนฐานข้อมูลรวมของระบบ มีการสร้างตารางที่ใช้ในการจัดเก็บข้อมูล ซึ่งในส่วนนี้แสดงความสัมพันธ์แต่ละตารางที่ใช้ในการสร้างโปรเจกการทดสอบ แสดงด้วยแผนภาพ ER-Diagram ดังรูปที่ 3.13

ภายในส่วนฐานข้อมูลรวมของระบบ ประกอบด้วยตารางดังนี้

Define Team เป็นตารางที่ใช้ในการเก็บรูปแบบของทีมที่สร้างขึ้น ซึ่งมีความสัมพันธ์แบบ 1 ต่อ m กล่าว คือ รูปแบบทีม 1 แบบ สามารถสร้างโปรเจกการทดสอบได้หลายโปรเจก

Team Member เป็นตารางที่ใช้ในการเก็บข้อมูลของสมาชิกในทีม ซึ่งมีความสัมพันธ์แบบ 1 ต่อ m กล่าว คือ สมาชิก 1 คน สามารถอยู่ในทีมโปรเจกการทดสอบได้หลายโปรเจก

Setting Organization เป็นตารางที่ใช้ในการเก็บข้อมูลขององค์กร หรือ ลูกค้า ซึ่งมีความสัมพันธ์แบบ 1 ต่อ m กล่าว คือ องค์กร หรือ ลูกค้า 1 คน สามารถอยู่ในโปรเจกการทดสอบได้หลายโปรเจก

Project Test เป็นตารางหลักที่ใช้ในการเก็บคีย์ตัวแปรที่ใช้เรียกเพื่อสร้างความสัมพันธ์กับตารางอื่น ๆ ที่มีข้อมูลสัมพันธ์กัน

Test Plan เป็นส่วนที่ใช้ในการเก็บแผนการทดสอบ

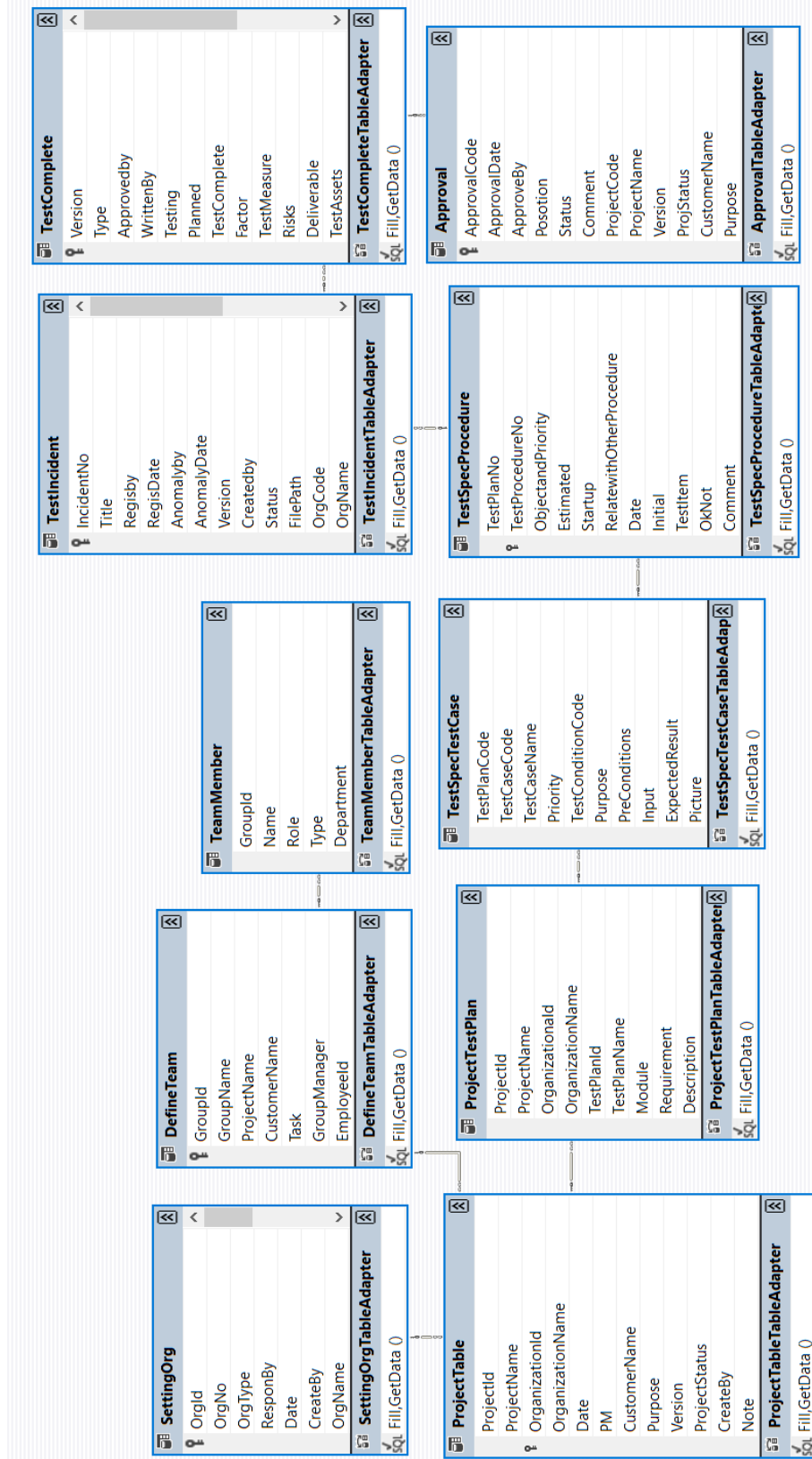
Test Specification Test Case เป็นตารางที่ใช้ในการเก็บกรณีทดสอบ ซึ่งมีความสัมพันธ์แบบ 1 ต่อ m กล่าว คือ แผนการทดสอบ 1 แผน สามารถมีกรณีทดสอบได้หลายแผน

Test Specification Test Procedure เป็นตารางที่ใช้ในการเก็บขั้นตอนการทดสอบ ซึ่งมีความสัมพันธ์แบบ 1 ต่อ 1 กล่าว คือ กรณีทดสอบ 1 กรณี สามารถมีขั้นตอนการทดสอบได้ 1 ขั้นตอน

Test Incident เป็นตารางที่ใช้ในการเก็บแบบรายงานที่แสดงเหตุการณ์ที่เกิดขึ้นระหว่างการทดสอบ ซึ่งมีความสัมพันธ์แบบ 1 ต่อ m กล่าว คือ แผนการทดสอบ 1 แผน สามารถเกิดเหตุการณ์ระหว่างการทดสอบได้มากกว่า 1 เหตุการณ์

Test Complete เป็นตารางที่ใช้ในการเก็บรายงานที่ทดสอบสมบูรณ์ ซึ่งมีความสัมพันธ์แบบ 1 ต่อ 1 กล่าว คือ แผนทดสอบ 1 แผน มีรายงานการทดสอบ 1 รายงาน

Approval เป็นตารางที่ใช้ในการเก็บสถานการณ์อนุมัติของโปรเจก ซึ่งมีความสัมพันธ์แบบ 1 ต่อ 1 กล่าว คือ โปรเจก 1 โปรเจกมีการอนุมัติที่สมบูรณ์เพียง 1 ครั้ง



รูปที่ 3.13 แผนภาพ ER-Diagram ของระบบรวม

บทที่ 4

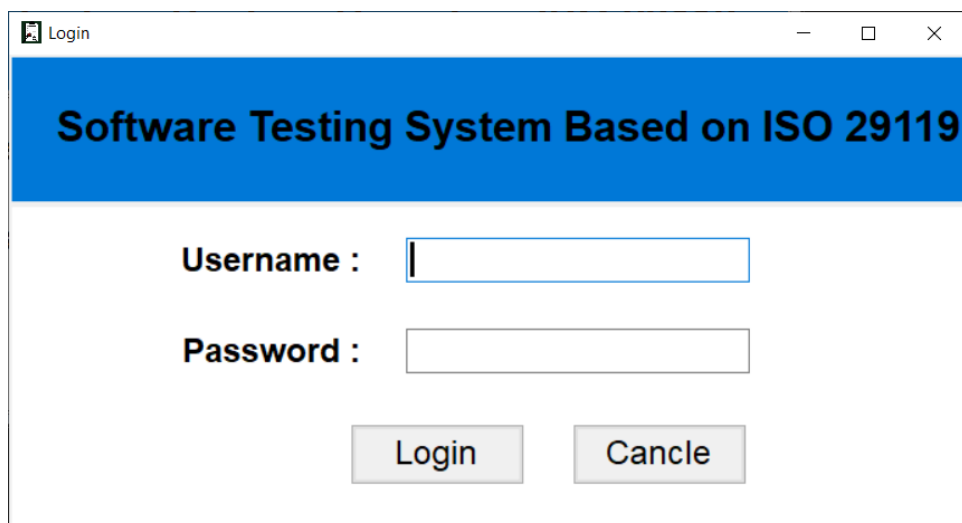
ผลการวิจัยและอภิปรายผล

จากวิธีดำเนินงานวิจัยสามารถสร้างระบบทดสอบซอฟต์แวร์ได้ตามวิธีการที่วางไว้ โดยภายในบทนี้แสดงผลการดำเนินงานวิจัย แบ่งออกเป็น 2 ส่วน คือ ส่วนผลการใช้งานของระบบ และส่วนผลแบบสำรวจความพึงพอใจของผู้ใช้งานระบบ ซึ่งในส่วนผลการใช้งานของระบบสามารถแบ่งฟังก์ชันของระบบออกเป็น 11 ส่วน ประกอบด้วย ส่วนการเข้าสู่ระบบ (System Login), ส่วนกำหนดค่าพื้นฐานให้แก่ระบบ, ส่วนการกำหนดสมาชิกในทีม (Define Team Member), ฟังก์ชันกำหนดข้อมูลองค์กรที่เป็นลูกค้า (Setting Organization Information), ส่วนกำหนดข้อมูลโปรเจกต์ (Define Test Project), กำหนดข้อมูลแผนการทดสอบ (Define Test Plan), กำหนดข้อมูลเฉพาะสำหรับการทดสอบ (Define Test Specification), ส่วนทดสอบ (Test Execution), การวัดประสิทธิภาพ (Measurement), การอนุมัติการทดสอบ (Approved) และ ส่วนแสดงผลรายงาน (Report) ซึ่งทางผู้วิจัยได้ทำการอธิบายรายละเอียดการทำงานในแต่ละส่วน ดังนี้

4.1 ผลการใช้งานของระบบทดสอบซอฟต์แวร์

4.1.1 เข้าสู่ระบบ (System Login)

เมื่อเริ่มใช้งานระบบจะปรากฏหน้าจอตั้งชื่อเข้าใช้งาน ที่ประกอบด้วย username และ password ดังรูปที่ 4.1



รูปที่ 4.1 สมัครสมาชิก

หลังจากเข้าสู่ระบบเรียบร้อยแล้ว จะแสดงหน้าจอหลักในการใช้งาน เพื่อทำการเข้าสู่ฟังก์ชันต่าง ๆ ของระบบ ซึ่งจะปรากฏดังรูปที่ 4.2

เมื่อทำการเข้าสู่ระบบเรียบร้อยแล้ว ในหน้าจอการใช้งานหลักจะแสดงข้อมูลออกเป็น 4 ส่วน ประกอบด้วย ข้อมูลส่วนบุคคล (Information), งานที่กำลังดำเนินการอยู่ (Working Status), งานทั้งหมดภายในบริษัท (All Tasks) และทีมที่ได้รับมอบหมายทั้งหมด (Team Existing) โดยในส่วนของทีมที่ได้รับมอบหมายงานนั้นข้อมูลจะแสดงเฉพาะผู้ใช้งานที่เป็นผู้จัดการโครงการเท่านั้น

The screenshot shows a web application interface with the following sections:

- Information:**
 - Employee Id : 00005
 - Name : Test Team
 - Start working : 1/3/2018
 - Department : TesterTeam
 - Position : Main
- Working Status (My tasks):**

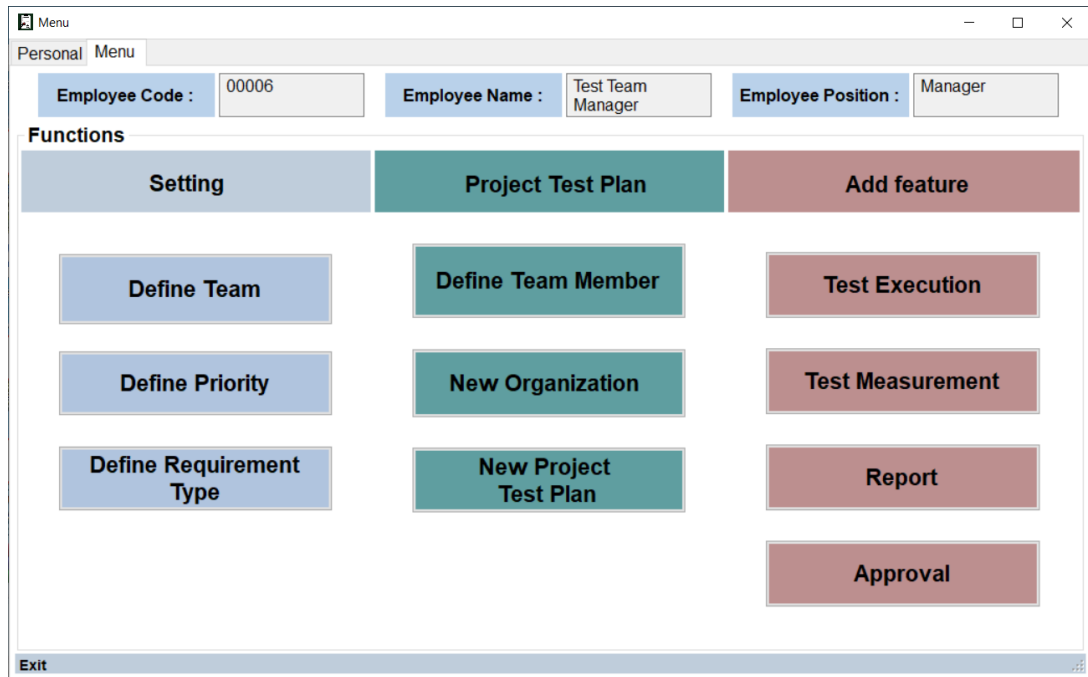
ProjectId	ProjectCode	Organization	ProjectName	Date	W
96400	FR0001	OG0001	Testing R...	12/13/2019	1
10380	TT0001	1121.0001	Test	12/13/2019	1
- All Tasks:**

ProjectId	ProjectCode	Organization	ProjectName	Date
92249	2323		d	12/11/2019
96400	FR0001	OG0001	Testing R...	12/13/2019
74186	ICCP001	ICC001	ทดสอบระบบขอ...	12/12/2019
60991	MN001	OG0001	ข่าวสารสาม	12/12/2019
38460	PC0001.01	00005	Registrati...	12/9/2019
85426	POL001	00006	ff	12/13/2019
75088	Project001	00006	Test	12/11/2019
88100	PT001	00006	ทดสอบ Inter...	11/24/2019
10380	TT0001	1121.0001	Test	12/13/2019
- Team Existing:** (Empty table)

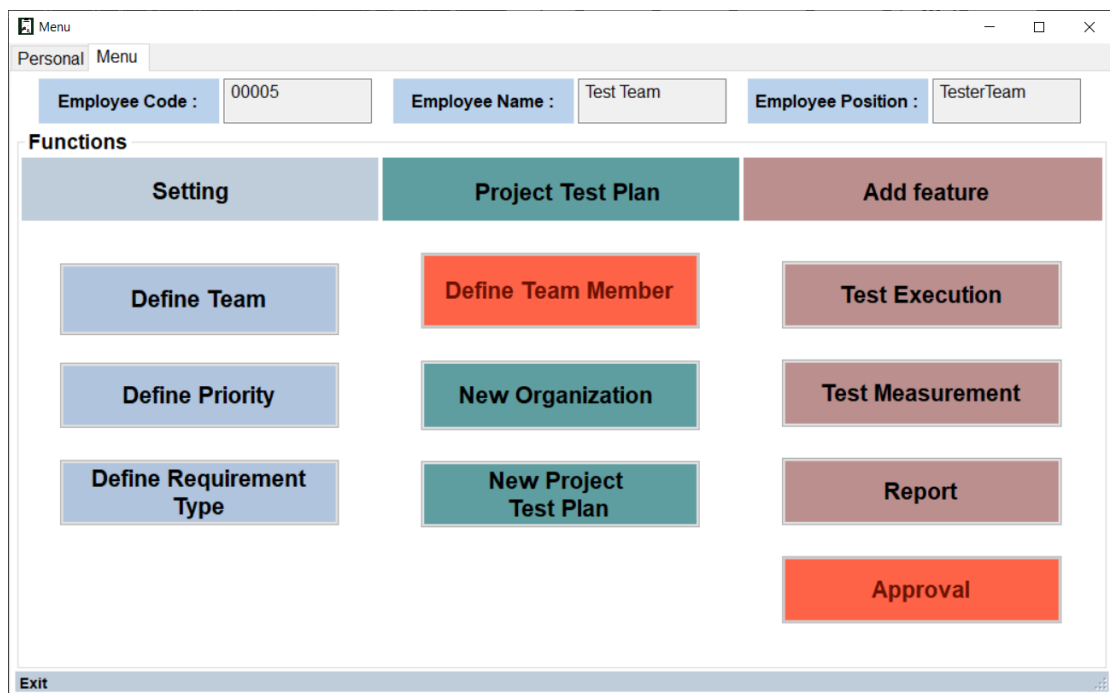
รูปที่ 4.2 หน้าจอการใช้งานหลัก

เมื่อเข้าสู่หน้าจอหลักเรียบร้อยแล้ว ทำการเลือกเมนูโดยกดที่แถบเมนู **Menu** จะปรากฏหน้าจอหลักในการใช้งาน แสดงดังรูปที่ 4.3 และ รูปที่ 4.4 ซึ่งทั้งสองรูปแสดงผลต่างกัน โดยแสดงผลตามการลงชื่อเข้าใช้งาน รูปที่ 4.3 คือ เมนูการใช้งานของผู้จัดการโปรเจค (Project Manager) ที่สามารถใช้งานได้ทุกฟังก์ชัน และ รูปที่ 4.4 คือ เมนูการใช้งานของทีมทดสอบ (Tester Team) ที่ถูกจำกัดการเข้าใช้งานในบางฟังก์ชัน คือ ฟังก์ชันกำหนดสมาชิกในทีม (Define Team Member) และ ฟังก์ชันการอนุมัติการทดสอบ (Approval) ที่ผู้จัดการโครงการสามารถดำเนินการได้เพียงผู้เดียว

ซึ่งเมนูหลักข้างต้น จะประกอบด้วยเมนูย่อย คือ Define Team, Define Priority, Define Requirement Type, Define Team Member, New Organization, New Project Test Plan, Test Execution, Test Measurement, Report และ Approval



รูปที่ 4.3 หน้าจอเมนูการใช้งานของผู้จัดการโปรเจค (Project Manager)



รูปที่ 4.4 หน้าจอเมนูการใช้งานของทีมทดสอบ (Tester Team)

4.1.2 กำหนดค่าพื้นฐานให้แกระบบ

การตั้งค่าพื้นฐานแกระบบประกอบด้วย เมนูสำหรับดำเนินการ 3 เมนู คือ Define Team, Define Priority และ Define Requirement Type แสดงรายละเอียดดังนี้

1. การตั้งค่าทีม (Define Team) เพื่อกำหนดทีมสำหรับใช้งานเพื่อทดสอบโปรเจกต์ เมื่อเข้าสู่เมนูหลักเรียบร้อยแล้ว กดที่ปุ่ม Define Team จะปรากฏหน้าจอตั้งรูปที่ 4.5

GroupNo	GroupName	Description
1	ทีมทดสอบ 01	Tester
2	Interface Test Te...	Testing interface
3	Acceptance Team	ทดสอบการยอมรับ
4	Mango Team	ทดสอบ

รูปที่ 4.5 หน้าจอเมนูการตั้งค่าทีม (Define Team)

2. การตั้งค่าลำดับความสำคัญของโปรเจกต์ (Define Priority) เพื่อกำหนดกลุ่มของลำดับความสำคัญของโปรเจกต์ โดยกดที่ปุ่ม Define Priority เมื่อกดแถบเมนูเรียบร้อยแล้ว จะปรากฏหน้าจอตั้งรูปที่ 4.6

PriorityNo	PriorityName	PriorityValue
1	low	0
4	super high	4
1	low	1
2	medium	2
3	high	3

รูปที่ 4.6 หน้าจอเมนูการตั้งค่าลำดับความสำคัญ (Priority)

3. การตั้งค่าประเภทของความต้องการของลูกค้า (Define Requirement Type) เพื่อกำหนดความต้องการเพิ่มเติมที่ต้องการใช้ในโปรเจกต์การทดสอบ ซึ่งระบบเปิดให้ผู้ใช้สามารถเพิ่มประเภทของความต้องการได้ โดยกดที่ปุ่ม Define Requirement Type ที่หน้าจอหลัก เมื่อกดแถบเมนูเรียบร้อยแล้ว จะปรากฏหน้าจอตั้งรูปที่ 4.7

TypeCode	TypeName	TypeDescription
1	Other	Other
2	Architectural req.	Architectural req.
3	Business require.	Business require.
4	User (State)code	User requirement
5	Functional (Solut.	Functional requir...
6	Non-Functional r.	Non requirement
7	Implementation (I.	Implementation re.
8	UI Test	User Interface re.

รูปที่ 4.7 หน้าจอการตั้งค่าประเภทของความต้องการ (Requirement Type)

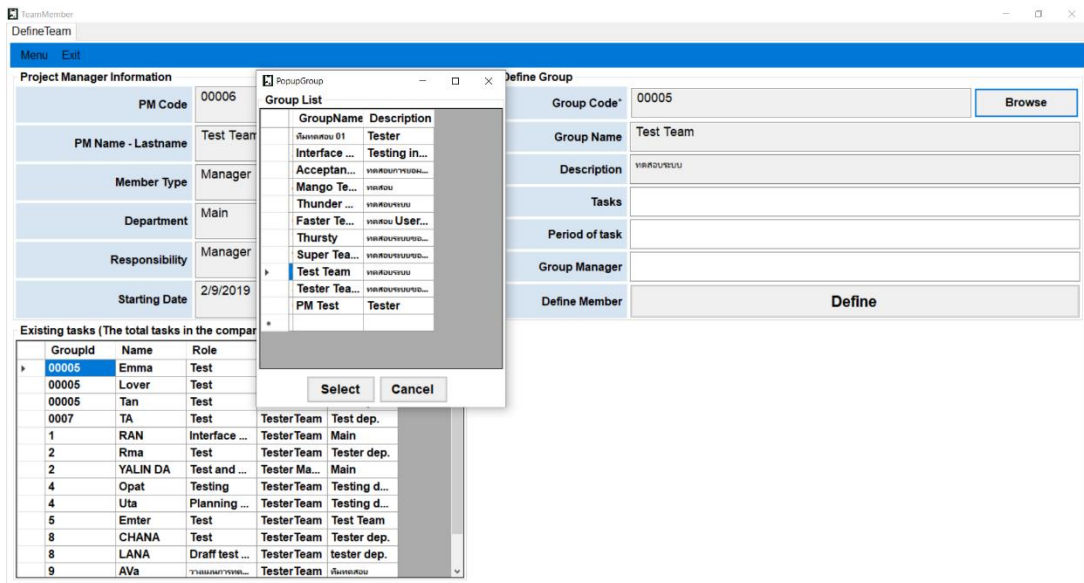
4.1.3 กำหนดสมาชิกในทีม (Define Team Member)

เมื่อกดปุ่ม Define Team Member ในหน้าเมนูหลัก เมื่อกดเมนูเรียบร้อยแล้ว จะปรากฏหน้าจอ ดังรูปที่ 4.8

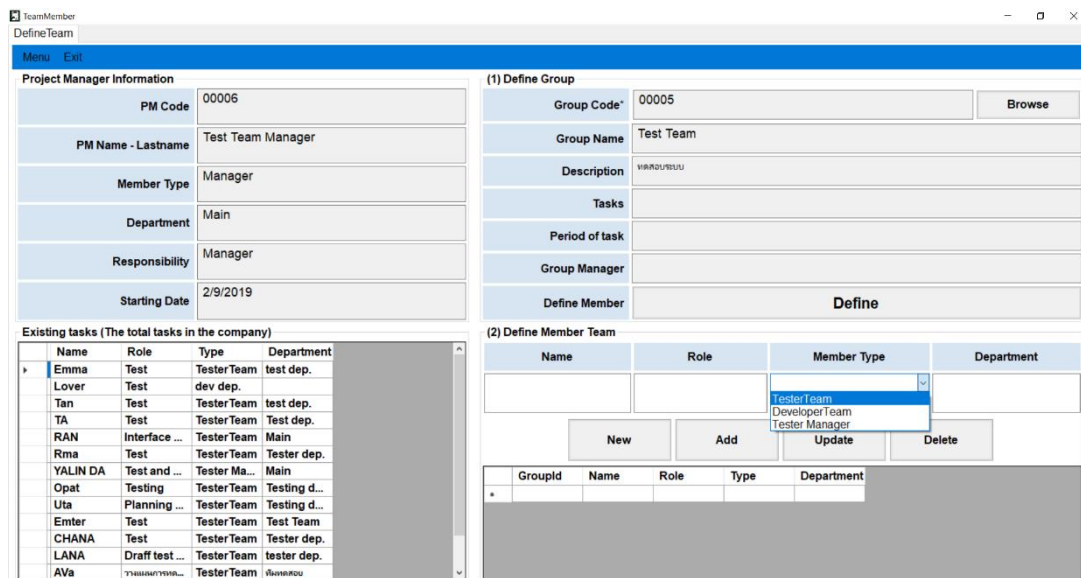
ฟังก์ชันกำหนดสมาชิกในทีมซึ่งสิทธิการใช้งานในฟังก์ชันนี้จะใช้ได้เฉพาะ Project Manager เริ่มด้วยการกำหนดกลุ่มของสมาชิก โดยผู้ใช้งานต้องกดปุ่ม Browse ในหัวข้อ Group Code เพื่อทำการเลือกกลุ่มที่ได้สร้างไว้ เมื่อกรอกข้อมูลเรียบร้อยแล้ว กดปุ่ม Define เพื่อกำหนดสมาชิกในทีม พร้อมระบุ บทบาท หน้าทีการทำงานภายในโปรเจกต์ แสดงดังรูปที่ 4.9 พร้อมทั้งเลือกประเภทของสมาชิก แสดงดังรูปที่ 4.10

Name	Role	Type	Department
TA	Test	TesterTeam	Test dep.
RAN	Interface ...	TesterTeam	Main
Rma	Test	TesterTeam	Tester dep.
YALIN DA	Test and ...	Tester Ma...	Main
Opat	Testing	TesterTeam	Testing d...
Uta	Planning ...	TesterTeam	Testing d...
Emter	Test	TesterTeam	Test Team
CHANA	Test	TesterTeam	Tester dep.
LANA	Draff test ...	TesterTeam	tester dep.
AVa	วางแผนการทดสอบ...	TesterTeam	ทีมทดสอบ
Tana	ทดสอบ	TesterTeam	ทีมทดสอบ

รูปที่ 4.8 กำหนดกลุ่มสมาชิกในทีม



รูปที่ 4.9 กำหนดหน้าที่สมาชิกในทีม



รูปที่ 4.10 แสดงประเภทของสมาชิกในทีม

4.1.4 ฟังก์ชันกำหนดข้อมูลองค์กรที่เป็นลูกค้า (Setting Organization Information)

เมื่อทำการกดปุ่ม Create Organization Information ในหน้าเมนูหลักแล้ว จะปรากฏหน้าจอตั้งรูปที่ 4.9 ซึ่งภายในฟังก์ชัน Setting Organization Information ประกอบด้วยแถบเมนูย่อย 2 แถบ คือ ข้อมูลองค์กร (Define Information) และ ข้อมูลเฉพาะการทดสอบ (Test Specification)

1. ข้อมูลขององค์กร หรือ ข้อมูลลูกค้า (Define Information)

เมื่อเข้าสู่หน้าจอแล้ว จะปรากฏดังรูปที่ 4.11 โดยในช่อง Organization Id ระบบจะทำการสร้างรหัสขึ้นมาให้ โดยผู้ใช้งานไม่สามารถแก้ไขข้อมูลในช่องนี้ได้ โดยผู้ใช้งานสามารถเรียกข้อมูลองค์กรเก่ามาทำการแก้ไขได้โดยกดที่เมนู Edit เมื่อเข้าเรียบร้อยแล้ว จะแสดงดังรูปที่ 4.12

รูปที่ 4.11 ข้อมูลองค์กร หรือ ลูกค้า

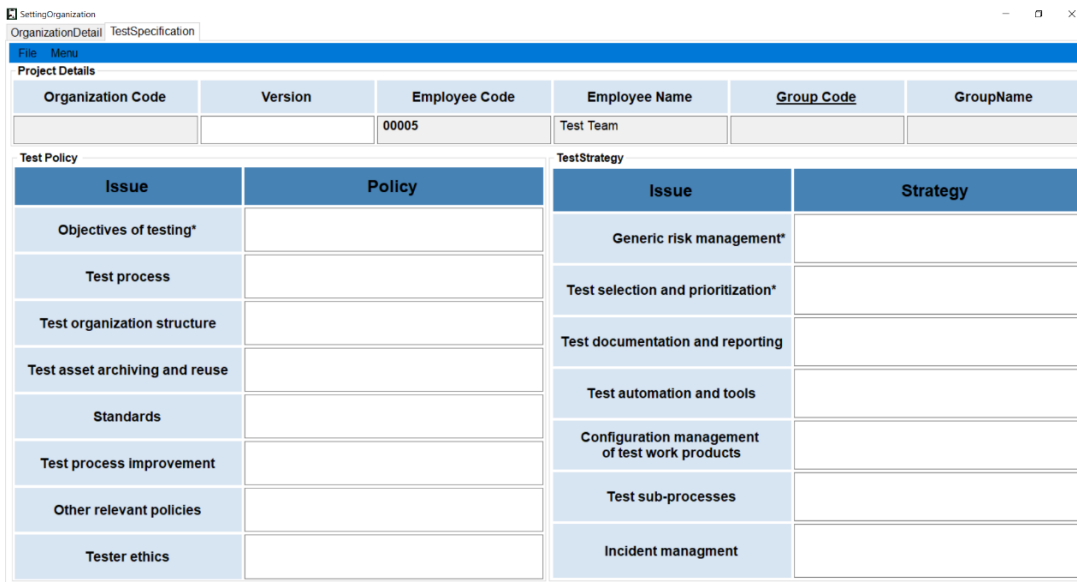
OrgNo	OrgType	ResponBy	Date	CreateBy	Or
OR0001	SMEs Bu...	HR Busin...	11/10/201...	Project M...	สาขา
OG0001	Software ...	User in co...	12/8/2019 ...	Tester team	LA
LNC0001	Software ...	User in co...	12/8/2019 ...	Tester team	LN

รูปที่ 4.12 ข้อมูลองค์กร หรือ ลูกค้า

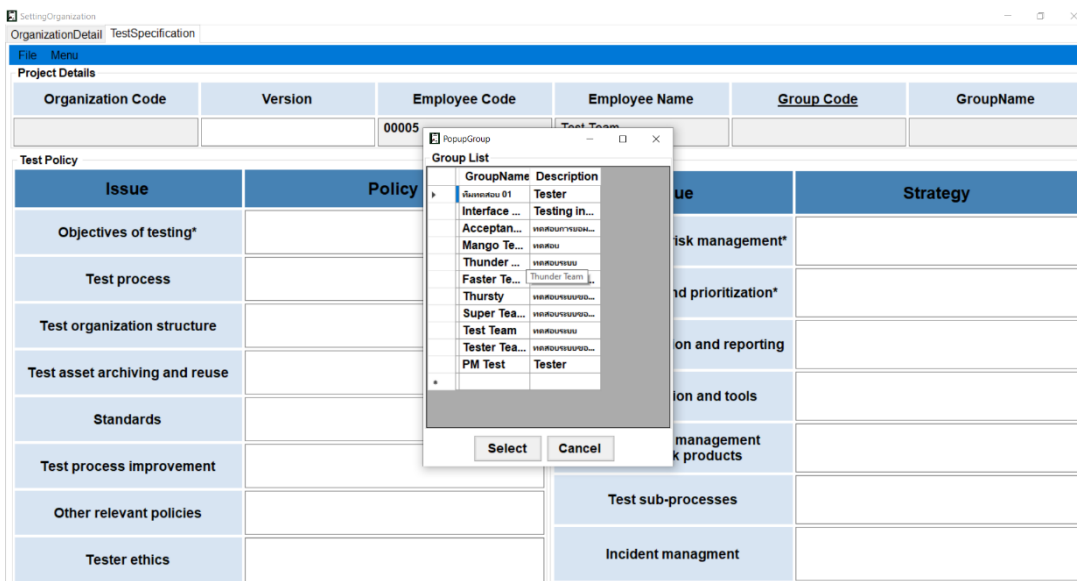
2. ข้อมูลเฉพาะการทดสอบ (Test Specification)

เมื่อเข้าสู่หน้าจอแล้วจะปรากฏหน้าจอดังรูปที่ 4.13 ซึ่งจะแสดงข้อมูลนโยบาย และกลยุทธ์ขององค์กร (Define Organization Test Strategy and Policy) โดยในส่วน Project Details

ในช่อง Organization Id, Project Id, Test Specification ระบบจะทำการสร้างรหัสขึ้นมาเอง โดยที่ผู้ใช้งานจะไม่สามารถเปลี่ยนแปลงแก้ไข ข้อมูลในช่องนี้ได้ โดยต้องการทำการเลือกทีมทดสอบที่จะเข้ามาทำการทดสอบงานขององค์กร กลุ่ม Group Code เพื่อเลือกทีมที่สร้างไว้ แสดงดังรูปที่ 4.14



รูปที่ 4.13 กำหนดนโยบาย และกลยุทธ์ขององค์กร



รูปที่ 4.14 กำหนดนโยบาย และกลยุทธ์ขององค์กร

4.1.5 ส่วนกำหนดข้อมูลโปรเจค (Define Test Project)

ในส่วนกำหนดข้อมูลโปรเจค เมื่อคลิกปุ่ม Create New Project Test Plan ในหน้าเมนูหลักเรียบร้อยแล้ว จะปรากฏหน้าจอแสดงผล ดังรูปที่ 4.15 ภายในฟังก์ชัน Project Test Plan ประกอบด้วยเมนูย่อย คือ Purpose, Function, Requirement, Product Risk, Project Risk, Test Policy, Strategy, Schedule และ Staffing

The screenshot shows the 'Project Test Plan' application window. The top menu bar includes 'Setting', 'ProjectTestPlan', 'TestPlan', 'TestSpecification', 'TestExecution', 'TestReport', and 'Exit'. The main interface is divided into several sections:

- Project Details:** Contains fields for Project Code*, Version, Date (Tuesday, December 1), Project Name, Project Type, and Project Status. There is also an 'Activated' checkbox.
- Organization Existing:** A table listing organizations with columns for OrgNo and OrgN. The table contains:

OrgNo	OrgN.
00005	H.K C
7009	Y.TK C
00006	MFEC
OR0001	สำนักงาน
- Organization Detail:** Fields for Organization Code* and Organization Name.
- Employee Information:** Fields for Employee Code (00005), Employee Name (Test Team), Employee Position (Tester Team), and Team.
- Purpose of Project Test Plan:** A table with columns: System Name, Purpose, Tools, Automation Test, and Description.
- Scope of Project Test Plan:** A table with columns: Scope No*, Test Type, Scope, and Responsible.
- Buttons:** 'New', 'Add', 'Update', and 'Delete' buttons are located below the tables.
- Table:** A table at the bottom with columns: ScopeNo, TestType, Scope, and Responsibility. It has one row with a '*' in the first column.

รูปที่ 4.15 หน้าจอหลักของ Project Test Plan

เมื่อเข้าสู่ฟังก์ชันเรียบร้อยแล้ว ผู้ใช้งานต้องทำการเลือกประเภทของโปรเจกต์ (Project Type) เพื่อระบุประเภทของโปรเจกต์ที่ได้ทดสอบ ดังรูปที่ 4.16 และต้องเลือก Organization ที่ต้องการทดสอบ แสดงดังรูปที่ 4.17

This screenshot is identical to the previous one, but with a dropdown menu open over the 'Project Type' field. The menu options are:

- AddNewProject
- Adjust (Add Requirement)
- Adjust (Incident Test)
- Adjust (Not Approve)

รูปที่ 4.16 หน้าจอ Test Project ในส่วน Project Type

Organization Existing

	OrgNo	OrgName ^
▶	00005	H.K Co
	7009	Y.T.K C
	00006	MFEC
	OR0001	ข้าราชการ

รูปที่ 4.17 รายละเอียดขององค์กรที่มีอยู่ในระบบ

- ฟังก์ชัน Purpose เมื่อกดเข้ามาสู่ฟังก์ชันนี้จะประกอบด้วย 2 ส่วน คือ ส่วน Purpose เพื่อกำหนดวัตถุประสงค์ของการทดสอบ และ ส่วน Scope เพื่อกำหนดขอบเขตของการทดสอบ จะปรากฏหน้าจอแสดงข้อมูล ดังรูปที่ 4.15
- ฟังก์ชัน Function เป็นส่วนที่ใช้ในการกำหนดฟังก์ชันที่มีอยู่ในระบบที่ต้องการทดสอบ เมื่อเข้าสู่หน้าจอจะปรากฏหน้าจอแสดงข้อมูล ดังรูปที่ 4.18

The screenshot shows the 'Function of Test' section of the ProjectTestPlan application. At the top, there is a menu bar with 'File' and 'Menu'. Below it is a toolbar with buttons for 'New', 'Add', 'Update', and 'Delete'. The main area contains a table with the following columns: 'FunctionNo', 'FunctionName', 'Scope', and 'Description'. The table is currently empty, with a single row containing an asterisk (*). Above the table, there are input fields for 'Function No*', 'Function Name*', 'Function Scope', and 'Description'. To the right of the table, there are buttons for 'New', 'Add', 'Update', and 'Delete'. At the top of the interface, there is a dropdown menu for 'Organization Existing' which is open, showing a list of organizations: 00005 (H.K Co), 7009 (Y.T.K C), 00006 (MFEC), and OR0001 (ข้าราชการ). Other sections of the interface include 'Project Details' and 'Employee Information'.

รูปที่ 4.18 โปรแกรมทดสอบในส่วนฟังก์ชัน (Function)

- ฟังก์ชัน Requirement เพื่อกำหนด และ สกัด (Extract) ความต้องการของลูกค้า เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังภาพที่ 4.19 ผู้ใช้งานต้องทำการเลือกประเภทของความ ต้องการของลูกค้าในแต่ละรายการที่ได้สร้างขึ้น โดยต้องกดกล่อง combo เมื่อกดเรียบร้อยแล้ว จะปรากฏดังภาพที่ 4.20

ProjectTestPlan

Setting ProjectTestPlan TestPlan TestSpecification TestExecution TestReport Exit

Project Details

Project Code*	Version	Date
		Tuesday , December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Existing

OrgNo	OrgNa
00005	H.K C
7009	Y.TK C
00006	MFEC
OR0001	ชวบท

Organization Detail

Organization Code*	Employee Code	Employee Name
00005	00005	Test Team
Organization Name	Employee Position	Team
	TesterTeam	

Employee Information

Employee Code	Employee Name
00005	Test Team
Employee Position	Team
TesterTeam	

Purpose: Function Requirement ProductRisk ProjectRisk TestPolicy Strategy Schedule Staffing

File Menu

Requirement

Requirement No*	Requirement Type	Requirements*
	Other	
Priority*	Informant	Description

New Add Update Delete

Requirement	Requirement	Requirement	Priority	Informant	Description
*					

รูปที่ 4.19 โปรแกรมทดสอบในส่วนกำหนดความต้องการ (Requirement)

ProjectTestPlan

Setting ProjectTestPlan TestPlan TestSpecification TestExecution TestReport Exit

Project Details

Project Code*	Version	Date
		Tuesday , December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Existing

OrgNo	OrgNa
00005	H.K C
7009	Y.TK C
00006	MFEC
OR0001	ชวบท

Organization Detail

Organization Code*	Employee Code	Employee Name
00005	00005	Test Team
Organization Name	Employee Position	Team
H.K Company	TesterTeam	Interface Test Team

Employee Information

Employee Code	Employee Name
00005	Test Team
Employee Position	Team
TesterTeam	Interface Test Team

Purpose: Function Requirement ProductRisk ProjectRisk TestPolicy Strategy Schedule Staffing

File Menu

Requirement

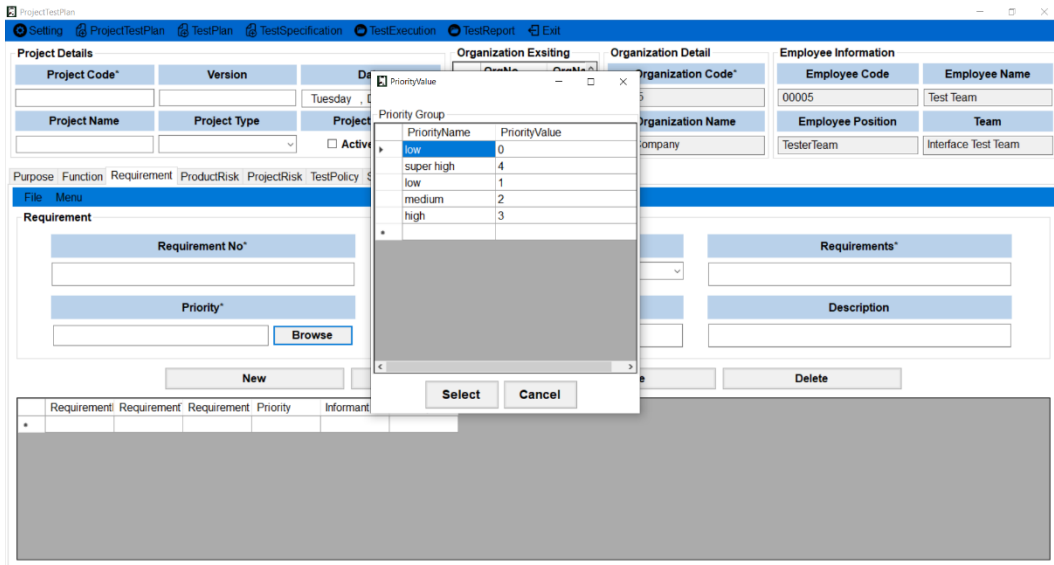
Requirement No*	Requirement Type	Requirements*
	Other	
Priority*	Informant	Description

New Add Update Delete

Requirement	Requirement	Requirement	Priority	Informant	Description
*					

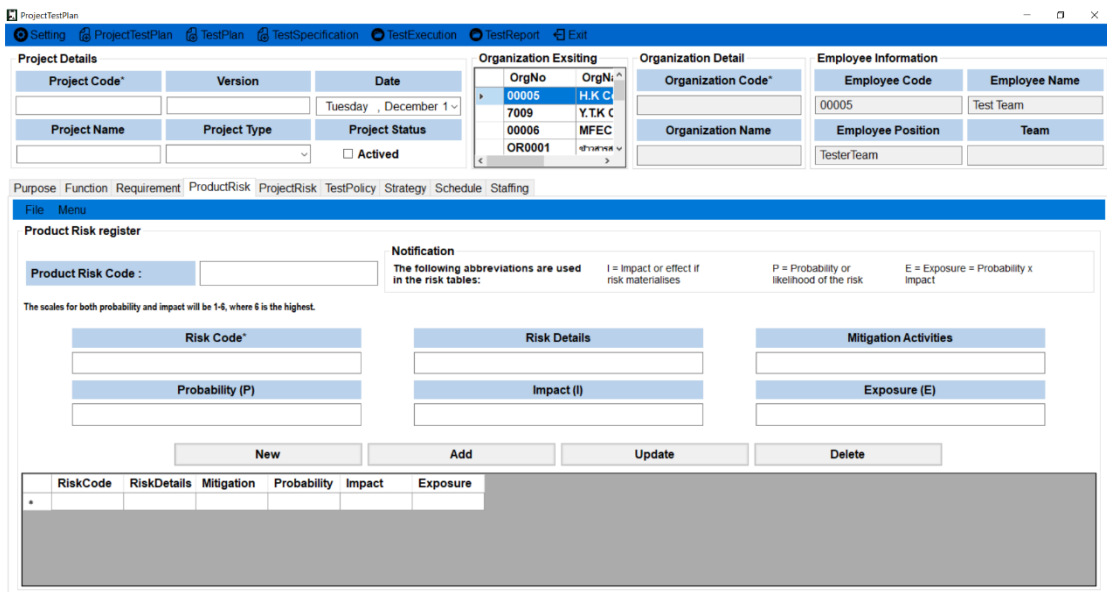
รูปที่ 4.20 ประเภทของความต้องการ (Requirement Type)

เมื่อทำการเลือกประเภทความต้องการเรียบร้อยแล้ว ต้องทำการเลือกลำดับความสำคัญ (Priority) ที่ช่อง Priority โดยกดที่ปุ่ม Browse ปรากฏดังรูปที่ 4.21



รูปที่ 4.21 ลำดับความสำคัญของความต้องการ

4. ฟังก์ชัน Product Risk เพื่อกำหนดความเสี่ยงของผลิตภัณฑ์ เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.22



รูปที่ 4.22 โปรแกรมทดสอบในส่วนความเสี่ยงของผลิตภัณฑ์ (Product Risk)

5. ฟังก์ชัน Project Risk เพื่อกำหนดความเสี่ยงของโปรเจกต์ เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.23

Project Test Plan

Setting ProjectTestPlan TestPlan TestSpecification TestExecution TestReport Exit

Project Details

Project Code*	Version	Date
		Tuesday , December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Exsiting

OrgNo	OrgNi
00005	H.K C
7009	Y.TK C
00006	MFEC
OR0001	

Organization Detail

Organization Code*	Employee Code	Employee Name
	00005	Test Team
Organization Name	Employee Position	Team
	TesterTeam	

Employee Information

Employee Code	Employee Name
00005	Test Team
Employee Position	Team
TesterTeam	

Purpose Function Requirement ProductRisk ProjectRisk TestPolicy Strategy Schedule Staffing

File Menu

Project Risk Register

Project Risk Code : Notification: D = Development T = Test

Risk Code*	Consequences	Prevention	Mitigation
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

New Add Update Delete

RiskCode	Consequen	Prevention	Mitigation
*			

รูปที่ 4.23 โปรแกรมทดสอบในส่วนความเสี่ยง (Risk)

6. ฟังก์ชัน Test Policy เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.24

Project Test Plan

Setting ProjectTestPlan TestPlan TestSpecification TestExecution TestReport Exit

Project Details

Project Code*	Version	Date
		Tuesday , December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Exsiting

OrgNo	OrgNi
00005	H.K C
7009	Y.TK C
00006	MFEC
OR0001	

Organization Detail

Organization Code*	Employee Code	Employee Name
	00005	Test Team
Organization Name	Employee Position	Team
	TesterTeam	

Employee Information

Employee Code	Employee Name
00005	Test Team
Employee Position	Team
TesterTeam	

Purpose Function Requirement ProductRisk ProjectRisk TestPolicy Strategy Schedule Staffing

File Menu

Test Policy

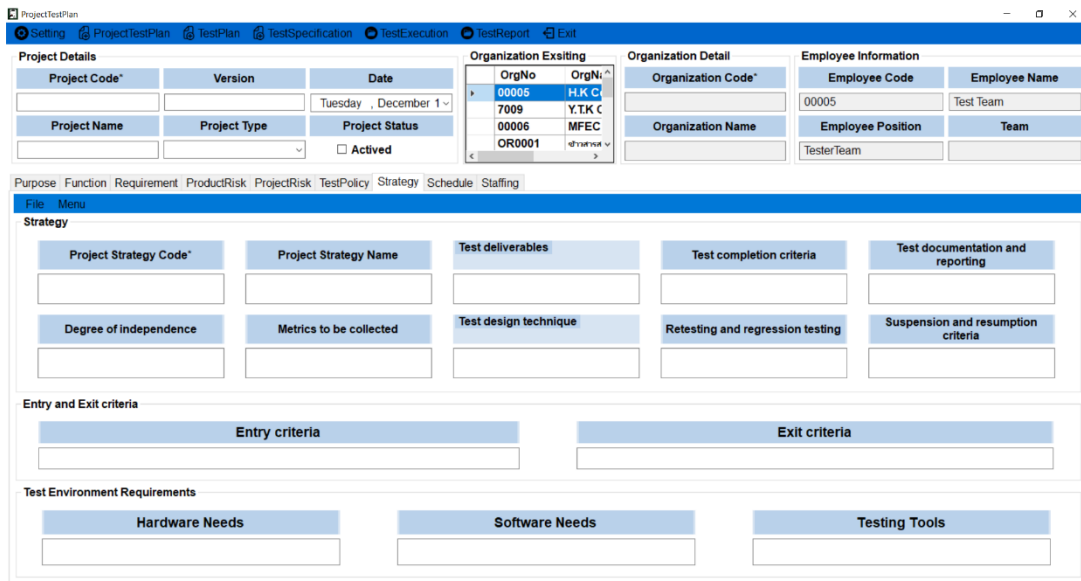
Test Policy Code*	Objective	Testing process	Organization
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Evaluation of Testing	Standards	Approach to test process improvement	Policies
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

New Add Update Delete

TestPolicyCo	Objective	TestProcess	Organization	EvaluationTe	Standards	ApproachTes	Policies
*							

รูปที่ 4.24 โปรแกรมทดสอบในส่วนนโยบาย (Test Policy)

7. ฟังก์ชัน Strategy เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.25



Project Test Plan

Setting Project Test Plan Test Plan Test Specification Test Execution Test Report Exit

Project Details

Project Code*	Version	Date
		Tuesday , December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Existing

OrgNo	OrgNi
00005	H.K.C
7009	Y.T.K.C
00006	MFEC
OR0001	

Organization Detail

Organization Code*	Employee Code	Employee Name
	00005	Test Team
Organization Name	Employee Position	Team
	TesterTeam	

Employee Information

Purpose Function Requirement ProductRisk ProjectRisk TestPolicy Strategy Schedule Staffing

File Menu

Strategy

Project Strategy Code*	Project Strategy Name	Test deliverables	Test completion criteria	Test documentation and reporting
Degree of independence	Metrics to be collected	Test design technique	Retesting and regression testing	Suspension and resumption criteria

Entry and Exit criteria

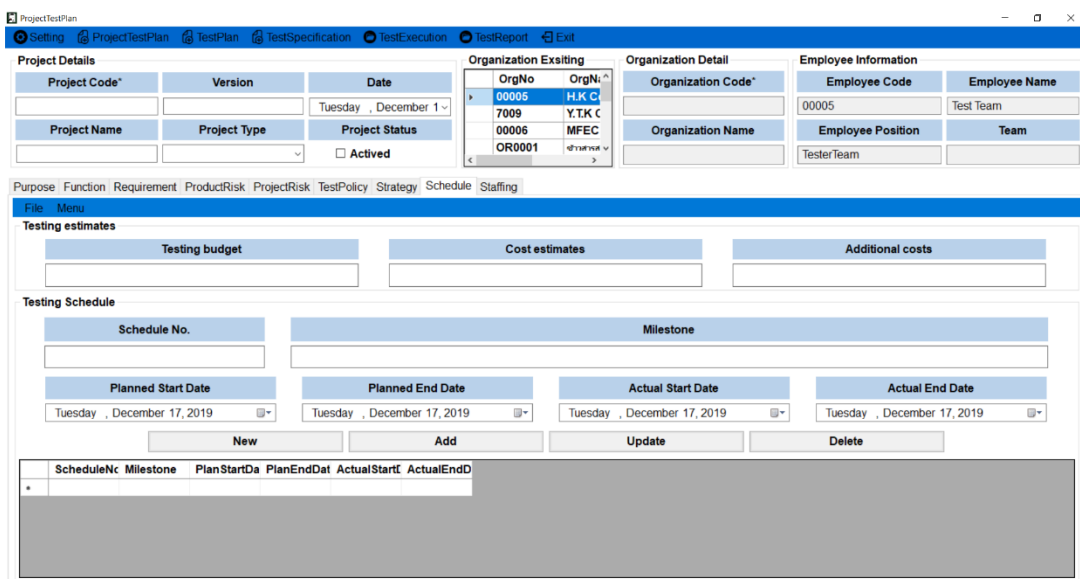
Entry criteria	Exit criteria

Test Environment Requirements

Hardware Needs	Software Needs	Testing Tools

รูปที่ 4.25 โปรแกรมทดสอบในส่วนกลยุทธ์ (Strategy)

8. ฟังก์ชัน Schedule เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.26



Project Test Plan

Setting Project Test Plan Test Plan Test Specification Test Execution Test Report Exit

Project Details

Project Code*	Version	Date
		Tuesday , December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Existing

OrgNo	OrgNi
00005	H.K.C
7009	Y.T.K.C
00006	MFEC
OR0001	

Organization Detail

Organization Code*	Employee Code	Employee Name
	00005	Test Team
Organization Name	Employee Position	Team
	TesterTeam	

Employee Information

Purpose Function Requirement ProductRisk ProjectRisk TestPolicy Strategy Schedule Staffing

File Menu

Testing estimates

Testing budget	Cost estimates	Additional costs

Testing Schedule

Schedule No.	Milestone		
Planned Start Date	Planned End Date	Actual Start Date	Actual End Date
Tuesday , December 17, 2019	Tuesday , December 17, 2019	Tuesday , December 17, 2019	Tuesday , December 17, 2019
New	Add	Update	Delete

ScheduleNo	Milestone	PlanStartDa	PlanEndDat	ActualStartE	ActualEndD
*					

รูปที่ 4.26 โปรแกรมทดสอบในส่วน Schedule

9. ฟังก์ชัน Staffing เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.27

Project TestPlan

Setting | ProjectTestPlan | TestPlan | TestSpecification | TestExecution | TestReport | Exit

Project Details

Project Code*	Version	Date
		Tuesday, December 1
Project Name	Project Type	Project Status
		<input type="checkbox"/> Activated

Organization Exiting

OrgNo	OrgName
00005	H.K.C.
7009	Y.T.K.C.
00006	MFEC
OR0001	...

Organization Detail

Organization Code*	Employee Code	Employee Name
	00005	Test Team
Organization Name	Employee Position	Team
	TesterTeam	

Employee Information

Employee Code	Employee Name
00005	Test Team
Employee Position	Team
TesterTeam	

Purpose: Function | Requirement | ProductRisk | ProjectRisk | TestPolicy | Strategy | Schedule | Staffing

File | Menu

Staffing and Training Needs

Staffing and Training Code*	Staffing and Training Name	Requirements	
Department	Training Details	Tasks	Note
Module	Detail	Estimate time	Cost

Team Responsible

No.	Team Name	Role	Responsible Person

New | Add | Update | Delete

No	TeamName	Role	Responsible
*			

รูปที่ 4.27 โปรแกรมทดสอบในส่วน Staffing

4.1.6 กำหนดข้อมูลแผนการทดสอบ (Define Test Plan)

ส่วนการกำหนดแผนการทดสอบ เมื่อทำการกดเมนู Test Plan จะปรากฏหน้าต่างดังรูปที่ 4.28 ซึ่งประกอบด้วยฟังก์ชันย่อย 7 ฟังก์ชัน คือ Item Test, Project Risk, Product Risk, Strategy Function, Schedule Function และ Staffing Function แสดงรายละเอียดในแต่ละฟังก์ชัน ดังนี้

1. Item Test Function เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.28 โดยผู้ใช้งานต้องทำการเลือก Project Code ในตาราง Test Project

Test Plan

Setting | ProjectTestPlan | TestPlan | TestSpecification | TestExecution | TestReport | Exit

Project Details

Test Plan Code*	Project Code*	Version	Date
			Tuesday, December 17, 201
Test Plan Name	Project Name	Project Type	Project Status
			<input type="checkbox"/> Activated

Organization Detail

Organization Code*	Employee Code
	00005
Organization Name	Team Code

Employee Information

Employee Code	Team Code
00005	

ItemTest | ProductRisk | ProjectRisk | Strategy | Schedule | Staffing

File | Menu

Test Project

ProjectCode	ProjectName	Organization	OrgName
FR0001	Testing Re.	OG0001	LACON C.
TT0001	Test	1121.0001	E.T.C.Com.

Enter

Test Item Grouping function name

Test Item Code*	Test Item Name	Priority	Note
		Find	
Function Code	Function Name	Requirement Type	Requirement

New | Add | Update | Delete

TestItemCo	TestItemNa	FunctionCo	FunctionNa	Requiremer	Requiremer	Priority	Note
*							

รูปที่ 4.28 กำหนดฟังก์ชัน

เมื่อทำการเลือก Project Code เรียบร้อยแล้ว ระบบจะแสดงข้อมูล Function และ Requirement ที่ได้สร้างไว้ ซึ่งแสดงดังรูปที่ 4.29

The screenshot shows the Test Plan application interface. The top navigation bar includes: Settings, ProjectTestPlan, TestPlan, TestSpecification, TestExecution, TestReport, and Exit. The main content area is divided into several sections:

- Project Details:** Fields for Test Plan Code* (FR0001), Project Code* (FR0001), Version, Date (Tuesday, December 17, 2011), Test Plan Name (Testing Registration System), Project Name (Testing Registration System), Project Type, and Project Status (Active).
- Organization Detail:** Fields for Organization Code* (OG0001), Organization Name (LACON.CO.TH), Employee Code (00005), and Team Code.
- Employee Information:** Fields for Employee Code (00005) and Test Team.
- Test Item Grouping function name:** A table with columns: Test Item Code*, Test Item Name, Priority, and Note. Below this table are buttons for New, Add, Update, and Delete.
- Function and Requirement Lists:** Two lists on the left side of the interface, one for Functions (e.g., Login Page, Add New, Edit Member) and one for Requirements (e.g., Functional, SA Requirement, User Stack).

รูปที่ 4.29 Function และ Requirement ของระบบ

เมื่อทำการเลือก Function และ Requirement แล้ว ผู้ใช้งานต้องทำการกำหนดลำดับความสำคัญของ Test Item โดยกดปุ่ม Find ในช่อง Priority แสดงดังรูปที่ 4.30

The screenshot shows the Test Plan application interface with a dialog box open for selecting a priority value. The dialog box is titled "Priority Group" and contains a table with the following data:

PriorityName	PriorityValue
low	0
super high	4
low	1
medium	2
high	3

The dialog box also has "Select" and "Cancel" buttons at the bottom.

รูปที่ 4.30 กำหนดลำดับความสำคัญของ Test Item

2. Project Risk เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.31

Project Details

Test Plan Code*	Project Code*	Version	Date
	FR0001		Tuesday , December 17, 201*
Test Plan Name	Project Name	Project Type	Project Status
	Testing Registration System		<input type="checkbox"/> Activated

Organization Detail

Organization Code*	Employee Code
OG0001	00005
Organization Name	Team Code
LACON, CO. TH	Test Team

Employee Information

ItemTest ProductRisk ProjectRisk Strategy Schedule Staffing

File Menu

Product Risk register

Product Risk Code :

Notification
The following abbreviations are used in the risk tables: I = Impact or effect if risk materialises P = Probability or likelihood of the risk E = Exposure = Probability x Impact

The scales for both probability and impact will be 1-4, where 4 is the highest.

Risk Code*	Risk Details	Mitigation Activities
<input type="text"/>	<input type="text"/>	<input type="text"/>
Probability (P)	Impact (I)	Exposure (E)
<input type="text"/>	<input type="text"/>	<input type="text"/>

New Add Update Delete

RiskCode	RiskDetail	Mitigation	Probability	Impact	Exposure
**					

รูปที่ 4.31 กำหนดความเสี่ยงของโปรเจกต์

3. Product Risk เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.32

Project Details

Test Plan Code*	Project Code*	Version	Date
	FR0001		Tuesday , December 17, 201*
Test Plan Name	Project Name	Project Type	Project Status
	Testing Registration System		<input type="checkbox"/> Activated

Organization Detail

Organization Code*	Employee Code
OG0001	00005
Organization Name	Team Code
LACON, CO. TH	Test Team

Employee Information

ItemTest ProductRisk ProjectRisk Strategy Schedule Staffing

File Menu

Product Risk Register

Project Risk Code :

Notification
D = Development T = Test

Risk Code*	Consequences	Prevention	Mitigation
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

New Add Update Delete

RiskCode	Consequen	Prevention	Mitigate
*			

รูปที่ 4.32 กำหนดความเสี่ยงของผลิตภัณฑ์

4. Strategy เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.33

Project Details

Test Plan Code*	Project Code*	Version	Date	Organization Code*	Employee Code
	FR0001		Tuesday , December 17, 2019	OG0001	00005
Test Plan Name	Project Name	Project Type	Project Status	Organization Name	Team Code
	Testing Registration System		<input type="checkbox"/> Activated	LACON. CO. TH	Test Team

Strategy

Project Strategy Code*	Project Strategy Name	Test deliverables	Test completion criteria	Test documentation and reporting
Degree of independence	Metrics to be collected	Test design technique	Retesting and regression testing	Suspension and resumption criteria

Entry and Exit criteria

Entry criteria	Exit criteria

Test Environment Requirements

Hardware Needs	Software Needs	Testing Tools

รูปที่ 4.33 กำหนดกลยุทธ์การทดสอบ

5. Schedule Function เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.34

Project Details

Test Plan Code*	Project Code*	Version	Date	Organization Code*	Employee Code
	FR0001		Tuesday , December 17, 2019	OG0001	00005
Test Plan Name	Project Name	Project Type	Project Status	Organization Name	Team Code
	Testing Registration System		<input type="checkbox"/> Activated	LACON. CO. TH	Test Team

Testing estimates

Testing budget	Cost estimates	Additional costs

Testing Schedule

Schedule No.	Milestone		
Planned Start Date	Planned End Date	Actual Start Date	Actual End Date
Tuesday , December 17, 2019	Tuesday , December 17, 2019	Tuesday , December 17, 2019	Tuesday , December 17, 2019
New		Add	
Update		Delete	

ScheNo	Milestone	PlanStart	PlanEnd	ActualStart	ActualEnd
*					

รูปที่ 4.34 กำหนดวัน เวลา กิจกรรม ในแผนการทดสอบ

6. Staffing Function เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.35

The screenshot shows the 'Test Plan' window with the following sections:

- Project Details:**

Test Plan Code*	Project Code*	Version	Date
	FR0001		Tuesday, December 17, 201*
Test Plan Name	Project Name	Project Type	Project Status
	Testing Registration System		<input type="checkbox"/> Activated
- Organization Detail:**

Organization Code*	Employee Code
OG0001	00005
Organization Name	Team Code
LACON. CO. TH	Test Team
- Staffing and Training Needs:**

Staffing and Training Code*	Staffing and Training Name	Requirements	
Department	Training Details	Tasks	Note
Module	Detail	Estimate time	Cost
- Team Responsible:**

No.	Team Name	Role	Responsible Person

Buttons: New, Add, Update, Delete

รูปที่ 4.35 กำหนดบทบาทหน้าที่การทดสอบในแผนการทดสอบ

4.1.7 กำหนดข้อมูลเฉพาะสำหรับการทดสอบ (Define Test Specification)

เมื่อเข้าสู่ฟังก์ชัน Test Specification เรียบร้อยแล้ว จะปรากฏหน้าจอแสดงผลดังรูปที่ 4.36 ซึ่งประกอบด้วยฟังก์ชัน คือ Features Function, Test Condition Function, Test Coverage Function, Test Case Function, Environment Function และ Test Procedure Function แสดงรายละเอียดในแต่ละฟังก์ชัน ดังนี้

1. Features Function เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.36 ผู้ใช้งานต้องทำการเลือก Test Plan Code ในตาราง Test Project

The screenshot shows the 'Test Specification' window with the following sections:

- Project Details:** (Same as in Figure 4.35)
- Test Project:**

TestPlanCo	TestPlanNa	ProjectC
F-TP001	Login Syst...	FR0001

Buttons: Add
- Set Features Group:**

Features Set Code*	Features Set Name	Objective	Approach
Test Item Code	Function (Test) Name	Requirement (Test)	Priority

Buttons: New, Add, Update, Delete
- Item:**

TestItemCo	FunctionTe	Requiremer	PriorityFeat

รูปที่ 4.36 คุณสมบัติการทดสอบ

เมื่อทำการเลือก Test Plan Code เรียบร้อยแล้ว ผู้ใช้งานทำการกดปุ่ม Add เพื่อแสดง Test Item ที่ต้องการนำไปใช้ แสดงดังรูปที่ 4.37

รูปที่ 4.37 ตั้งค่า Feature ที่ต้องการทดสอบ

2. Test Condition เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.38 โดยผู้ใช้งานต้องทำการกดเลือก Feature ที่ต้องการทดสอบจากราย Feature

รูปที่ 4.38 กำหนดเงื่อนไขการทดสอบ

3. Test Coverage เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.33 โดยผู้ใช้งานต้องทำการเลือก Condition ที่ต้องการตั้งค่าเพิ่มเติมจากราย Condition

The screenshot shows the 'TestSpecification' application interface. At the top, there are tabs for 'TestProject', 'TestPlan', 'TestSpecification', 'TestExecution', and 'TestReport'. Below the tabs, there are sections for 'Project Details', 'Organization Detail', and 'Employee Information'. The 'Project Details' section includes fields for Test Plan Code (FTP001), Project Code (FR0001), Version, Date (Tuesday, December 17, 2011), Test Plan Name (Login System in Member), Project Name (Testing Registration System), Project Type, and Project Status (Active). The 'Organization Detail' section includes Organization Code (OG0001), Organization Name (LACON CO. TH), and Employee Code (00005). The 'Employee Information' section includes Employee Code (00005) and Test Team. Below these sections, there is a 'Features' menu with options for 'TestCondition', 'TestCoverRage', 'TestCase', 'Environment', and 'TestProcedure'. The 'TestCondition' form is displayed, featuring an 'Add' button and a list of conditions on the left. The list includes 'mn', 'employee', 'Alpha', 'manager', 'user', and 'space'. The form fields include 'Test Condition No', 'Condition', and 'Cover in test coverage number (Ref condition)'. At the bottom, there are buttons for 'New', 'Add', 'Update', and 'Delete', and a table with columns for 'TestCoverage', 'Condition', and 'ReferenceTest'.

รูปที่ 4.39 กำหนดขั้นตอนการทดสอบ

4. Test Case เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.40 โดยผู้ใช้งานต้องทำการเลือก Test Condition ที่ต้องการนำมาสร้างเป็นกรณีทดสอบจากราย Test Condition ซึ่งจะมีตาราง Re-Executed Test Case ที่แสดงกรณีทดสอบที่ต้องนำมาทดสอบซ้ำ ซึ่งผู้ใช้งานต้องทำการกดปุ่ม Find ที่ช่อง Priority เพื่อตั้งค่าลำดับให้แก่กรณีทดสอบ

The screenshot shows the 'TestSpecification' application interface. At the top, there are tabs for 'TestProject', 'TestPlan', 'TestSpecification', 'TestExecution', and 'TestReport'. Below the tabs, there are sections for 'Project Details', 'Organization Detail', and 'Employee Information'. The 'Project Details' section includes fields for Test Plan Code (FTP001), Project Code (FR0001), Version, Date (Tuesday, December 17, 2011), Test Plan Name (Login System in Member), Project Name (Testing Registration System), Project Type, and Project Status (Active). The 'Organization Detail' section includes Organization Code (OG0001), Organization Name (LACON CO. TH), and Employee Code (00005). The 'Employee Information' section includes Employee Code (00005) and Test Team. Below these sections, there is a 'Features' menu with options for 'TestCondition', 'TestCoverRage', 'TestCase', 'Environment', and 'TestProcedure'. The 'Test Case' form is displayed, featuring an 'Add' button and a list of test conditions on the left. The list includes 'mn', 'employee', 'Alpha', 'manager', 'user', and 'space'. The form fields include 'Test Case Code', 'Test Case Name', 'Priority', 'Test Condition No', 'Purpose', 'Preconditions', 'Input', 'Expected Result', and 'Status'. At the bottom, there are buttons for 'New', 'Add', 'Update', and 'Delete', and a table with columns for 'TestCaseCoc', 'TestCaseNa', 'Priority', 'TestConditio', 'Purpose', 'PreCondition', 'Input', 'ExpectedRes', and 'Status'.

รูปที่ 4.40 กำหนดกรณีทดสอบ

4.1.8 ส่วนทดสอบ (Test Execution)

เมื่อทำการเข้าสู่ฟังก์ชันส่วนการทดสอบเรียบร้อยแล้ว จะปรากฏหน้าจอดังรูปที่ 4.37 ในส่วนนี้ผู้ใช้งานต้องไปทำการทดสอบ ประกอบด้วยเมนูย่อย คือ Inform Test Result, Test Incident และ Test Complete จะปรากฏหน้าจอดังรูปที่ 4.43

1. Inform Test Result Inform Test Result เพื่อรายงานผลการทดสอบ เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.43 โดยผู้ใช้งานระบบต้องการเลือก Procedure ที่ต้องการทดสอบ จากตาราง Test Procedure จากนั้นทำการกดปุ่ม >>> เพื่อให้ระบบแสดง Test Case จากตาราง Test Case Details แสดงดังรูปที่ 4.44

The screenshot shows the 'Inform Test Result' application window. The 'TestResultDetail' section contains the following data:

Test Procedure Code	Organization Code	Project Code	Employee Code
			00005
Object and Priority	Organization Name	Project Name	Employee Name

The 'Test Procedure' table is as follows:

TestProcedi	ObjectandP	RelatewithC	Initial	Organizatio	ProjectCode	ProjectNam
FFF01	Login	no	Login	LACON. C...	FR0001	Testing R...
Proce01	Login	no	login to s...	LACON. C...	FR0001	Testing R...

The 'Test Case Details' section is currently empty.

Buttons at the bottom: New, Add, Update, Clear.

รูปที่ 4.43 รายงานผลการทดสอบ

The screenshot shows the 'Inform Test Result' application window with the 'Test Case Details' section populated. The 'TestResultDetail' section contains the following data:

Test Procedure Code	Organization Code	Project Code	Employee Code
Proce01	login to system	FR0001	00005
Object and Priority	Organization Name	Project Name	Employee Name
Login	LACON. CO. TH	Testing Registration System	Test Team

The 'Test Procedure' table is as follows:

TestProcedi	ObjectandP	RelatewithC	Initial	Organizatio	ProjectCode	ProjectNam
FFF01	Login	no	Login	LACON. C...	FR0001	Testing R...
Proce01	Login	no	login to s...	LACON. C...	FR0001	Testing R...

The 'Test Case Details' table is as follows:

Stepno	Activities	Input	ExpectedResi
TCFST01.01	manager สามารถใช้กรณมี...	manager in passw...	ใช้กรณีใช้กรณมี...
TCFST01.02	manager สามารถใช้กรณมี...	employee in pass...	ใช้กรณีใช้กรณมี...
TCFST01.04	ผู้ใช้ระบบสามารถค้นหา...	a, A, b, B	ค้นหาเอกสาร

Buttons at the bottom: New, Add, Update, Clear.

รูปที่ 4.44 กรณีทดสอบรายงานผลการทดสอบ

3. Test Complete Test Complete เมื่อทำการกดแถบเมนูเรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.42 โดยผู้ใช้งานระบบต้องทำการเลือก Test Procedure ที่เสร็จสมบูรณ์ พร้อมทั้งทำรายงานสถานะสมบูรณ์ จากตาราง Test Complete Details โดยข้อมูลภายในตารางนี้จะมาจากฟังก์ชัน Approval ที่ผู้จัดการโปรเจกต์ต้องทำการตรวจสอบ และยืนยันสถานะของโปรเจกต์ หากโปรเจกต์ใดทดสอบเสร็จสมบูรณ์แล้ว แต่ยังไม่ได้รับการอนุมัติจากผู้จัดการโปรเจกต์ ข้อมูลจะไม่แสดงในฟังก์ชันนี้ เมื่อผู้ใช้งานเลือก Test Procedure เสร็จแล้ว กดปุ่ม >> เพื่อแสดงรายละเอียดของ Test Procedure นั้น ดังรูปที่ 4.48

Test Case Code	Organization Code	Project Code	Test Procedure Code	Employee Code
			2	00005
Test Case Name	Organization Name	Project Name	Test Procedure Objective	Employee Name

Test Complete Details

TestProcedureCode
2
ICCTP.001
ME1.1.1
Proce01
TPC001

Project Status:

Test Complete Report Version:

Approved by:

Written by:

Summary of testing performed:

Deviations from planned testing:

Test completion evaluation:

Factors that blocked progress:

Residual risks:

Test deliverables:

Reusable test assets:

Lessons learned:

รูปที่ 4.47 รายงานผลการทดสอบ

Test Case Code	Organization Code	Project Code	Test Procedure Code	Employee Code
ICCTC001	ICC001	ICCP001	ICCTP.001	00005
Test Case Name	Organization Name	Project Name	Test Procedure Objective	Employee Name
POS	ICC Company	ทดสอบระบบของ ICC	ทดสอบการดำเนินงานในแบบปกติ	

Test Complete Details

TestProcedureCode
2
ICCTP.001
ME1.1.1
Proce01
TPC001

TestCaseNo	Activities	TestCaseName	Project Status	ProjectCode	ProjectName	OrganizationCode
ICCTC001	ทดสอบการใช้งาน	POS	Active	ICCP001	ทดสอบระบบของ ICC	ICC001

Project Status:

Test Complete Report Version:

Approved by:

Written by:

Summary of testing performed:

Deviations from planned testing:

Test completion evaluation:

Factors that blocked progress:

Residual risks:

Test deliverables:

Reusable test assets:

Lessons learned:

รูปที่ 4.48 รายงานผลการทดสอบส่วนกรณีทดสอบ

4.1.9 การวัดประสิทธิภาพ (Measurement)

ภายในระบบมีการวัดประสิทธิภาพ 2 ประเภท คือ Normative Requirement และ Software Testing Metrics ซึ่งรายละเอียดในแต่ละประเภทแสดงดังนี้

1. Normative Requirement การวัดประสิทธิภาพในประเภทนี้ จะเป็นแบบอัตโนมัติจากระบบ โดยเมื่อผู้ใช้งานกรอกข้อมูลลงในช่องต่าง ๆ ทางระบบจะทำการประเมิน พร้อมทำการแจ้งเตือน แสดงผลดังรูป 4.49

The screenshot shows the ProjectTestPlan application window. At the top, there are tabs for Settings, ProjectTestPlan, TestPlan, TestSpecification, TestExecution, and TestReport. Below the tabs, there are several data entry sections:

- Project Details:** Project Code* (TS001), Version (1.0.0.1), Date (Tuesday, December 1), Project Name (Test), Project Type (AddNewProject), Project Status (Active).
- Organization Exsiting:** A list of organizations with columns for OrgNo and OrgName. Selected entries include 00005 H.K.C., 7009 Y.T.K.C., 00006 MFEC, and OR0001.
- Organization Detail:** Organization Code* (00005), Organization Name (H.K. Company).
- Employee Information:** Employee Code (00005), Employee Name (Test Team), Employee Position (Tester Team), Team (Interface Test Team).
- Purpose of Project Test Plan:** System Name (POS), Purpose (Test POS), Tools (Normative Requirement), Automation Test, Description (Test).
- Scope of Project Test Plan:** Scope No* (1), Test Type (UI Test), Test (Test), Responsible (Tester Team).

A dialog box titled 'Normative Requirement' is open in the center, showing 'Normative Requirement Value at 90' and an 'OK' button. Below the dialog box, there are buttons for 'New', 'Add', 'Update', and 'Delete'. At the bottom, there is a table with columns: ScopeNo, TestType, Scope, and Responsibility.

รูปที่ 4.49 ผล Normative Requirement

2. Software Testing Metrics วัดประสิทธิภาพของกรณีทดสอบที่ได้สร้างขึ้น โดยผู้ใช้งานกดปุ่ม Measurement ที่หน้าเมนูหลัก เมื่อเข้าสู่ฟังก์ชัน Measurement เรียบบร้อยปรากฏหน้าจอดังรูปที่ 4.50

ผู้ใช้งานระบบต้องทำการเลือก Test Procedure ที่ต้องการวัดค่าประสิทธิภาพ เมื่อผู้ใช้งานทำการเลือก Test Procedure เรียบบร้อยแล้ว กดปุ่ม >> ทางระบบทำการแสดง Test Case ทั้งหมดภายใต้ Test Procedure ที่เลือกไว้ แสดงดังรูปที่ 4.51

Test Status Code	Description	Totals
	Total Number of Test Cases Executed	
	Number of Test Cases Passed	
	Number of Test Cases Failed	
	No. of Test cases not executed	
	No. of Test cases Blocked	
	No. of Test cases Re-Test	
	No. of Test cases has Incident	
	No. of Test cases written	

รูปที่ 4.50 วัดค่าประสิทธิภาพ

Test Status Code	Description	Totals
	Total Number of Test Cases Executed	
	Number of Test Cases Passed	
	Number of Test Cases Failed	
	No. of Test cases not executed	
	No. of Test cases Blocked	
	No. of Test cases Re-Test	
	No. of Test cases has Incident	
	No. of Test cases written	

รูปที่ 4.51 วัดค่าประสิทธิภาพในส่วนกรณีทดสอบ

เมื่อผู้ใช้งานทำการเลือก Test Procedure และ Test Case เรียบร้อยแล้ว ผู้ใช้งานต้องกดปุ่ม Search เพื่อทำการวัดประสิทธิภาพ หากไม่ต้องการสามารถกดปุ่ม Clear เพื่อล้างข้อมูลที่เลือกไว้ เมื่อทำการตรวจสอบข้อมูลเรียบร้อยแล้ว ผู้ใช้งานต้องทำการกรอก Test Status Code พร้อมทั้งบันทึกผลข้อมูลการวัดประสิทธิภาพ แสดงดังรูปที่ 4.52

ผู้ใช้งานสามารถกดแถบ Test Measures เพื่อดูประสิทธิภาพการวัดกรณีทดสอบ แสดงดังรูปที่ 4.53

Test Status Code	Description	Totals
Total Number of Test Cases Executed		5
Number of Test Cases Passed		2
Number of Test Cases Failed		1
No. of Test cases not executed		1
No. of Test cases Blocked		1
No. of Test cases Re-Test		14
No. of Test cases has Incident		14
No. of Test cases written		7

รูปที่ 4.52 สถานะการดำเนินงานของกรณีทดสอบ

1. Test Case Executed	No. of Test cases executed	5	5. Test Cases Blocked	No. of Test cases Blocked	1
	Total no. of Test cases written	7		Total no. of Test cases Executed	5
	Test cases Executed	71 4285714285714		Test cases Blocked	20
2. Test Case Not Executed	No. of Test cases not executed	1	6. Number of re-executed test cases	No. of Test cases re-executed	14
	Total no. of Test cases written	7		Total no. of Test cases Executed	5
	Test cases not executed	14 2857142857143		Total Test Case re-executed	2.8
3. Test Cases Passed	No. of Test cases Passed	2	7. Number of incidents per category	No. of Test cases Incident	14
	Total no. of Test cases Executed	5		Total no. of Test cases Executed	5
	Test cases Passed	40		Total Incident	2.8
4. Test Cases Failed	No. of Test cases Failed	1	8. Number of solved incidents per category	No. of Test cases Solved	5
	Total no. of Test cases Executed	5		Total no. of Test cases Incident	14
	Test cases Failed	20		Test solved incidents	0.357142857142857

รูปที่ 4.53 แสดงผลการวัดประสิทธิภาพ

4.1.10 การอนุมัติการทดสอบ (Approval)

ผู้ใช้งานทำการกดปุ่ม Approval ที่หน้าเมนูหลัก เมื่อทำการเข้าสู่ฟังก์ชัน Approved เรียบร้อยแล้ว จะปรากฏดังรูปที่ 4.54 โดยฟังก์ชันนี้ผู้จัดการโครงการสามารถใช้งานได้เพียงผู้เดียว

The screenshot shows the 'TasksApproval' application window. At the top, there is a menu bar with 'Menu' and 'Exit'. Below it, a section titled 'Tasks is working' contains a table with columns: TestProced, Stepno, and Activities. The table has several rows, with 'TPC001' selected. Below the table is a 'Project Details' section with fields for Approval Code, Approval Date, Test Procedure No, Test Case No, Test Case Activities, and Employee Code. There are also fields for Approved By, Position, Approval Status, and Add Comments. At the bottom of this section are buttons for New, Add, Update, and Delete.

TestProced	Stepno	Activities
ME1.1.1	TCSF1.1	ขานแบบไม่คิด...
2	tcww	ddd
Proce01	TCFST01....	manager ส...
Proce01	TCFST01....	สำหรับกรที่ต่างกั...
TPC001	TC(FS1.2_3)	input over...
TPC001	TC(FS1.3)	input som...

Approval Code *	Approval Date *	Test Procedure No	Test Case No	Test Case Activities	Employee Code
	Wednesday, Decemb	TPC001	TC(FS1.2_3)	input over data	00006

Approved By	Approval Status

Position	Add Comments

Buttons: New, Add, Update, Delete

รูปที่ 4.54 การอนุมัติผลการทดสอบ

4.1.11 ส่วนแสดงผลรายงาน (Report)

ผู้ใช้งานทำการกดปุ่ม Report ที่หน้าเมนูหลัก เมื่อทำการเข้าสู่ฟังก์ชัน Report เรียบร้อยแล้ว จะปรากฏหน้าจอจดังรูปที่ 4.55

The screenshot shows the 'Report' application window. At the top, there is a menu bar with 'Menu' and 'Exit'. Below it, a section titled 'Project Details' contains fields for Organization Code, Test Plan Code, Test Case Code, TestProcedure, Test Status Code, and Employee Code. Below this is a table with columns: Stepno, TestProcedureNo, TestPlanCode, and OrganizationCode. To the right of the table is a grid of report options: Organizational Report, Incident Report, Test Plan Report, Test Status Report, Test Case Report, and Complete Report.

Organization Code	Test Plan Code	Test Case Code	TestProcedure	Test Status Code	Employee Code
OG0001	FTP001	TCFST01.02	Proce01		00005

Stepno	TestProcedureNo	TestPlanCode	OrganizationCode
TCFST01.01	FTPPro01	FTP001	OG0001
TCFST01.01	Proce01	FTP001	OG0001
TCFST01.02	FTPPro01	FTP001	OG0001
TCFST01.02	Proce01	FTP001	OG0001
TCFST01.04	FTPPro01	FTP001	OG0001
TCFST01.04	Proce01	FTP001	OG0001

TestStatusCode	ProjectCode	OrganizationCode
FFTS001	FR0001	OG0001

Report Options:

- Organizational Report
- Incident Report
- Test Plan Report
- Test Status Report
- Test Case Report
- Complete Report

รูปที่ 4.55 เมนูหลักของการรายงานผล

ซึ่งภายในฟังก์ชัน Report แสดงเมนูย่อย ประกอบด้วย Organizational Report, Test Plan Report, Test Case Report, Incident Report, Test Status Report และ Complete Report

1. Organization Report เมื่อกดปุ่ม Organizational Report เรียบร้อยแล้ว จะปรากฏหน้าจอดังรูปที่ 4.56

Organization Name	Version	Date
LACON. CO. TH	1	12/8/2019 3:04:57 AM

Organization Details		
Sub Organization Name	CEO	Technique
-	ALEX LAZAN	Technique team in company
Telephone	Mobile	Fax
+8778644	+898888855	+29999999
Coordinator	Note	Email1
User in company	-	LACON@gmail.com
Address	Email2	
1211	hr@lacon.co.th	

Organization Policy	
Policy Process	ทดสอบการใช้งานของผลิตภัณฑ์
Policy Objective	จนสามารถใช้งานได้
Policy Organization	ทดสอบจนสามารถใช้งานได้สมบูรณ์
Policy Evaluation	ไม่พบข้อผิดพลาด
Policy Stand	ISO 9001:ISO 29119

รูปที่ 4.56 เมนูพิมพ์รายงานขององค์กร

โดยผู้ใช้งานต้องทำการกรอกรหัสองค์กร (Organization Code) และ ชื่อองค์กร (Organization Name) เมื่อกรอกข้อมูลเรียบร้อยแล้วกดปุ่ม Enter ซึ่งตัวอย่างของรายงานองค์กรแสดงดังรูปที่ 4.57 และ รูปที่ 4.58

Organization Report

Organization Name	Version	Date
LACON. CO. TH	1	12/8/2019 3:04:57 AM

Organization Details		
Sub Organization Name	CEO	Technique
-	ALEX LAZAN	Technique team in company
Telephone	Mobile	Fax
+8778644	+898888855	+29999999
Coorfinator	Note	Email1
User in company	-	LACON@gmail.com
Address	Email2	
1211	hr@lacon.co.th	

Organization Policy

Policy Process	ทดสอบการทำงานของฟังก์ชัน
Policy Objective	ระบบสามารถใช้งานได้
Policy Organization	ทดสอบจน สามารถใช้งานได้สมบูรณ์
Policy Evaluation	นโยบายฟังก์ชัน
Policy Stand	ISO 9001, ISO 29119
Policy Approach	พัฒนาจากการประเมินประสิทธิภาพขององค์กร
Policy	ไม่มี

รูปที่ 4.57 ตัวอย่างรายงานข้อมูลองค์กร พร้อมนโยบายการทดสอบ

Organization Report

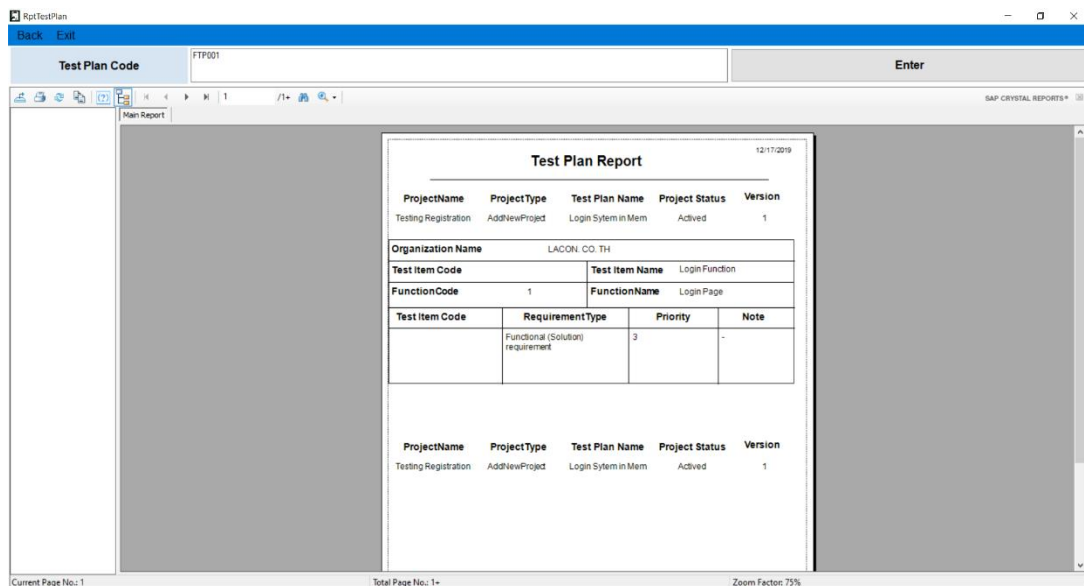
Organization Name	Version	Date
LACON. CO. TH	1	12/8/2019 3:04:57 AM

Organization Strategy	
Strategy Risk	หาแนวทางในการแก้ไขร่วมกับ tester team
Strategy Documentation	ต้องการเอกสารประกอบการทดสอบ
Strategy Selectection	เรียงตามลำดับการใช้งานในฟังก์ชัน
Strategy Automate	ใช้ manual test
Strategy Configuration	มีการฝึกอบรมการใช้งาน
Strategy Sub Process	ทดสอบทุกกระบวนการของระบบ
Strategy Incident	หากไม่เจอปัญหาใหญ่ tester team สามารถแก้ไขได้
Policy Ethics	ไม่เปิดเผยข้อมูลของลูกค้า

Page 2 of 2

รูปที่ 4.58 ตัวอย่างรายงานข้อมูลองค์กรในส่วนกลยุทธ์การทดสอบ

2. Test Plan Report เมื่อกดปุ่ม Test Plan Report เรียบร้อยแล้ว จะปรากฏหน้าจอจดังรูปที่ 4.59



รูปที่ 4.59 เมนูพิมพ์รายงานของแผนการทดสอบ

โดยผู้ใช้งานต้องทำการกรอกรหัสแผนการทดสอบ (Test Plan Code) เมื่อกรอกข้อมูลเรียบร้อยแล้วกดปุ่ม Enter ซึ่งตัวอย่างของรายงานแผนการทดสอบ แสดงดังรูปที่ 4.60

12/17/2019

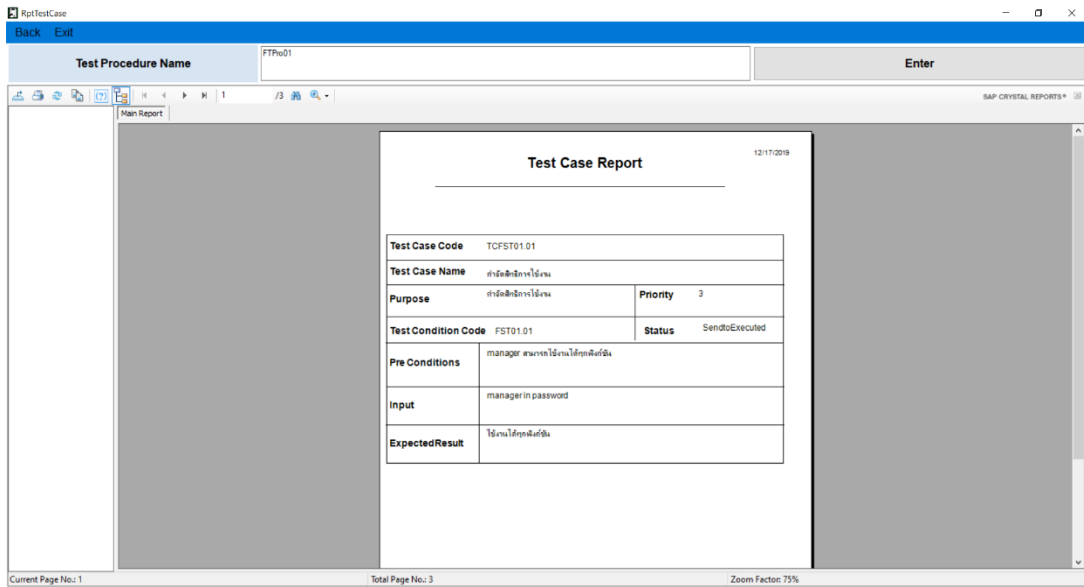
Test Plan Report

ProjectName	ProjectType	Test Plan Name	Project Status	Version
Testing Registration	AddNewProject	Login Sytem in Mem	Actived	1

Organization Name		LACON CO TH		
Test Item Code		Test Item Name	Login Function	
FunctionCode	1	FunctionName	Login Page	
Test Item Code	RequirementType	Priority	Note	
	Functional (Solution) requirement	3	-	

รูปที่ 4.60 ตัวอย่างรายงานแผนการทดสอบ

3. Test Case Report เมื่อกดปุ่ม Test Case Report เรียบร้อยแล้ว จะปรากฏหน้าจอ ดังรูปที่ 4.61



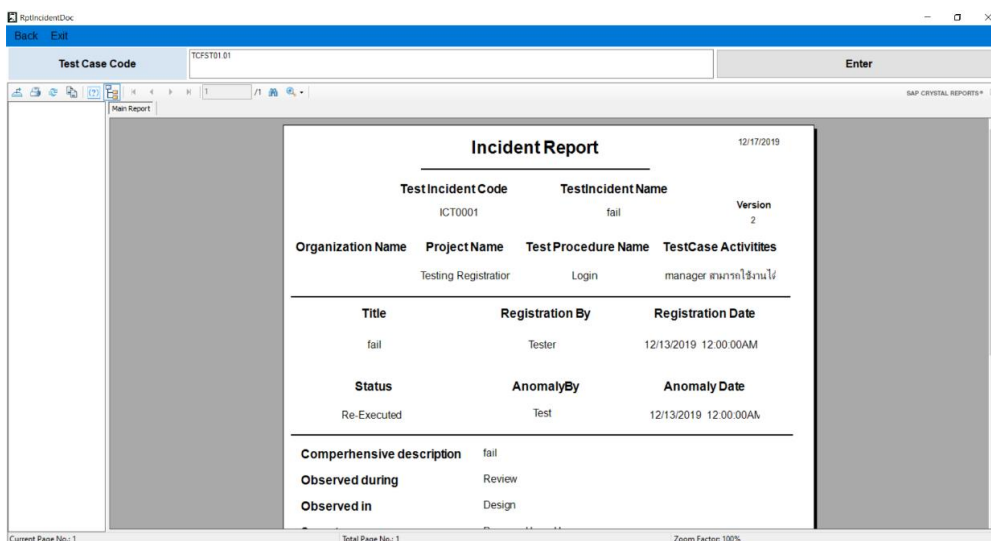
รูปที่ 4.61 เมนูพิมพ์รายงานของกรณีทดสอบ

โดยผู้ใช้งานต้องทำการกรอกรหัสขั้นตอนการทดสอบ (Test Procedure Code) เมื่อกรอกข้อมูลเรียบร้อยแล้วกดปุ่ม Enter ซึ่งตัวอย่างของรายงานกรณีทดสอบ แสดงดังรูปที่ 4.62

Test Case Report		12/17/2019	
Test Case Code	TCFST01.01		
Test Case Name	ก่าจัดลลลลลลลลลลลล		
Purpose	ก่าจัดลลลลลลลลลลลล	Priority	3
Test Condition Code	FST01.01	Status	SendtoExecuted
Pre Conditions	manager สามารถใช้งานได้ทุกฟังก์ชัน		
Input	manager in password		
ExpectedResult	ใช้งานได้ทุกฟังก์ชัน		

รูปที่ 4.62 ตัวอย่างรายงานกรณีทดสอบ

4. Incident Report เมื่อกดปุ่ม Incident Report เรียบร้อยแล้ว จะปรากฏหน้าจอ ดังรูปที่ 4.63



รูปที่ 4.63 เมนูพิมพ์รายงานของเหตุการณ์การทดสอบ

โดยผู้ใช้งานต้องทำการกรอกรหัสกรณีทดสอบ (Test Case Code) เมื่อกดปุ่มแล้วกรอกข้อมูลเรียบร้อยแล้วกดปุ่ม Enter ซึ่งตัวอย่างของรายงานเหตุการณ์การทดสอบ แสดงดังรูปที่ 4.64

Incident Report				12/17/2019
Test Incident Code		Test Incident Name		Version
ICT0001		fail		2
Organization Name	Project Name	Test Procedure Name	Test Case Activities	
Testing Registrator		Login	manager สามารถใช้งานได้	
Title		Registration By	Registration Date	
fail		Tester	12/13/2019 12:00:00AM	
Status		Anomaly By	Anomaly Date	
Re-Executed		Test	12/13/2019 12:00:00AM	
Comprehensive description	fail			
Observed during	Review			
Observed in	Design			
Symptom	Program Hang Up			
Impact	Medium			
Urgency	Medium			

รูปที่ 4.64 ตัวอย่างรายงานเหตุการณ์การทดสอบ

5. Test Complete Report เมื่อกดปุ่ม Test Complete Report เรียบร้อยแล้ว จะปรากฏหน้าจอดังรูปที่ 4.65

The screenshot displays a SAP Crystal Reports window titled 'TestComplete Report'. At the top, there is a 'Test Case Code' field containing 'TCFST01.04' and an 'Enter' button. The main report area shows the following data:

Organization Name	Project Name	Test Procedure Name
LACON.CO.TH	Testing Registration System	การติดตามกิจกรรมการเข้าใช้
Version	Approved By	Written By
2	ANCHIKA	Tester
Test Case Code	Test Case Name	Test Project Status
TCFST01.04	การเข้าระบบอัตโนมัติ	Active

Below the table is a section titled 'Test Complete Detail' with the following items:

1. Summary of testing performed : Complete
2. Deviations from planned testing : no
3. Test completion evaluation : follow testing system
4. Factors that blocked progress : no
5. Residual risks : no
6. Test deliverables : no

At the bottom of the report, it shows 'Current Page No.: 1', 'Total Page No.: 1', and 'Zoom Factor: 75%'.

รูปที่ 4.65 เมนูพิมพ์รายงานของการทดสอบที่สมบูรณ์

โดยผู้ใช้งานต้องทำการกรอกรหัสกรณีทดสอบ (Test Case Code) เมื่อกรอกข้อมูลเรียบร้อยแล้ว กดปุ่ม Enter ซึ่งตัวอย่างของรายงานการทดสอบสมบูรณ์ แสดงดังรูปที่ 4.66

12/17/2019

Test Complete Report

Organization Name	Project Name	Test Procedure Name
LACON. CO. TH	Testing Registration System	ตัวอักษรที่ต่างกันสามารถเข้าใช้งานได้ไ้
Version	Approved By	Written By
2	ANCHIKA	Tester
Test Case Code	Test Case Name	Test Project Status
TCFST01.04	การเข้าสู่ระบบมีข้อจำกัด	Actived

Test Complete Detail	
1. Summary of testing performed :	Complete
2. Deviations from planned testing :	no
3. Test completion evaluation :	follow testing system
4. Factors that blocked progress :	no
5. Residual risks :	no
6. Test deliverables :	no
7. Reusable test assets :	Test complete
6. Lessons learned :	n o

รูปที่ 4.66 แบบรายงานการทดสอบที่สมบูรณ์

6. Test Status Report เมื่อกดปุ่ม Test Status Report เรียบร้อยแล้ว จะปรากฏหน้าจอ ดังรูปที่ 4.67

Test Plan Code	Test Procedure Code	Test Status Code
FTP001	FFTS001	

Software Testing Metrics	
Total number of test case executed	3.00
Number of test cases passed	2.00
Number of test cases failed	1.00
Number of test cases not executed	1.00
Number of test cases has incident	5.00
Number of test cases Rn-Test	0.00
Number of test cases blocked	1.00
Number of test cases written	5.00

Software Testing Performance	
Test case executed	60.00
Test case not executed	20.00
Test cases passed	66.67
Test cases failed	33.33
Test cases blocked	33.33
Number of re-executed test cases	0.00
Number of incident per category	1.67
Number of solved incidents per category	0.60

รูปที่ 4.67 เมนูการพิมพ์รายงานของสถานการณ์ทดสอบ

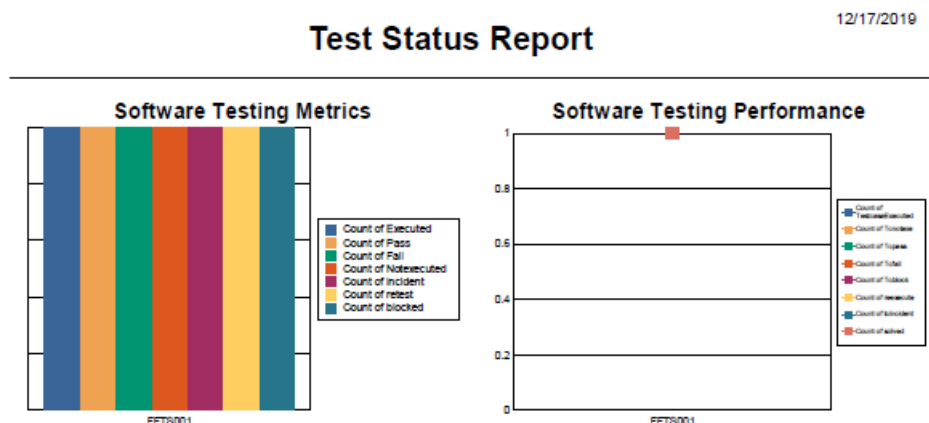
โดยผู้ใช้งานต้องทำการกรอกรหัสสถานะการทดสอบ (Test Status Code) เมื่อกรอกข้อมูลเรียบร้อยแล้วกดปุ่ม Enter ซึ่งตัวอย่างของรายงานสถานะการทดสอบ แสดงดังรูปที่ 4.68

12/17/2019

Test Status Report

Test Plan Code	Test Procedure Code	Test Status Code
FTP001		FFTS001
Software Testing Metrics		
Total number of test case executed		3.00
Number of test cases passed		2.00
Number of test cases failed		1.00
Number of test cases not executed		1.00
Number of test cases has incident		5.00
Number of test cases Re-Test		0.00
Number of test cases blocked		1.00
Number of test cases written		5.00
Software Testing Performance		
Test case executed		60.00
Test case not executed		20.00
Test cases passed		66.67
Test cases failed		33.33
Test cases blocked		33.33
Number of re-executed test cases		0.00
Number of incident per category		1.67
Number of solved incidents per category		0.60

รูปที่ 4.68 ตัวอย่างรายงานประสิทธิภาพการทดสอบ



รูปที่ 4.69 ตัวอย่างรายงานประสิทธิภาพการทดสอบแบบกราฟ

4.2 ผลแบบสำรวจความพึงพอใจของผู้ใช้งานระบบ

เมื่อหลังจากที่ผู้วิจัยได้สร้างระบบสร้างเอกสารประกอบการทดสอบซอฟต์แวร์สำเร็จแล้ว จึงได้ทำแบบสำรวจความพึงพอใจจากผู้ใช้งานระบบ โดยระดับของความพึงพอใจในการสำรวจประกอบด้วย มีความพึงพอใจมากที่สุด มีความพึงพอใจมาก มีความพึงพอใจปานกลาง มีความพึงพอใจน้อยและมีความพึงพอใจน้อยที่สุด ซึ่งแบบสำรวจความพึงพอใจของผู้ใช้งานระบบจะประกอบด้วย ความพึงพอใจด้านการใช้งาน ความพึงพอใจในการใช้บริการของระบบ และความพึงพอใจด้านความมีประสิทธิภาพ แสดงในตารางที่ 1

ผู้วิจัยได้ทำการสอบถามผู้ใช้งานระบบจำนวน 20 คน ซึ่งเป็นบุคคลที่เกี่ยวข้องกับงานด้านการทดสอบซอฟต์แวร์ การพัฒนาซอฟต์แวร์และระบบซอฟต์แวร์ จากการสำรวจสามารถสรุปได้ว่าด้านความพึงพอใจด้านการใช้งานอยู่ในระดับพึงพอใจมากและ 84 เปอร์เซ็นต์ ความพึงพอใจในการใช้บริการของระบบอยู่ในระดับพึงพอใจมากและ 83 เปอร์เซ็นต์ ด้านความพึงพอใจด้านความมีประสิทธิภาพอยู่ในระดับพึงพอใจมากและ 81 ดังนั้นระบบจัดเตรียมแผนการทดสอบซอฟต์แวร์มีค่าความพึงพอใจอยู่ที่ 82.67 เปอร์เซ็นต์

ตารางที่ 4.1 แบบสำรวจความพึงพอใจของผู้ใช้งานระบบ

ด้านการประเมิน	มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด
ความพึงพอใจด้านการใช้งาน		84			
1. ท่านมีความพึงพอใจในการใช้งานระบบโดยภาพรวมเท่าใด		84			
ความพึงพอใจในการใช้บริการของระบบ		83			
1. ระบบมีความพร้อมในการใช้งาน		83			
2. ระบบมีความครบถ้วนสมบูรณ์ของข้อมูล		85			
3. ข้อมูลภายในระบบมีความถูกต้องสมบูรณ์ และครบถ้วน		80			
4. ระบบมีความน่าเชื่อถือของข้อมูลในระบบ		81			

ด้านการประเมิน	มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด
5. ระบบมีความสะดวกในการใช้งาน		87			
6. ประโยชน์จากการใช้งานระบบ		86			
7. ระบบสามารถตรวจสอบได้		79			
ความพึงพอใจด้านความมีประสิทธิภาพ		81			
1. ระบบใช้งานได้ง่าย และไม่ซับซ้อน		82			
2. ระบบมีความเร็วในการตอบสนองต่อการใช้งาน		81			
3. ข้อมูลมีประโยชน์ต่อการใช้งาน		79			
4. ระบบมีการจัดระดับความปลอดภัย และมีการกำหนดสิทธิในการเข้าถึงข้อมูล		83			

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินการ

สามารถพัฒนาระบบทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119 ได้ตามเป้าหมาย ดังนี้

1) ส่วนกำหนดข้อมูลผู้ใช้งาน เป็นส่วนที่ควบคุมสิทธิการเข้าใช้งานระบบ สัมพันธ์กับการเข้าใช้งานในแต่ละฟังก์ชัน โดยป้องกันการเข้าถึงข้อมูลในบางส่วนกับผู้ใช้งานบางกลุ่ม เพื่อความปลอดภัยของข้อมูล และความเป็นส่วนตัวของลูกค้า

2) ส่วนกำหนดสมาชิกในโปรเจค เป็นส่วนที่จัดทำขึ้นเพื่อสร้างกลุ่ม และสมาชิกในทีมที่ต้องการดำเนินการทดสอบในแต่ละโปรเจคที่ได้รับมอบหมาย

3) ส่วนกำหนดข้อมูลลูกค้า เป็นส่วนที่จัดทำขึ้นเพื่อสร้างกลุ่มของลูกค้าที่ต้องการทดสอบระบบ โดยจัดเก็บข้อมูลเกี่ยวกับองค์กร หรือ ลูกค้า เป็นข้อมูลในเรื่องทั่วไป และข้อมูลเชิงนโยบายในการทดสอบ

4) ส่วนข้อมูลโปรเจคทดสอบ เป็นส่วนที่จัดทำขึ้นเพื่อสร้างแผนการทดสอบที่นำไปใช้ในการทดสอบ ทำการสกัดความต้องการของลูกค้า โมดูลที่ต้องการ นำมาสร้างแผนการทดสอบ กรณีทดสอบ

โดยระบบทดสอบซอฟต์แวร์อ้างอิงตามมาตรฐานการทดสอบซอฟต์แวร์ สามารถนำไปใช้ในส่วนการทดสอบของซอฟต์แวร์ โดยใช้ได้กับซอฟต์แวร์หลายประเภท ซึ่งช่วยลดระยะเวลาในการศึกษาข้อมูล ขั้นตอนและกระบวนการในการทดสอบซอฟต์แวร์ พร้อมทั้งยังลดต้นทุนในด้านการทดสอบซอฟต์แวร์ โดยระบบภายในงานวิจัยนี้มีการรวบรวมข้อมูล และรายละเอียดทั้งหมดของมาตรฐานการทดสอบซอฟต์แวร์ตามมาตรฐานสากลไว้เรียบร้อยแล้ว ซึ่งระบบสามารถตรวจสอบ และชี้แนวทางในการปรับปรุงซอฟต์แวร์ให้มีความถูกต้อง สมบูรณ์ มีคุณภาพ สามารถบอกระดับความจำเป็นของข้อมูล ทำให้การทดสอบซอฟต์แวร์สามารถทำได้ง่าย อีกทั้งสามารถวัดประสิทธิภาพในการทดสอบซอฟต์แวร์ เพื่อให้ทราบถึงคุณภาพของซอฟต์แวร์ และส่วนที่ต้องปรับปรุงก่อนส่งมอบหรือนำไปใช้งานซึ่งป้องกันการเกิด failure ของระบบได้อย่างดี พร้อมทั้งสามารถสร้างเอกสารทดสอบซอฟต์แวร์ที่ครอบคลุมทุกขั้นตอนการทดสอบและถูกต้องตามมาตรฐานในการทดสอบซอฟต์แวร์ ผู้ใช้งานสามารถนำไปใช้งานได้ถูกต้อง ครบถ้วน

5.2 ข้อเสนอแนะ

- 1) ควรทำการแปลงไฟล์รายงานได้หลายประเภทมากขึ้น เช่น สกุล .docx
- 2) มีประเภทของความต้องการที่ครอบคลุมมากกว่านี้
- 3) กำจัดสิทธิการใช้งานที่มากขึ้น เพื่อเพิ่มความปลอดภัยให้แก่ระบบ

เอกสารอ้างอิง

- [1] สำนักงานปลัดกระทรวงศึกษาธิการ. (2561). ยุค Digital 4.0 เมื่อโลกขับเคลื่อนด้วยเทคโนโลยี. สืบค้นเมื่อ 29 สิงหาคม 2562, จาก <http://www.ops.moe.go.th/ops2017/สารหน้ารู้/2876-ยุค-digital-4-0-เมื่อโลกขับเคลื่อนด้วยเทคโนโลยี-25-ก-ย-2561>.
- [2] P. Eira, R. J. Machado, and A. Silva, "Tailoring ISO/IEC/IEEE 29119-3 standard for small and medium-sized enterprises," IEEE Trans. On computer society, pp. 380-389, 2018.
- [3] E. Veenendaal, "TMMi and ISO/IEC 29119: Friends or Foes?" TMMi Foundation, pp. 1-12, March 2016.
- [4] The Home of International Standards for Software Testing. (2018). IEEE Std 29119, Part 1, Part 2, Part 3, Part 4, Part 5. สืบค้นเมื่อ 6 เมษายน 2562, จาก <https://www.softwaretestingstandard.org/index.php>.
- [5] A. Rana, A. Singh Rawat, and A. Bijalwan, "Process of finding defects in software testing," Intelligent and Computing in Engineering, vol. 10, pp. 297-300, 2017.
- [6] International Standard ISO/IEC/IEEE 29119-2. (2013). Software and systems engineering – Software Testing–Part 2: Test processes. IEEE Xplore.
- [7] International Standard ISO/IEC/IEEE 29119-3. (2013). Software and systems engineering – Software Testing–Part 3: Test documentation. IEEE Xplore.
- [8] Spring MVC. [Online]. Available : <http://www.fusionidea.biz/spring-mvc-framework-part-1-ยินดีต้อนรับสู่-spring-mvc-framework>. เข้าถึงเมื่อวันที่ 1 ธันวาคม 2559
- [9] อุไร ทองหัวไผ่ (2558). การทดสอบซอฟต์แวร์ สืบค้นวันที่ 24 พฤศจิกายน 2561 จาก เว็บไซต์ <https://www.tci-thaijo.org/index.php/jkbu/article/download/47567/39432/>
- [10] D. Bhatt, "A survey of efficient software testing technique and analysis," IREJOURNALS, vol. 1, pp. 1-4, July 2017.
- [11] U. Kumar Tiwari and S. Kumar, "Component level testing in component-based software development," International Journal of Innovations & Advancement in Computer Science: IJIACS, vol. 6, pp. 69-73, January 2017.
- [12] M. Felderer, C. Haisjackl, V. Peker, and R. Brey, "A risk assessment framework for software testing," Springer-Verlag, pp. 294-308, 2014.

- [13] Mart (2560). *Non-Functional Test Cases* สืบค้นวันที่ 24 พฤศจิกายน 2561 จากเว็บไซต์ <https://www.howtoautomate.in.th/how-to-design-non-functional-test-case>.
- [14] Atom (2560). *Software Test* สืบค้นวันที่ 24 พฤศจิกายน 2561 จากเว็บไซต์ <http://iamssoftwaretester.blogspot.com/2014/04/functional-testing-vs-non-functional.html>
- [15] SoftwareTestingStandard.org (2556). *Software Testing Standard* สืบค้นวันที่ 24 พฤศจิกายน 2561 จากเว็บไซต์ <http://www.softwaretestingstandard.org/part2.php>
- [16] IEEE Std 29119, Part 1, Part 2, Part 3, Part 4 and Part 5. Available from <http://ieee.org>.
- [17] Claire Lohr (2561). *IEEE 829-2008 - IEEE Standard for Software and System Test Documentation* สืบค้นวันที่ 24 พฤศจิกายน 2561 จากเว็บไซต์ <https://standards.ieee.org/standard/829-2008.html>
- [18] SOFTWARETESTINGHELP (2562). *Important Software Test Metrics And Measurements – Explained With Examples And Graphs* สืบค้นวันที่ 20 พฤศจิกายน 2562 จากเว็บไซต์ <https://www.softwaretestinghelp.com/software-test-metrics-and-measurements/>
- [19] T. Honglei1, S. Wei1 and Z. Yanan. (2009). The Research on Software Metrics and Software Complexity Metrics. International Forum on Computer Science-Technology and Applications.
- [20] T. Abdou, P. Grogono and P. Kamthan. (2012). A Conceptual Framework for Open Source Software Test Process. IEEE 36th International Conference on Computer Software and Applications Workshops.
- [21] L. LAZIC, “Software quality & testing metrics,” Recent Advances in Information Science, pp. 415-423.
- [22] N. Chen, Ethan W. Chen, and Ian S. Chen, “Integration software testing standard ISO/IEC/IEEE 29119 to agile development,” SERP’18. Research and Practice, pp. 89-95, 2018.
- [23] Gurock (2562). *TestRail* สืบค้นวันที่ 14 ธันวาคม 2562 จากเว็บไซต์ <https://www.gurock.com/testrail>

ภาคผนวก

ภาคผนวก ก

แบบสอบถามความพึงพอใจ

การใช้งานระบบทดสอบซอฟต์แวร์ตามมาตรฐาน ISO 29119

คำชี้แจงแบบสอบถาม

1. เพื่อให้ผู้พัฒนาระบบทราบถึงผลการดำเนินงานของระบบ และเพื่อประโยชน์ในการปรับปรุงระบบให้มีประสิทธิภาพมากขึ้น
2. โปรดเติมเครื่องหมาย ✓ และกรอกข้อความให้สมบูรณ์

ส่วนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

- 1.1 เพศ () ชาย () หญิง
- 1.2 อายุ () 18-22 ปี () 23-27 ปี () 28-32 ปี
- 1.3 ตำแหน่ง () โปรแกรมเมอร์ () ผู้ทดสอบระบบ () ผู้มีส่วนเกี่ยวข้องด้านซอฟต์แวร์ () อื่น ๆ

ส่วนที่ 2 สํารวจความพึงพอใจ

คำชี้แจง ระดับ 5 = มากที่สุด ระดับ 4 = มาก ระดับ 3 = ปานกลาง ระดับ 2 = น้อย ระดับ 1 = น้อยที่สุด

รายละเอียด	ระดับความพึงพอใจ				
	มากที่สุด 5	มาก 4	ปานกลาง 3	น้อย 2	น้อยที่สุด 1
ความพึงพอใจด้านการใช้งาน					
1. ท่านมีความพึงพอใจในการใช้งานระบบโดยภาพรวมเท่าใด					
ความพึงพอใจในการใช้บริการของระบบ					
1. ระบบมีความพร้อมในการใช้งาน					
2. ระบบมีความครบถ้วนสมบูรณ์ของข้อมูล					
3. ข้อมูลภายในระบบมีความถูกต้องสมบูรณ์ และครบถ้วน					
4. ระบบมีความน่าเชื่อถือของข้อมูลในระบบ					

รายละเอียด	ระดับความพึงพอใจ				
	มากที่สุด 5	มาก 4	ปานกลาง 3	น้อย 2	น้อยที่สุด 1
5. ระบบมีความสะดวกในการใช้งาน					
6. ประโยชน์จากการใช้งานระบบ					
7. ระบบสามารถตรวจสอบได้					
ความพึงพอใจด้านความมีประสิทธิภาพ					
1. ระบบใช้งานได้ง่าย และไม่ซับซ้อน					
2. ระบบมีความเร็วในการตอบสนองต่อการใช้งาน					
3. ข้อมูลมีประโยชน์ต่อการใช้งาน					
3. ระบบมีการจัดระดับความปลอดภัย และมีการกำหนดสิทธิในการเข้าถึงข้อมูล					

ส่วนที่ 3 ข้อเสนอแนะเพื่อปรับปรุงระบบ

.....

.....

.....

ภาคผนวก ข



O-ST 001

การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

การพัฒนากระบวนการวางแผนการทดสอบซอฟต์แวร์ตามวงจรการทดสอบซอฟต์แวร์และ
มาตรฐาน ISO 29119, IEEE 829
The Development of Planning Testing System Follow by Software Testing Life Cycle
and ISO 29119, IEEE 829 Standard

ชฎาธาร รักสวัสดิ์¹ (Chadatar Raksawat)¹ ปัทมา เจริญพร (Pattama Charoenporn)²

บทคัดย่อ

ในกระบวนการพัฒนาซอฟต์แวร์ควรมีการทดสอบซอฟต์แวร์ เพื่อตรวจสอบและปรับปรุงให้ซอฟต์แวร์มีความถูกต้อง สมบูรณ์ คุณภาพดี และสามารถระบุ ค้นหาหรือแก้ไขข้อผิดพลาดของซอฟต์แวร์ได้ การทดสอบซอฟต์แวร์ให้มีประสิทธิภาพ ถูกต้อง และเชื่อถือได้ จำเป็นต้องอ้างอิงมาตรฐานในการทดสอบซอฟต์แวร์ โดยศึกษาและทำความเข้าใจในมาตรฐานการ ทดสอบให้ครบถ้วนและละเอียด รอบคอบ ซึ่งเป็นสิ่งที่ยุ่งยาก เสียเวลา และยังเพิ่มต้นทุนในการพัฒนาซอฟต์แวร์ที่มีภาระงาน ด้านการทดสอบซอฟต์แวร์เพิ่ม โดยงานวิจัยนี้จัดทำเพื่อใช้ในกระบวนการทดสอบซอฟต์แวร์ ซึ่งเริ่มพัฒนาระบบโดยหา ความสัมพันธ์ระหว่างวงจรการทดสอบซอฟต์แวร์ และมาตรฐานการทดสอบซอฟต์แวร์ทั้ง ISO 29119 และ IEEE 829 ที่เป็น มาตรฐานด้านเอกสารการทดสอบ เพื่อศึกษาความเกี่ยวข้อง สอดคล้อง ระหว่างวงจรการทดสอบและเอกสารการทดสอบ แล้วนำมาสร้างฟังก์ชันในระบบวางแผนการทดสอบซอฟต์แวร์ให้สามารถปรับใช้ได้กับซอฟต์แวร์ทุกประเภท และมีการวัด ประสิทธิภาพทั้งในด้านข้อกำหนดเชิงบรรทัดฐานและประสิทธิภาพในการทดสอบซอฟต์แวร์ เพื่อควบคุม ตรวจสอบ และปรับปรุง คุณภาพของซอฟต์แวร์ โดยระบบช่วยลดต้นทุนและความยุ่งยากในการทดสอบซอฟต์แวร์ ฟังก์ชันภายในงานวิจัยประกอบด้วย test plan, test procedure specification, test data requirement, test environment requirement, test status, test result, measurement software performance และ documentation ซึ่งเอกสารการทดสอบที่ผู้ใช้งานจะได้รับมีความถูกต้อง ครบคลุม ครบถ้วนทุกขั้นตอนในการทดสอบ และมีเนื้อหาในเอกสารการทดสอบที่เป็นไปตามมาตรฐานสากล

คำสำคัญ การทดสอบซอฟต์แวร์ การพัฒนาซอฟต์แวร์ ประสิทธิภาพของซอฟต์แวร์ มาตรฐานสากลในการทดสอบ ซอฟต์แวร์ เอกสารการทดสอบ

¹นักศึกษาลูกศรวิทยาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง e-mail address : 60605076@kmitl.ac.th

² ดร. ประจำสาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง e-mail address : pattama.ch@kmitl.ac.th



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

Abstract

In software development process there should be software testing to inspection and improve the software to be correctness, completely, good quality and able to identify, finding or fix software errors. Software testing to be efficient, correctness and reliability that are necessary to refer to the software testing standard. By studying and understanding the testing standards to completely and thoroughly which is complicated, time-consuming, and increases the cost of software development with additional software testing workloads. In this research, develops for use in the software testing process. That developed by finding a relationship between software testing life cycle and software testing standards in ISO 29119 and IEEE 829 that are software test documentation standard. To study the related between the testing life cycle and the test documentation and then use to create functions in the software testing planning system that can be used in all types of software and measure the performance of a normative requirement and software testing performance to control, verify and improve software quality. Including the system able to reduce the cost and difficulty of software testing. So, function in the research consists of a test plan, test procedure specification, test data requirement, test environment requirement, test status, test result, measurement software performance and documentation. The user will receive the test documents that are correctness, cover and completeness in every step of the test, and the content of the test document that meets international standards.

Keywords: Software testing, Software development, Software performance, Software testing standard, Software documentation



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

บทนำ

ในปัจจุบันนี้เทคโนโลยีมีการพัฒนาอย่างก้าวหน้าและรวดเร็ว เมื่อโลกในปัจจุบันขับเคลื่อนด้วยเทคโนโลยี (25 ก.ย. 2561) จึงเรียกว่า ยุค Digital 4.0 หรือ “ดิจิทัล 4.0” หรือ “ดิจิทัลไทยแลนด์” เป็นวลีที่คนไทยเริ่มจะได้ยินบ่อยขึ้นในช่วงหลายปีมานี้ (สำนักงานปลัดกระทรวงศึกษาธิการ, 2561) ซึ่งเทคโนโลยีเป็นที่สนใจของคนทุกมุมโลก ทุกเพศ ทุกวัย ทุกสาขาอาชีพ อย่างแพร่หลายและมีการนำมาใช้ในการทำงาน ชีวิตประจำวัน และการเรียนการศึกษา ส่งผลให้ในปัจจุบันนี้มีหลักสูตรที่เกี่ยวกับเทคโนโลยีเพิ่มเข้าไปด้วย เพื่อให้เด็ก ๆ รู้เท่าทันถึงภัยที่มาพร้อมกับโลกยุคดิจิทัล ซึ่งเทคโนโลยีและโลกดิจิทัลเปลี่ยนแปลงอย่างรวดเร็ว และเคลื่อนที่ไม่มีหยุด องค์กรธุรกิจต่าง ๆ จึงจำเป็นต้องปรับตัวให้ทันตามเทรนด์ พัฒนานวัตกรรมเพื่อต่อยอดธุรกิจบนการแข่งขันที่รวดเร็วและรอบด้านจาก SME (Small and Medium Enterprises) ที่แปลว่า วิชากิจขนาดกลางและขนาดย่อมให้กลายเป็น Smart Enterprise ที่มีศักยภาพสูงขึ้น จากบริการธรรมดาให้กลายเป็น High Value Service (P. Eira, R. J. Machado and A. Silva, 2018) เพื่อความมั่นคง มั่งคั่ง และยั่งยืนของธุรกิจ (สำนักงานปลัดกระทรวงศึกษาธิการ, 2561) โดยหลายธุรกิจมีการนำซอฟต์แวร์มาปรับใช้ในการทำงานมากขึ้นเพื่อเพิ่มประสิทธิภาพการทำงาน ลดต้นทุน ลดระยะเวลา ควบคุมคุณภาพของงาน เพิ่มโอกาสทางการตลาดและโอกาสในการทำงาน ส่งผลให้ในปัจจุบันนี้มีซอฟต์แวร์ออกสู่ท้องตลาดเพิ่มขึ้นมากในรูปแบบที่หลากหลายเพื่อให้ตอบสนองต่อกลุ่มธุรกิจที่มีหลายประเภท โดยส่วนใหญ่ที่ออกสู่ท้องตลาดจะอยู่ในรูปแบบซอฟต์แวร์ประยุกต์ที่ประกอบด้วยซอฟต์แวร์สำเร็จและซอฟต์แวร์ใช้งานเฉพาะ ซึ่งซอฟต์แวร์สำเร็จ (software package) เป็นซอฟต์แวร์ที่มีความนิยมใช้กันสูงมาก ดังนั้นบริษัทที่จะผลิตซอฟต์แวร์ขายจะต้องมีคุณสมบัติที่เป็นจุดเด่นเฉพาะตัว เพื่อดึงดูดลูกค้าและต้องสร้างความน่าเชื่อถือให้แก่ลูกค้า สร้างความมั่นใจให้แก่ลูกค้า เมื่อใช้ซอฟต์แวร์ของทางบริษัทว่าจะไม่ส่งผลให้ธุรกิจของลูกค้าเกิดความเสียหายเป็นอันขาด ดังนั้นการทดสอบซอฟต์แวร์ก่อนที่จะทำการส่งมอบให้ลูกค้าถือเป็นสิ่งที่สำคัญ หากขาดการทดสอบซอฟต์แวร์ก่อนส่งมอบให้ลูกค้าทำให้เกิด ค่าใช้จ่ายในการแก้ไขซอฟต์แวร์สูง ซอฟต์แวร์มีคุณภาพต่ำ ส่งผลต่อซอฟต์แวร์ทั้งระบบทำให้เกิด failure ระบบไม่สามารถใช้งานได้ ส่งผลให้ลูกค้าขาดความเชื่อมั่นในซอฟต์แวร์ของบริษัท

การทดสอบซอฟต์แวร์จะต้องทำการศึกษาค้นคว้าหาข้อมูลก่อนที่จะทดสอบซอฟต์แวร์ทั้งขั้นตอนในการทดสอบซอฟต์แวร์ ความเหมาะสมที่จะนำมาปรับใช้กับซอฟต์แวร์ที่ได้พัฒนาขึ้น หรือแผนการทดสอบซอฟต์แวร์ที่บริษัทมีอยู่แล้วด้วยซอฟต์แวร์มีหลายประเภทและหลายขนาด อีกทั้งยังมีคุณลักษณะความต้องการเฉพาะตัว การที่จะหาข้อสรุปมาว่า การทดสอบที่ได้ทำอยู่นั้นดีหรือไม่นั้นทำได้ยาก การนำมามาตรฐานการทดสอบมาใช้ในการทดสอบซอฟต์แวร์เป็นสิ่งที่ยุ่งยากเสียเวลา แต่เพื่อป้องกันข้อผิดพลาดที่จะเกิดขึ้นในกระบวนการทดสอบ การทดสอบที่ครอบคลุมครบถ้วน สามารถนำไปใช้งานได้จริง (E. Veenendaal, 2016) อีกทั้งยังสามารถสร้างความน่าเชื่อถือให้แก่ผู้ใช้งานระบบ ความสำคัญของการทดสอบตามมาตรฐานสากลจึงเป็นสิ่งที่ควรทำ (The Home of International Standards for Software Testing, 2018)

ทางผู้วิจัยเห็นถึงความจำเป็นพร้อมทั้งหาแนวทางในการแก้ไขปัญหาข้างต้น จึงได้พัฒนาระบบที่ช่วยในการทดสอบซอฟต์แวร์ โดยเริ่มจากการหาความสัมพันธ์ระหว่างวงจรการทดสอบและมาตรฐานในการทดสอบ ทางผู้วิจัยได้อ้างอิงมาตรฐาน ISO 29119 และมาตรฐาน IEEE 829 ที่เป็นมาตรฐานสากลในการทดสอบซอฟต์แวร์ (A. Rana, A. Singh Rawat and A. Bijalwan, 2017) ซึ่งเป็นที่แน่นอนว่ามาตรฐานที่ได้ประกาศใช้นั้นมีการตรวจสอบความถูกต้องทั้งกระบวนการทำงานและความเหมาะสมที่สามารถนำมาปรับใช้กับงานในปัจจุบันนี้ (ISO/IEC/IEEE 29119-2, 2013), (ISO/IEC/IEEE 29119-3, 2013) การที่นำมามาตรฐานการทดสอบ ISO 29119 และ IEEE 829 หาความสัมพันธ์เพื่อต้องการให้ระบบที่สร้างขึ้นมา



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

ครอบคลุมทุกขั้นตอนในการดำเนินงาน ครอบคลุมเอกสารที่ในการทดสอบ อีกทั้งนำไปจับคู่กับวงจรการพัฒนาซอฟต์แวร์ที่เป็นขั้นตอนพื้นฐานในการทดสอบเพื่อให้สามารถนำมาปรับใช้ในซอฟต์แวร์หลายประเภท ส่งผลให้ระบบสามารถสร้างเอกสารการทดสอบที่ครอบคลุมทั้งวงจรการทดสอบซอฟต์แวร์ที่เป็นไปตามมาตรฐานสากลและสามารถนำไปใช้งานได้จริง ซึ่งภายในระบบนี้ได้มีการวัดประสิทธิภาพของซอฟต์แวร์ที่ทำการทดสอบ โดยมีการวัดประสิทธิภาพ 2 ประเภท คือด้านข้อกำหนดเชิงบรรทัดฐานและด้านความสามารถในการทดสอบซอฟต์แวร์ เพื่อให้ทราบถึงคุณภาพ ประสิทธิภาพ และความสมบูรณ์ของซอฟต์แวร์ก่อนที่จะส่งมอบให้ลูกค้าหรือนำไปใช้งานจริง

วัตถุประสงค์

1. เพื่อตรวจสอบ และปรับปรุงให้ซอฟต์แวร์มีความถูกต้อง สมบูรณ์ คุณภาพที่ดี
2. เพื่อวัดประสิทธิภาพของซอฟต์แวร์ที่นำมาทดสอบ
3. เพื่อสามารถสร้างและนำเอกสารการทดสอบซอฟต์แวร์ไปใช้งาน โดยอ้างอิงตามมาตรฐานการทดสอบซอฟต์แวร์

ระเบียบวิธีวิจัย

ระบบวางแผนการทดสอบซอฟต์แวร์ ผู้วิจัยได้ทำการหาความสัมพันธ์ระหว่างวงจรการทดสอบซอฟต์แวร์กับมาตรฐานการทดสอบซอฟต์แวร์ทั้ง ISO 29119 และ IEEE 829 เพื่อสร้างระบบและฟังก์ชันการทำงานให้ครอบคลุมทุกขั้นตอนของการทดสอบและให้ได้ออกสารประกอบการทดสอบซอฟต์แวร์ที่ครบถ้วนสมบูรณ์ตามขั้นตอนการทดสอบและมาตรฐานในการทดสอบซอฟต์แวร์ (N. Chen, Ethan W. Chen and Ian S. Chen, 2018) ซึ่งความสัมพันธ์ระหว่างขั้นตอนในการทดสอบซอฟต์แวร์และมาตรฐานในการทดสอบซอฟต์แวร์แสดงดังภาพที่ 1 โดยผู้วิจัยได้เริ่มจากศึกษาวงจรในการทดสอบซอฟต์แวร์ 6 ขั้นตอน คือ (1) กำหนดความต้องการในการทดสอบ (2) วางแผนในการทดสอบ (3) กำหนดกรณีทดสอบต่าง ๆ ที่ต้องการทดสอบ (4) ตั้งค่าสภาพแวดล้อมที่ต้องการทดสอบ (5) ทดสอบการทำงานของซอฟต์แวร์ และ (6) รายงานผลการทดสอบซอฟต์แวร์ (U. Kumar Tiwari and S. Kumar, 2017) จากนั้นศึกษามาตรฐานในการทดสอบซอฟต์แวร์และหาความสัมพันธ์กับขั้นตอนการทดสอบซอฟต์แวร์ที่มีอยู่โดยทั่วไป เพื่อให้ได้กระบวนการในการทดสอบซอฟต์แวร์เป็นไปตามมาตรฐานสากล ครอบคลุมทุกขั้นตอนของการทดสอบและสามารถนำไปใช้งานได้จริง ซึ่งในงานวิจัยนี้อ้างอิงมาตรฐานสากลในการทดสอบซอฟต์แวร์ คือ ISO 29119 ประกอบด้วย 5 ส่วนดังนี้ (1) แนวความคิดและการนิยาม (2) กระบวนการทดสอบ (3) เอกสารการทดสอบ (4) เทคนิคในการทดสอบ และ (5) การทดสอบด้วยค่าหลัก (N. Chen, Ethan W. Chen and Ian S. Chen, 2018) และ มาตรฐาน IEEE 829 ประกอบด้วย 9 ส่วน ดังนี้ (1) master test plan (2) level test plan (3) level test design (4) level test case (5) level test procedure (6) level test log (7) anomaly report และ (8) test report โดยที่ในงานวิจัยจะเริ่มจาก ISO 29119-2 คือ ส่วนกระบวนการทดสอบ เป็นส่วนหลักของมาตรฐานที่ใช้ในการกำหนดขั้นตอนของการทดสอบซอฟต์แวร์ แบ่งออกเป็น 3 ส่วน คือ (1) กระบวนการขององค์กร (2) กระบวนการจัดการทดสอบ และ (3) กระบวนการทดสอบแบบไดนามิก (ISO/IEC/IEEE 29119-2, 2013) และต่อมาศึกษามาตรฐาน ISO 29119-3 และ IEEE 829 ที่เป็นส่วนมาตรฐานด้านเอกสารการทดสอบซอฟต์แวร์มีการทำงานซึ่งแบ่งออกเป็น 3 ส่วน เช่นเดียวกับกับมาตรฐาน ISO 29119-2 (ISO/IEC/IEEE 29119-3, 2013) เอกสารการทดสอบประกอบด้วย (1) แผนการทดสอบ (2) ข้อกำหนดการทดสอบ (3) กำหนดความต้องการข้อมูลในการทดสอบ (4) กำหนดความต้องการสภาพแวดล้อมในการทดสอบ (5) รายงานสถานะการทดสอบ (6) เปรียบเทียบพร้อมวัดประสิทธิภาพการทดสอบ (7) รายงาน



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

เหตุการณ์ และ (8) เอกสารประกอบการทดสอบทั้งหมดที่พร้อมในการพิมพ์ ซึ่งโปรแกรมที่ใช้ในการพัฒนาระบบวางแผนการทดสอบซอฟต์แวร์ คือ โปรแกรม visual studio พัฒนาโดยภาษา java ในรูปแบบ windows application

จากการหาความสัมพันธ์ระหว่างมาตรฐาน ISO 29119-2, ISO 29119-3, IEEE 829 และ วงจรการทดสอบในข้างต้น สามารถสร้างฟังก์ชันสำหรับการเตรียมเอกสารการทดสอบให้ครอบคลุมทั้งหมด ซึ่งในระบบประกอบด้วยฟังก์ชัน test plan, test procedure specification, test data requirement, test environment requirement, test status, test result, measurement software performance and documentation โดยสามารถแบ่งความสัมพันธ์ออกได้เป็น 4 ส่วน ดังนี้

ส่วนที่ 1 : test management documentation ในส่วนนี้เกี่ยวข้องกับการสร้างแผนการทดสอบซอฟต์แวร์ คือ ฟังก์ชัน test plan

ส่วนที่ 2 : dynamic test documentation เกี่ยวข้องกับกระบวนการจัดการทดสอบซอฟต์แวร์ ที่ประกอบด้วย 5 ฟังก์ชันดังนี้ (1) ฟังก์ชัน test procedure specification คือ ข้อกำหนดการทดสอบ (2) ฟังก์ชัน test data requirement คือ กำหนดความต้องการของข้อมูลในการทดสอบ (3) ฟังก์ชัน test environment requirement คือ กำหนดความต้องการของสภาพแวดล้อมที่ใช้ในการทดสอบ (4) ฟังก์ชัน test status คือ รายงานสถานการณ์ทดสอบ (5) ฟังก์ชัน test result คือ เปรียบเทียบผลการทดสอบ

ส่วนที่ 3 : measurement เป็นขั้นตอนในการวัดประสิทธิภาพซอฟต์แวร์ ในระบบแบ่งการวัดประสิทธิภาพออกเป็น 2 ด้าน คือ ข้อกำหนดเชิงบรรทัดฐาน และการวัดประสิทธิภาพในการทดสอบซอฟต์แวร์

3.1 การวัดประสิทธิภาพในด้านข้อกำหนดเชิงบรรทัดฐาน จะทำการศึกษาความสัมพันธ์ระหว่างข้อกำหนดเชิงบรรทัดฐานใน ISO 29119-2 กับรายการข้อมูลของ ISO 29119-3 แบ่งส่วนในการวัดประสิทธิภาพออกเป็น 3 ส่วน คือ (1) กระบวนการขององค์กร (2) กระบวนการจัดการทดสอบ (3) กระบวนการทดสอบแบบไดนามิก ซึ่งการวัดประสิทธิภาพข้อกำหนดเชิงบรรทัดฐานจะดำเนินในฟังก์ชัน test status ไปยัง Measurement โดยคลิกปุ่ม Normative requirement โดยมีค่าระดับความจำเป็นของข้อมูล 3 ระดับ คือ may, should, shall งานวิจัยได้กำหนดค่าดังนี้

may มีค่าความจำเป็นของข้อมูลอยู่ที่ร้อยละ 40

should มีค่าความจำเป็นของข้อมูลอยู่ที่ร้อยละ 60

shall มีค่าความจำเป็นของข้อมูลอยู่ที่ร้อยละ 80

3.2 การวัดประสิทธิภาพในการทดสอบซอฟต์แวร์ จะทำการวัดทั้งหมด 5 ข้อ คือ (1) จำนวนที่ทดสอบตามกรณีทดสอบที่กำหนดไว้ (number of executed test cases) (2) จำนวนที่นำกรณีทดสอบมาทดสอบซ้ำ (number of re-executed test cases) (3) จำนวนเหตุการณ์ที่เกิดขึ้น (number of incidents per category) (4) จำนวนปัญหาที่ได้รับการแก้ไขอย่างสมบูรณ์ (number of solved incidents per category) และ (5) จำนวนเวลา (ชั่วโมง) ที่ใช้ในการทดสอบ (number of hours spent) โดยการวัดประสิทธิภาพจะดำเนินการในฟังก์ชัน test result โดยคลิกปุ่ม Select Metric ซึ่งสูตรที่ใช้ในการคำนวณประสิทธิภาพแสดงดังนี้

จำนวนที่ทดสอบตามกรณีทดสอบที่กำหนดไว้ = (จำนวนกรณีทดสอบที่ถูกทดสอบ/จำนวนกรณีทดสอบทั้งหมด)

จำนวนที่นำกรณีทดสอบมาทดสอบซ้ำ = (จำนวนกรณีทดสอบที่มีจำนวนครั้งในการทดสอบมากกว่า 1 ครั้ง/จำนวนกรณีทดสอบทั้งหมด)

จำนวนเหตุการณ์ที่เกิดขึ้น = (จำนวนเหตุการณ์ที่พบในกรณีทดสอบ/จำนวนกรณีทดสอบที่ถูกทดสอบ)



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

จำนวนปัญหาที่ได้รับการแก้ไขอย่างสมบูรณ์ = (จำนวนกรณีทดสอบที่แก้ไข/จำนวนกรณีทดสอบที่พบปัญหา)

จำนวนเวลา (ชั่วโมง) ที่ใช้ในการทดสอบ = (เวลาที่ใช้ในการทดสอบ/จำนวนกรณีทดสอบที่ถูกทดสอบ)

ส่วนที่ 4 : documentation เป็นขั้นตอนสุดท้ายของระบบที่เกี่ยวข้องเอกสารการทดสอบซอฟต์แวร์ ซึ่งประกอบด้วยเอกสารที่รายงานเหตุการณ์ที่เกิดขึ้นระหว่างการทดสอบ คือ ฟังก์ชัน test incident report และฟังก์ชัน documentation ที่ใช้ในการพิมพ์เอกสารการทดสอบ



ภาพที่ 1 แผนภาพความสัมพันธ์ระหว่างวงจรการทดสอบซอฟต์แวร์ มาตรฐานการทดสอบ และฟังก์ชันการทำงานในระบบ

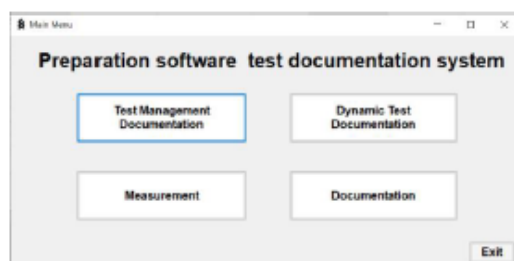
ผลการวิจัย

ระบบวางแผนการทดสอบซอฟต์แวร์สามารถตรวจสอบ และปรับปรุงให้ซอฟต์แวร์มีความถูกต้อง สมบูรณ์ คุณภาพดี สามารถวัดประสิทธิภาพของซอฟต์แวร์ได้ โดยแบ่งออกเป็น 2 ส่วน โดย ส่วนแรกวัดประสิทธิภาพในด้านข้อกำหนดเชิงบรรทัดฐาน โดยดำเนินการในรูปของฟังก์ชัน Normative Requirement ซึ่งเป็นส่วนที่บอกระดับความจำเป็นของข้อมูลซึ่งจะแจ้งเตือนในทุกฟังก์ชัน เพื่อเป็นการเตือนผู้ใช้งานให้ทราบถึงระดับความจำเป็นของข้อมูล ส่วนที่สองวัดประสิทธิภาพในการทดสอบซอฟต์แวร์ โดยดำเนินการในรูปของฟังก์ชัน Measurement เพื่อเป็นการบอกถึงคุณภาพและประสิทธิภาพของซอฟต์แวร์ ซึ่งจะแสดงผลให้ผู้ใช้งานทราบถึงส่วนที่ต้องปรับปรุงแก้ไข และตรวจสอบความถูกต้องสมบูรณ์ และสามารถสร้างและนำเอกสารการทดสอบไปใช้ได้จริง โดยแสดงในรูปฟังก์ชัน documentation โดยจากระเบียบวิธีวิจัยแบ่ง

ระบบออกเป็น 4 ส่วน คือ test management documentation, dynamic test documentation, measurement และ documentation แสดงดังภาพที่ 2 ซึ่งเป็นหน้าแรกของระบบวางแผนการทดสอบซอฟต์แวร์ ซึ่งภายในระบบวางแผนการทดสอบซอฟต์แวร์ประกอบด้วยฟังก์ชันดังนี้

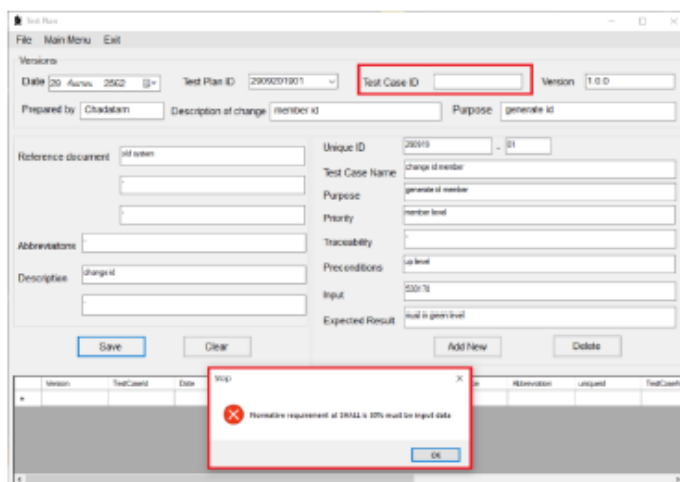


การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference



ภาพที่ 2 หน้าหลักของระบบวางแผนการทดสอบซอฟต์แวร์

1. ฟังก์ชัน test plan เป็นฟังก์ชันในการสร้างแผนการทดสอบ แสดงดังภาพที่ 3 ผู้ใช้งานจำเป็นต้องกรอกข้อมูลให้ครบถ้วน เนื่องจากทุกช่องต้องมีการเก็บค่าประสิทธิภาพของข้อกำหนดเชิงบรรทัดฐานเพื่อนำไปคำนวณค่าประสิทธิภาพ หากผู้ใช้งานไม่กรอกข้อมูลในช่องนั้นระบบทำการแจ้งเตือน ดังภาพที่ 3 เพื่อเป็นการควบคุมคุณภาพ และตรวจสอบความถูกต้อง ความสมบูรณ์ของซอฟต์แวร์ เมื่อผู้ใช้งานทำการทดสอบระบบเรียบร้อยแล้วคลิก “Save” เพื่อบันทึกข้อมูลลงตารางด้านล่าง จากนั้นคลิกไปที่ช่องเมนูบาร์เพื่อไปยังฟังก์ชัน Test Procedure Specification ต่อไป



ภาพที่ 3 ฟังก์ชันในการสร้างแผนการทดสอบ

2. ฟังก์ชัน test procedure specification เป็นฟังก์ชันในการกำหนดขั้นตอนการทดสอบ แสดงดังภาพที่ 4 ผู้ใช้งานจำเป็นต้องเลือก Test Case ID ขึ้นมาเพื่อทำการรายงานผลที่เกิดขึ้นหลังจากการทดสอบซอฟต์แวร์ โดยในส่วนของ 1 ผู้ใช้งานทำการกรอกข้อมูลรายละเอียดการทดสอบที่ประกอบด้วย วัตถุประสงค์ ลำดับความสำคัญ และเนื้อหาของการทดสอบ ส่วนที่ 2 ผู้ใช้งานกรอกข้อมูลเกี่ยวกับขั้นตอนในการทดสอบที่ประกอบด้วย จุดเริ่มต้น กรณีทดสอบที่ใช้ ความสัมพันธ์ในแต่ละขั้นตอน จุดหยุดหรือเปลี่ยนการทดสอบ เมื่อผู้ใช้งานทำการกรอกข้อมูลเรียบร้อยแล้วคลิก “Save” เพื่อบันทึกข้อมูลลงตารางด้านล่าง จากนั้นคลิกไปที่ช่องเมนูบาร์เพื่อไปยังฟังก์ชัน test data requirement ต่อไป



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

3. ฟังก์ชัน **test data requirement** เป็นฟังก์ชันในการกำหนดความต้องการของข้อมูลที่จะนำมาใช้ในการทดสอบ แสดงดังภาพที่ 5 ประกอบด้วย ให้คำจำกัดความข้อมูล กำหนดสถานะให้ข้อมูล กำหนดหน้าที่ความรับผิดชอบ ระยะเวลาที่ต้องการ การตั้งค่าข้อมูล เมื่อผู้ใช้งานทำการกรอกข้อมูลเรียบร้อยแล้วคลิก “Save” เพื่อบันทึกข้อมูลลงตารางด้านล่าง จากนั้นคลิกไปที่ช่องเมนูบาร์เพื่อไปยังฟังก์ชัน Environment requirement ต่อไป

4. ฟังก์ชัน **test environment requirement** เป็นฟังก์ชันในการกำหนดสภาพแวดล้อมที่ใช้ในการทดสอบ แสดงดังภาพที่ 6 ซึ่งประกอบด้วย การระบุความต้องการ สถานะของสภาพแวดล้อม ช่วงเวลาที่ต้องการใช้งาน ส่วนแสดงความคิดเห็น เมื่อผู้ใช้งานทำการกรอกข้อมูลเรียบร้อยแล้วคลิก “Save” เพื่อบันทึกข้อมูลลงตารางด้านล่าง จากนั้นคลิกไปที่ช่องเมนูบาร์เพื่อไปยังฟังก์ชัน Test result ต่อไป

ภาพที่ 4 ข้อกำหนดของการทดสอบ

ภาพที่ 5 กำหนดความต้องการของข้อมูลในการทดสอบ



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

ภาพที่ 6 ข้อกำหนดความต้องการสภาพแวดล้อมในการทดสอบ

5. ฟังก์ชัน test result เป็นฟังก์ชันที่ใช้ในการกำหนดผลลัพธ์ที่ต้องการ พร้อมทั้งเลือกเมตริกที่จะนำไปใช้ในการวัดประสิทธิภาพของซอฟต์แวร์ แสดงดังภาพที่ 7 ซึ่งในฟังก์ชันนี้ประกอบด้วย กำหนดหมายเลข ระบุชื่อกรณีทดสอบ กิจกรรมที่ทดสอบ ผลลัพธ์ที่คาดหวัง ผลลัพธ์ที่เกิดขึ้นจริง เลือกเมตริกโดยคลิกปุ่ม “Select Metric” เมื่อคลิกแล้วแสดงดังภาพที่ 8 ผู้ใช้งานเลือกเมตริกที่ต้องการนำมาทดสอบจากนั้นทำการบันทึกจะกลับมาสู่หน้าฟังก์ชันเดิม และเลือกสถานะของการทดสอบที่ทดสอบเสร็จแล้วโดยคลิกปุ่ม “Add status” ระบบจะไปยังฟังก์ชัน test status เป็นฟังก์ชันที่ใช้ในการรายงานสถานะกรณีทดสอบ (test case) ที่ได้ทำการทดสอบเสร็จแล้ว แสดงดังภาพที่ 9 ซึ่งประกอบด้วย ระบุวันที่ที่กำหนดสถานะการทดสอบ ช่วงเวลาความคืบหน้าแผนการทดสอบ

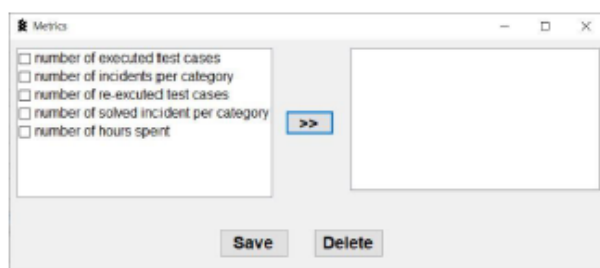
ภาพที่ 7 ข้อมูลผลการทดสอบ

เมื่อกรอกรายละเอียดสมบูรณ์แล้ว ต่อไปทำการคำนวณประสิทธิภาพของซอฟต์แวร์โดยคลิกปุ่ม “Measurement” เพื่อที่จะคำนวณประสิทธิภาพในด้านข้อกำหนดเชิงบรรทัดฐาน แสดงดังภาพที่ 10 แสดงระบบสามารถแสดงข้อมูลเป็นกราฟได้โดยคลิกปุ่ม

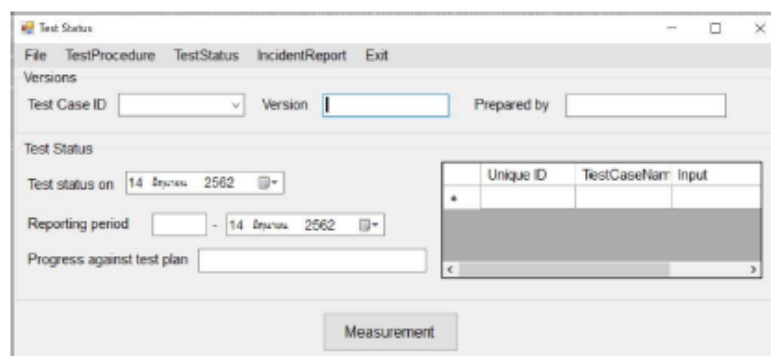


การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

“Show graphs” แสดงดังภาพที่ 11 และเมื่อต้องการคำนวณประสิทธิภาพในการทดสอบซอฟต์แวร์ ไปที่แถบเมนูบาร์คลิกปุ่ม “Normative requirement” เพื่อคำนวณประสิทธิภาพแสดงดังภาพที่ 12 เมื่อผู้ใช้งานทำการกรอกข้อมูลเรียบร้อยแล้วคลิก “Save” เพื่อบันทึกข้อมูลลงตารางด้านล่าง หากการทดสอบมีสถานะไม่ผ่านจำเป็นต้องไปยังช่องเมนูบาร์ที่ฟังก์ชัน Incident report เพื่อสร้างรายงานข้อผิดพลาดที่เกิดขึ้น



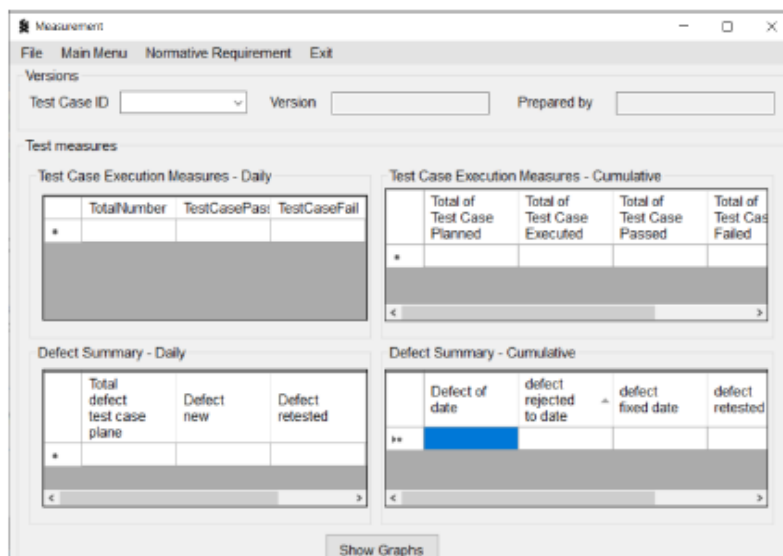
ภาพที่ 8 เมตริกซ์ที่ใช้ในการวัดประสิทธิภาพ



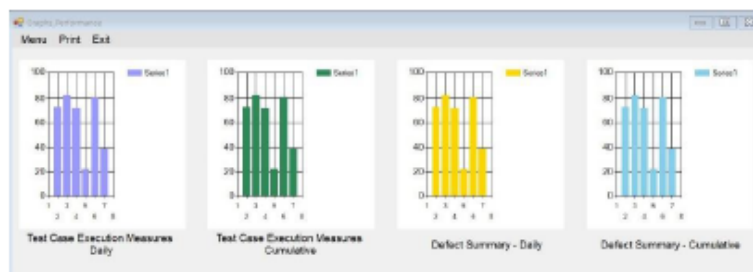
ภาพที่ 9 สถานการณ์ทดสอบ



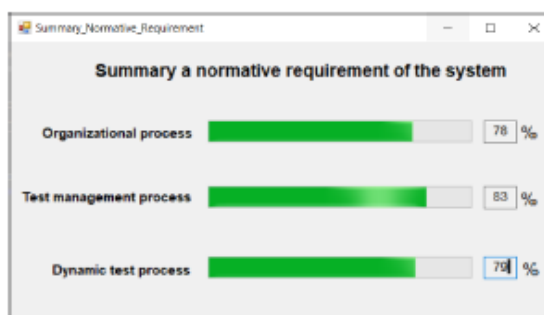
การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference



ภาพที่ 10 วัดประสิทธิผลการทดสอบ



ภาพที่ 11 แสดงผลประสิทธิผลการทดสอบ



ภาพที่ 12 ประสิทธิภาพในด้านข้อกำหนดเชิงบรรทัดฐาน



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

เมื่อผู้ใช้งานทำการกรอกข้อมูลเรียบร้อยแล้วคลิก “Save” เพื่อบันทึกข้อมูลลงตารางด้านล่าง หากการทดสอบมีสถานะไม่ผ่าน จำเป็นต้องไปยังช่องเมนูบาร์ที่ฟังก์ชัน Incident report เพื่อสร้างรายงานข้อผิดพลาดที่เกิดขึ้น

6. ฟังก์ชัน incident report เป็นฟังก์ชันในการรายงานเหตุการณ์ที่เกิดขึ้นระหว่างการทดสอบซอฟต์แวร์ แสดงดังภาพที่ 13 ประกอบด้วย หมายเลขรายงาน หัวข้อรายงาน ชนิดของซอฟต์แวร์ สถานะของการทดสอบ เวอร์ชันที่ทดสอบ ชื่อผู้สร้างรายงาน ชื่อผู้รายงานข้อผิดพลาด พร้อมระยะเวลารายงาน คำอธิบายรายงาน (comprehensive description) สถานการณ์รายงาน (observed during) ข้อผิดพลาดเกิดขึ้นในขั้นตอนไหน (observed in) ข้อผิดพลาดมีอาการแบบไหน (symptom) ผลกระทบต่อผู้ใช้งานในระดับไหน (user impact) ผู้ใช้งานมีความต้องการใช้งานในระดับไหน (user urgency)

ภาพที่ 13 รายงานข้อผิดพลาด

7. ฟังก์ชัน documentation เป็นฟังก์ชันในการพิมพ์เอกสารการทดสอบซอฟต์แวร์ออกมาใช้งาน ผู้ใช้งานสามารถสร้างและนำเอกสารการทดสอบซอฟต์แวร์ไปใช้ได้จากฟังก์ชันนี้ ซึ่งพิมพ์ออกมาได้ในรูปแบบสกุลไฟล์ pdf โดยสามารถคลิกปุ่ม Documentation ในภาพที่ 2 แล้วทำการเลือกเอกสารที่ต้องการ โดยภาพที่ 14 เป็นตัวอย่างเอกสารแผนการทดสอบที่สมบูรณ์ แสดงอยู่ในไฟล์สกุล pdf



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

Test Plan Report

7/6/2562

Organization Name : KMITL	Project Name : ทดสอบ UI	Test Plan Id : 1		
Project Manager Name : -	Project Id : 1	Test Case Id : 1		
Project Executor Name : ๑๑1๑๑๑	Created By :-	Created Date : 2562-07-06		
Test Project Detail				
TestPlan Name : ทดสอบ UI	Scope : ทดสอบเว็บไซต์ระบบขายได้			
Module Name : ผลิตภัณฑ์	Module Test : ผลิตภัณฑ์			
Test Cost Planning : ๘ 0.00	TestCost Actual : ๘ 0.00			
Overview of System : ทดสอบ UI	Standard : ISO 29119			
Context Testing : ทดสอบการใช้งานในส่วน UI	Test Delivery : ระบบสามารถทำงานได้บนมือถือและคอมพิวเตอร์			
Risk Contingency : การเข้าถึง มีผลต่อตัวงานในการใช้งาน	Object Test : ทดสอบการใช้งานมือถือและคอมพิวเตอร์			
Criteria Pass : ระบบทดสอบมือถือและคอมพิวเตอร์	Criteria Fail : ระบบทดสอบมือถือและคอมพิวเตอร์			
Requirement : ทดสอบเว็บไซต์ใช้งานและส่วนที่แสดงผลต่างๆ	Requirement : ทดสอบเว็บไซต์การใช้งาน			
Requirement :	Requirement :			
Reference : -	Glossary : -			
Staff Name	Staff Role	Staff Activities	Staff Responsibility	Staff Hiring
๑๑๑๑๑๑	ทดสอบ	ทดสอบ	ทดสอบ	๘ 0,000.00
Test Activities		Date Activities		
ทดสอบการทำงานระบบการลงทะเบียน		2562-07-06		

ภาพที่ 14 ตัวอย่างเอกสารประกอบการทดสอบ

อภิปรายผลการวิจัย

ระบบวางแผนการทดสอบซอฟต์แวร์อ้างอิงตามมาตรฐานการทดสอบซอฟต์แวร์ และวงจรการทดสอบซอฟต์แวร์สามารถนำไปใช้ในส่วนการทดสอบของซอฟต์แวร์ โดยใช้ได้กับซอฟต์แวร์หลายประเภท ซึ่งช่วยลดระยะเวลาในการศึกษาข้อมูล ขั้นตอนและกระบวนการในการทดสอบซอฟต์แวร์ พร้อมทั้งยังลดต้นทุนในด้านการทดสอบซอฟต์แวร์ โดยระบบภายในงานวิจัยนี้มีการรวบรวมข้อมูล และรายละเอียดทั้งหมดของมาตรฐานการทดสอบซอฟต์แวร์ตามมาตรฐานสากลไว้เรียบร้อยแล้ว ซึ่งระบบสามารถตรวจสอบ และชี้แนวทางในการปรับปรุงซอฟต์แวร์ให้มีความถูกต้อง สมบูรณ์ มีคุณภาพ สามารถบอกระดับความจำเป็นของข้อมูล ทำให้การทดสอบซอฟต์แวร์สามารถทำได้ง่าย อีกทั้งสามารถวัดประสิทธิภาพในการทดสอบซอฟต์แวร์ เพื่อให้ทราบถึงคุณภาพของซอฟต์แวร์ และส่วนที่ต้องปรับปรุงก่อนส่งมอบหรือนำไปใช้งานซึ่งป้องกันการเกิด failure ของระบบได้อย่างดี พร้อมทั้งสามารถสร้างเอกสารทดสอบซอฟต์แวร์ที่ครอบคลุมทุกขั้นตอนการทดสอบและถูกต้องตามมาตรฐานในการทดสอบซอฟต์แวร์ ผู้ใช้งานสามารถนำไปใช้งานได้ถูกต้อง ครบถ้วน



การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9
The 9th STOU National Research Conference

ข้อเสนอแนะ

สามารถนำงานวิจัยนี้ไปปรับใช้ในกระบวนการทดสอบซอฟต์แวร์ได้ในทุกประเภทของซอฟต์แวร์ ซึ่งนำไปใช้ในส่วนการทดสอบ และเอกสารการทดสอบซอฟต์แวร์ เพื่อให้ซอฟต์แวร์ที่ทดสอบนั้นมีความถูกต้อง ครบถ้วน ตามหลักมาตรฐานการทดสอบซอฟต์แวร์สากล ซึ่งสามารถใช้งานได้ง่ายขึ้น ครอบคลุมเนื้อหามาตรฐานการทดสอบซอฟต์แวร์ทั้งหมด

เอกสารอ้างอิง

- The Home of International Standards for Software Testing. (2018). IEEE Std 29119, Part 1, Part 2, Part 3, Part 4, Part 5. สืบค้นเมื่อ 6 เมษายน 2562, จาก <https://www.softwaretestingstandard.org/index.php>.
- สำนักงานปลัดกระทรวงศึกษาธิการ. (2561). ยุค Digital 4.0 เมื่อโลกขับเคลื่อนด้วยเทคโนโลยี. สืบค้นเมื่อ 29 สิงหาคม 2562, จาก <http://www.ops.moe.go.th/ops2017/สาระนั้นารู/2876-ยุค-digital-4-0-เมื่อโลกขับเคลื่อนด้วยเทคโนโลยี-25-ก-ย-2561>.
- P. Eira, R. J. Machado and A. Silva. (2018). Tailoring ISO/IEC/IEEE 29119-3 Standard for Small and Medium-sized Enterprises. *IEEE Trans, On computer society*, 380-389.
- N. Chen, Ethan W. Chen and Ian S. Chen. (2018). Integration Software Testing Standard ISO/IEC/IEEE 29119 to Agile Development. *SERP'18*, 89-95.
- A. Rana, A. Singh Rawat and A. Bijalwan. (2017). Process of finding defects in Software testing. *Intelligent and Computing in Engineering*. 10, 297-300.
- U. Kumar Tiwari, S. Kumar. (2017). Component Level Testing in Component-Based Software Development. *International Journal of Innovations & Advancement in Computer Science*. 6, 69-73.
- E. Veenendaal. (2016). TMMi and ISO/IEC 29119: Friends or Foes?. *TMMi Foundation*, 1-12.
- International Standard ISO/IEC/IEEE 29119-3. (2013). Software and systems engineering – Software Testing- Part 3: Test documentation. *IEEE Xplore*.
- International Standard ISO/IEC/IEEE 29119-2. (2013). Software and systems engineering – Software Testing- Part 2: Test processes. *IEEE Xplore*.
- จักรินทร์ สันติรัตนกัทดี และจิตติมนต์ อังสกุล. (2012). การประเมินความสามารถในการใช้งานซอฟต์แวร์วางแผนมื้ออาหารเพื่อการควบคุมน้ำหนัก. *วารสารเทคโนโลยีสุรนารี*, 6(2), 73-98.
- KM Tank. (2553). การรักษาความมั่นคงปลอดภัยของระบบสารสนเทศ. สืบค้นเมื่อ 2 กันยายน 2562, จาก <https://www.tistr.or.th/tistrblog/?p=703>.

ประวัติผู้เขียน

ชื่อ	นางสาวชฎาธาร รักสวัสดิ์
วัน เดือน ปีเกิด	6 กันยายน 2537
ที่อยู่ปัจจุบัน	25/1 หมู่ที่ 3 ตำบลวัดประตู่ อำเภอสุราษฎร์ธานี จังหวัดสุราษฎร์ธานี 84000
ประวัติการศึกษา	(2560) วิทยาศาสตรบัณฑิต สาขา วิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.38 (สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง)
ทุนการศึกษาที่ได้รับ	ทุนอุดหนุนค่าเล่าเรียน จาก คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานทางวิชาการ	<ol style="list-style-type: none"> เรื่อง ระบบต้นแบบการจัดการความเสี่ยงสำหรับการประเมินการติดตั้งซอฟต์แวร์ภายใต้มาตรฐาน COBIT, COSO (The prototype of Risk Management System for Implement Software) งานประชุมวิชาการระดับชาติ “นวัตกรรมและเทคโนโลยีวิชาการ 2017” “วิจัยจากองค์ความรู้สู่การพัฒนาอย่างยั่งยืน” ณ ITC 2017 Innovation and Technology Conference เรื่อง Software Testing System Development based on ISO 29119 ณ International Journal of Machine Learning and Computing (IJMLC, ISSN: 2010-3700) 2019 8th International Conference on Software and Computing Technology (Annual meeting of JCP April 5-7,2019, Hong Kong) เรื่อง การพัฒนาระบบวางแผนการทดสอบซอฟต์แวร์ตามวงจรการทดสอบซอฟต์แวร์และมาตรฐาน ISO 29119, IEEE 829 การประชุมเสนอผลงานวิจัยระดับชาติ มหาวิทยาลัยสุโขทัยธรรมาธิราช ครั้งที่ 9