

การวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งาน
ระบบสำหรับศูนย์บริการข้อมูลลูกค้า

THE ISSUE ANALYSIS, CHANGE REQUEST DESIGN
AND SYSTEM ROLL OUT FOR CONSOLIDATION
PORTAL SYSTEM ON CUSTOMER CONTACT CENTER



โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

การวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งาน
ระบบสำหรับศูนย์บริการข้อมูลลูกค้า

THE ISSUE ANALYSIS, CHANGE REQUEST DESIGN
AND SYSTEM ROLL OUT FOR CONSOLIDATION
PORTAL SYSTEM ON CUSTOMER CONTACT CENTER



โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE ISSUE ANALYSIS, CHANGE REQUEST DESIGN
AND SYSTEM ROLL OUT FOR CONSOLIDATION
PORTAL SYSTEM ON CUSTOMER CONTACT CENTER



COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE
IN COMPUTER SCIENCE FACULTY OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการสหกิจศึกษา	การวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งานระบบ สำหรับศูนย์บริการข้อมูลลูกค้า
ชื่อนักศึกษา	นายพัฒนayu แผลงพาลี รหัสนักศึกษา 56050326
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	ผศ.กฤษฎา บุศรา

บทคัดย่อ

โครงการสหกิจศึกษานี้นำเสนอเกี่ยวกับการวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งานระบบสำหรับศูนย์บริการข้อมูลลูกค้า ในเครือข่ายโทรศัพท์เคลื่อนที่แห่งหนึ่ง พัฒนาโดยบริษัท FusionWare Company Limited ซึ่งพัฒนาขึ้นเพื่อเชื่อมโยงข้อมูลจากระบบต่างๆมากกว่า 10 ระบบ มารวบรวมไว้ในระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) เพียงระบบเดียว โดยแบ่งเป็นกลุ่มบริการ เพื่ออำนวยความสะดวกแก่ผู้รับสายในระบบศูนย์บริการข้อมูลลูกค้า โดยมีวัตถุประสงค์เพื่อลดจำนวนหน้าจอกการทำงานจาก 10 ระบบ เหลือเพียง 4 หน้า เพื่อลดระยะเวลาในการรับสายของผู้ใช้งานระบบ (Agent) ในหนึ่งครั้งลง 9% และสามารถรองรับการทำงานสำหรับส่วนของการส่งเสริมการขาย (Promotion) ซึ่งมีความสามารถในการทำงานต่างๆ (Feature) ที่ระบบ สามารถทำงานได้ เช่น แสดงข้อมูลลูกค้า (Customer Information) แสดงการแจ้งเตือนในเรื่องต่างๆ (Alert) แสดงข้อมูลการใช้งาน และบริการเครือข่ายทางโทรศัพท์เคลื่อนที่ (Package and Service) ที่ลูกค้าใช้งานอยู่ การเปลี่ยนโปรโมชั่นหลัก (Main Promotion) และโปรโมชั่นเสริมต่างๆ (On Top) เป็นต้น

ในการทำโครงการสหกิจศึกษาเรื่องการวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งานระบบสำหรับศูนย์บริการข้อมูลลูกค้า สามารถทำการทดสอบได้โดยการเข้าใช้งานที่หน้า Front Office และ Back Office โดยในส่วนของ Front Office จะเป็นส่วนของการทำรายการ เมื่อมีลูกค้าโทรเข้ามา ผู้ใช้งานระบบ (Agent) จะทำการกรอกหมายเลขโทรศัพท์เพื่อทำการค้นหาข้อมูล และทำรายการเกี่ยวกับโปรโมชั่น และบริการต่างๆ จนเสร็จสิ้นการทำงาน และในส่วนของ Back Office จะเป็นส่วนของการจัดการข้อมูลต่างๆที่จะมาแสดงผลบน Front Office และเป็นส่วนที่กำหนดค่าต่างๆของระบบ ซึ่งผู้ใช้แต่ละคนจะมีสิทธิ์การเข้าใช้งานแตกต่างกันตามระดับของผู้ใช้งานระบบ (Agent)

คำสำคัญ : การบริการทางด้านเว็บไซต์ การบูรณาการ การปรับปรุงกระบวนการทางธุรกิจ
เว็บแอปพลิเคชัน สถาปัตยกรรมเชิงบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตัดอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	The Issue Analysis, Change Request Design And System Roll Out For Consolidation Portal System On Customer Contract Center
Students	Mr. Pattanayu Phlangpalee Student ID 56050326
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2016
Advisor	Asst.Prof. Kridsada Budsara

Abstract

The Issue Analysis, Change Request Design and System Roll Out for Consolidation Portal System on Customer Contract Center is a cooperative education project at FusionWare Company Limited. The objective of the project had developed to monitor decrease from 10 systems to 4 pages, make decrease time for agent work time 9% and support connection more than 10 systems about Promotion and Feature for customer contact center such as, show customer information, show alert, show packages and services, show agent log report, show agent role and change main packages and on top packages.

Cooperative education subject The Development of Promotion-Updating Batch Process and System Integration Test for Consolidation Portal System on Customer Contract Center divide 2 features is Front Office and Back Office. Front Office can do the order while customer incoming call to customer contact center. After that agent will input mobile phone number into textbox for verify about customer information. In the first page, there are customer information. The second page are about searching packages and services. The third page are about package management. Finally, there are about submit order to change packages or services.

Keywords : Web Service, Integration, Process Redesign, Web Application, Service-Oriented Architecture

กิตติกรรมประกาศ

จากการที่ข้าพเจ้าได้ไปโครงการสหกิจศึกษา โดยได้เข้าร่วมพัฒนาส่วนการวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งานสำหรับศูนย์บริการข้อมูลลูกค้า ตั้งแต่วันที่ 11 มกราคม 2559 ถึง กรกฎาคม 2559 กับทางบริษัท Fusionware Company Limited โดยใช้ภาษา Java เป็นภาษาหลัก ในการทำระบบ ซึ่งส่งผลให้ได้รับประสบการณ์การทำงานในชีวิตจริง และความรู้เพิ่มมากขึ้น ซึ่ง โครงการงานสหกิจศึกษาเล่มนี้จะเกิดขึ้นไม่ได้ หากไม่ได้รับการสนับสนุน และช่วยเหลือจากบุคลากร หลายฝ่ายดังนี้

1. คุณปรีชากร	ตั้งงาม	ตำแหน่ง	Project Manager
2. คุณภาสพงษ์	ไทยทัตกุล	ตำแหน่ง	Solution Architect
3. คุณสาคร	แสนเมือง	ตำแหน่ง	Developer Leader
4. คุณวริษา	ผดุง	ตำแหน่ง	Developer
5. คุณเมธาวี	สุทธิกุล	ตำแหน่ง	Developer
6. คุณภูธร	เกตุรัตน์	ตำแหน่ง	Developer
7. คุณธนวัฒน์	เชียงใหม่	ตำแหน่ง	Developer
8. คุณธีธัช	นิทราศรีสมุทร	ตำแหน่ง	Developer
9. คุณภาณุพัฒน์	ช่อลัดดา	ตำแหน่ง	Developer
10. คุณจรงค์	กองโปธิ	ตำแหน่ง	Quality Assurance
11. คุณสมิตรา	โพธิลา	ตำแหน่ง	Business Analyst
12. คุณสร้อยฟ้า	จันทร์แสงศรี	ตำแหน่ง	Business Analyst
13. คุณกฤษฎิ์ชนก	อินทร์กลัด	ตำแหน่ง	Business Analyst

ทางข้าพเจ้าต้องขอขอบพระคุณทุกท่านที่ให้การสนับสนุน และคอยช่วยเหลือตลอดการทำ โครงการงานสหกิจศึกษา ตลอดการปฏิบัติงานที่บริษัท มา ณ ที่นี้ด้วย

ขอขอบพระคุณ ผศ.กฤษณา บุศรา ที่ให้คำปรึกษาตลอดการทำโครงการงานสหกิจศึกษาเล่มนี้ และอบรมให้ความรู้แก่ข้าพเจ้า ในด้านการปฏิบัติงาน และแก้ปัญหาต่างในการทำงาน

สุดท้ายนี้ ขอขอบพระคุณบิดา มารดา สมาชิกในครอบครัว ครูบาอาจารย์ รวมทั้งผู้มีพระคุณ ที่คอยให้กำลังใจ และคอยสนับสนุนในเรื่องต่างๆ รวมถึงเพื่อนๆ ที่ให้คำปรึกษาตลอดมา

พัฒนayu แผลงพาลี 56050326

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และตัดอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตการดำเนิน	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 อุปกรณ์และเครื่องมือในการดำเนินงาน	3
1.6 ขั้นตอนการดำเนินงาน	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 ความเป็นมาของ Web Technology	5
2.2 Web Services	8
2.2.1 ความเป็นมาของ Web Service.....	8
2.2.2 ประโยชน์ของ Web Service.....	9
2.2.3 เทคโนโลยีที่ใช้ในการพัฒนา Web Services.....	9
2.3 Web Application	10
2.3.1 ความหมายของ Web Application.....	10
2.3.2 ลักษณะการทำงานของ Web Application.....	10
2.3.3 Rendering Engine	11
2.4 Software Configuration Management (SCM).....	12
2.4.1 ที่มาและนิยามของ Software Configuration Management (SCM).....	12
2.4.2 วัตถุประสงค์ของ Software Configuration Management (SCM).....	12
2.5 ความต้องการ (Requirement).....	12
2.5.1 ความต้องการด้านซอฟต์แวร์ (Software Requirement).....	13
2.5.2 เอกสารความต้องการด้านซอฟต์แวร์ (Software Requirement Document)	13
2.6 Configuration Control	15
2.6.1 ความหมายของ Configuration Control	15
2.6.2 วัตถุประสงค์ของ Configuration Control	15
2.6.3 ข้อมูลที่จำเป็นต่อการทำ Change Control	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.4 Configuration Control Process	15
2.6.4 ขั้นตอนในการทำ Configuration Control Process	16
2.7 Change Control Board	18
2.7.1 Change Control Board Team	18
2.8 การทดสอบระบบ (Testing)	21
2.8.1 ความแตกต่างระหว่าง Verification และ Validation	22
2.8.2 เทคนิคที่ใช้ในการทดสอบระบบ	22
2.8.3 ประเภทของการทดสอบโปรแกรม.....	22
2.8.4 ระดับทดสอบ Software Testing มี 4 ระดับ	22
2.8.5 รูปแบบการทดสอบระบบ	23
2.8.6 ขั้นตอนการจัดการความผิดพลาด (Life Cycle Defect Management).....	24
2.8.7 ความสำคัญของข้อผิดพลาด (Defect Priority).....	24
2.8.8 ระดับในการทดสอบระบบ	25
2.9 Defect Management	25
2.9.1 Defect	25
2.9.2 Defect Life Cycle	25
2.9.3 Defect Status	25
2.9.4 Defect Type	26
2.9.5 Defect Severity	26
2.9.6 Defect Tracking	27
2.10 Jira	27
2.10.1 ประเภทของงานใน Jira	28
2.10.2 ความสำคัญ (Priority).....	29
2.11 ระบบบริหารความสัมพันธ์ลูกค้า (Customer Relationship Management หรือ CRM)	30
2.11.1 ประโยชน์ของ CRM ต่อธุรกิจ.....	31
2.11.2 การเพิ่มประสิทธิภาพของเจ้าหน้าที่ Call Center ด้วยระบบ CRM	31
2.12 ฐานข้อมูล (Database)	32
2.12.1 ประโยชน์ของฐานข้อมูล	32
2.13 เครื่องมือ หรือโปรแกรมที่ใช้งาน (Tools).....	33
2.13.1 โปรแกรม Oracle SQLdeveloper.....	33
2.13.2 SOAP (Simple Object Access Protocol).....	33
2.13.2.1 จุดประสงค์ของ SOAP	34
2.13.2.2 โครงสร้างของ SOAP.....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

2.13.3 SVN (Subversion).....	37
2.13.3.1 Repository.....	40
2.13.3.2 คำสั่งใน SVN.....	40
2.13.4 DynaTrace AJAX Edition	41
บทที่ 3 วิธีการดำเนินงานวิจัย	42
3.1 การตรวจสอบและดูแลการทำงานของระบบในช่วง Go-Live (Monitor).....	42
3.2 การติดตามปัญหา (Defect Tracking).....	43
3.3 การทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request).....	43
3.4 การทำเอกสารส่งมอบ (Delivery).....	44
บทที่ 4 ผลการวิจัยและอภิปรายผล	46
4.1 ผลการดำเนินการของการตรวจสอบและดูแลการทำงานของระบบ ในช่วง Go-Live (Monitor)	46
4.2 ผลการดำเนินการของการติดตามปัญหา (Defect Tracking)	47
4.3 ผลการดำเนินการของการทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request)	47
4.4 ผลการดำเนินการของการทำเอกสารส่งมอบ (Delivery DOC.)	47
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	48
5.1 สรุปผลการวิจัย	48
5.2 ข้อเสนอแนะ	48
เอกสารอ้างอิง	49
ภาคผนวก	51
ภาคผนวก ก. การเริ่มต้นใช้งานโปรแกรม	
Oracle SQLdeveloper.....	52
ก.1 การเพิ่ม Connection ในฐานข้อมูล.....	53
ก.2 การ Export Schema ฐานข้อมูลไปใช้งานในเครื่องอื่น.....	56
ก.3 การ Import Schema ฐานข้อมูลที่ได้จากเครื่องอื่น	57
ก.4 การสร้าง Package.....	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์	14
2.2 Change Request Form (CRF)	19
2.3 แนวความคิดในการทดสอบระบบระหว่าง Programmer และ Tester	21
3.1 แบบฟอร์ม เอกสารส่งมอบ (Delivery) ส่วนของ Feature List.....	45



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 Host Base Architecture.....	5
2.2 Client – Server Architecture.....	6
2.3 Web Technology Architecture.....	7
2.4 Web Technology Architecture ยุคที่ 4.....	7
2.5 Web Service	8
2.6 XML (The Extensible Markup Language)	9
2.7 Web Application.....	10
2.8 Rendering Engines.....	11
2.9 Configuration control process	16
2.10 System Development Life Cycle	21
2.11 V Model	23
2.12 Jira	27
2.13 Jira Screen	29
2.14 ระบบบริหารความสัมพันธ์ลูกค้า (CRM Software).....	31
2.15 ตัวอย่างโปรแกรม Oracle sqlDeveloper	33
2.16 Simple Object Access Protocol.....	34
2.17 SOAP Envelope	35
2.18 SVN (Subversion).....	37
2.19 SVN cycle.....	38
2.20 Screenshot SVN.....	38
2.21 จัดเก็บไฟล์ใน SVN.....	39
2.22 Plugin ที่รองรับ	39
2.23 Repository.....	40
2.24 หน้าจอ DynaTrace AJAX Edition.....	41
3.1 Go-Live Cycle.....	42
ก.1 การเพิ่ม Connection ในฐานข้อมูล	53
ก.2 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]	53
ก.3 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]	54
ก.4 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]	54
ก.5 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]	55
ก.6 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]	55
ก.7 การ Export Schema ในฐานข้อมูล	56
ก.8 การ Export Schema ในฐานข้อมูล [ต่อ]	56
ก.9 การ Export Schema ในฐานข้อมูล [ต่อ]	57
ก.10 การ Import Schema ในฐานข้อมูล.....	57
ก.11 การ Import Schema ในฐานข้อมูล [ต่อ]	58

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงวิชาการเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.12 การ Import Schema ในฐานข้อมูล [ต่อ]	58
ก.13 การ Import Schema ในฐานข้อมูล [ต่อ]	59
ก.14 การ Import Schema ในฐานข้อมูล [ต่อ]	59
ก.15 การ Package ในฐานข้อมูล.....	60
ก.16 การ Package ในฐานข้อมูล [ต่อ]	61
ก.17 การ Package ในฐานข้อมูล [ต่อ]	61



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท Fusionware Company Limited ได้พัฒนาระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) ซึ่งเป็นระบบที่มีคุณสมบัติในการเชื่อมโยงข้อมูลจากระบบต่างๆ มากกว่า 10 ระบบ เพื่ออำนวยความสะดวกแก่ผู้รับสายในระบบศูนย์บริการข้อมูลลูกค้า ในเครือข่ายโทรศัพท์เคลื่อนที่แห่งหนึ่ง ระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) สามารถรองรับการทำงานสำหรับส่วนของการส่งเสริมการขายภายในประเทศ (Promotion) และ ส่วนของการส่งเสริมการขายต่างประเทศ (International Roaming) ซึ่งมีความสามารถในการทำงานต่างๆ (Feature) ที่ระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) สามารถทำงานได้ เช่น แสดงข้อมูลลูกค้า (Customer Information) แสดงโปรโมชั่นและบริการเครือข่าย ปัจจุบันที่ลูกค้าใช้งานอยู่ (Existing Package) แสดงการแจ้งเตือนในเรื่องต่างๆ (Alert) แสดงข้อมูลการใช้งานและบริการเครือข่ายทางโทรศัพท์ (Package and Service) การทำรายการการขายโปรโมชั่นหลัก (Main Promotion) และโปรโมชั่นเสริมต่างๆ (On Top) แสดงรายงานการทำ Order ของผู้ใช้งานระบบ (Agent) เป็นต้น โดยในระบบจะใช้ Batch Master Data เป็นส่วนที่ให้นำเข้าข้อมูลโปรโมชั่นจริงมาใช้ในการพัฒนาระบบ CRM เพื่อให้ข้อมูลที่ใช้ในการพัฒนามีความเหมือนจริงมากที่สุด สาเหตุที่มีการพัฒนาระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) เนื่องจากการทำงานของผู้ใช้งานระบบ (Agent) จากการรับสายในแต่ละครั้งจะต้องเรียกใช้งานข้อมูลจากหลากหลายหน้าจอจากหลากหลายระบบ เพื่อใช้ในการทำงานภายในหนึ่งครั้ง ซึ่งใช้ระยะเวลาในการรอการประมวลผลนานและมีเมนูตัวเลือกในระบบเดิมมากเกินไปจนเกิดความจำเป็น ลูกค้าที่ต้องการปรึกษาในเรื่องต่างๆ ต้องใช้เวลานานในการรอสาย ซึ่งเมื่อเปรียบเทียบระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) กับระบบเดิม ระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) จะทำให้ผู้ใช้งานระบบ (Agent) ใช้ระยะเวลาในการทำรายการในแต่ละครั้งลดลง 9% ลดจำนวนในการเปิดหน้าจอเพื่อการใช้งานเหลือเพียง 4 หน้า ลดความซ้ำซ้อนในการทำงาน สามารถตัดเมนูตัวเลือกหรือบางหน้าจอที่ไม่มี ความจำเป็น และลดระยะเวลาศึกษาระบบลง 25%

1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 เพื่อลดจำนวนหน้าจอการทำงานจากเดิมมากกว่า 10 ระบบ เหลือเพียง 4 หน้า ที่ผู้ใช้งานระบบ (Agent) ต้องใช้งานในการให้บริการลูกค้าในแต่ละสายที่โทรมา
- 1.2.2 เพื่อลดเวลาในการศึกษาระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) ของผู้ใช้งานในระบบศูนย์บริการข้อมูลลูกค้าลง 25%
- 1.2.3 เพื่อลดระยะเวลาในการทำงานของผู้ใช้งานระบบ (Agent) ต่อการรับสายภายในหนึ่งครั้งลง 9% ทำให้ผู้ใช้งานระบบ (Agent) หนึ่งคน สามารถรับสายได้เพิ่มขึ้นมากกว่าเดิม
- 1.2.4 เพื่อลดความผิดพลาดของผู้ใช้งาน (Agent) จากการทำงานแบบ Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตการดำเนินงาน

- 1.3.1 สามารถรวมระบบหลายๆระบบมาไว้ในระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) เพียงระบบเดียว เพื่อให้สะดวกต่อการใช้งาน
- 1.3.2 สามารถประหยัดเวลาในการศึกษาระบบของผู้ใช้งานระบบ (Agent) จากที่ตอนแรกต้องศึกษาหลายระบบ มาเป็นศึกษาระบบบูรณาการข้อมูลสำหรับศูนย์บริการข้อมูลลูกค้า (Consolidation Portal System) เพียงระบบเดียว
- 1.3.3 สามารถลดระยะเวลาในการทำงานของผู้ใช้งานระบบ (Agent) ต่อการรับสายภายในหนึ่งครั้ง เนื่องมาจากใช้ระยะเวลาในการเข้าถึงระบบลดน้อยลง
- 1.3.4 สามารถแสดงผลข้อมูลบนระบบตามที่ออกแบบมาได้อย่างถูกต้อง
- 1.3.5 ระบบสามารถรองรับการทำงานในการให้บริการลูกค้าแบบทั่วไป และแบบองค์กรรวมไปถึงรองรับประเภทเครือข่ายโทรศัพท์เคลื่อนที่ทั้งประเภท 2G 3G และ 4G
- 1.3.6 สามารถรองรับการใช้งานได้ทุกวันตลอด 24 ชั่วโมง
- 1.3.7 สามารถดึงข้อมูลมาเพื่อ Update ข้อมูลโปรโมชั่นประจำวันได้
- 1.3.8 ผู้ใช้งานระบบ (Agent) คือ พนักงานในการติดต่อกับลูกค้าและผู้ดูแลพนักงาน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 องค์กร
 - สามารถเพิ่มประสิทธิภาพขององค์กรในการให้บริการและดูแลลูกค้า
 - เพิ่มเครื่องมือในการทำงานของพนักงาน
 - เพิ่มผลกำไรขององค์กรที่ได้จากการบริการลูกค้า
- 1.4.2 ผู้ใช้งาน (Agent)
 - ผู้ใช้สามารถลดเวลาในการรอการประมวลผลของข้อมูลมาแสดงบนหน้าจอ เพื่อความสะดวกในการให้บริการแก่ลูกค้า
 - ผู้ใช้สามารถลดเวลาในการศึกษาระบบ และใช้ระยะเวลาในการเข้าถึงระบบลดน้อยลง
 - ผู้ใช้สามารถดูรายงานต่างๆ เพื่อประกอบการตัดสินใจในการบริหารจัดการงานได้
 - ลดความผิดพลาดซึ่งเกิดจากตัวของผู้ใช้งาน (Agent) ลง จากการทำงานแบบ Manual
 - สามารถเพิ่มประสิทธิภาพในการทำงานและทำงานได้ตรงตามเป้าหมาย
 - ลดความซับซ้อนของหน้าจอการทำงาน ทำให้ผู้ใช้ไม่สับสน
- 1.4.3 ผู้ศึกษาและวิจัย
 - ได้เรียนรู้การทำงานในสถานการณ์จริง
 - ได้รู้จักแก้ไขปัญหาในการทำงานร่วมกับผู้อื่น
 - เพิ่มทักษะในการสื่อสารกับผู้อื่น
 - เพิ่มคุณลักษณะของตนเอง เช่น การตรงต่อเวลา ความรับผิดชอบ ความอดทน
 - เพิ่มความสามารถในการใช้เครื่องมือต่างๆ ที่ใช้ในการทำงานจริง
 - ได้วิเคราะห์การทำงานของระบบ สามารถเข้าใจหลักการคิดของระบบได้ง่ายขึ้น
 - สามารถเข้าใจหลักการคิดในเชิงธุรกิจ
 - สามารถเข้าใจความต้องการของผู้ใช้งานระบบ และอธิบายให้ผู้อื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 อุปกรณ์และเครื่องมือในการดำเนินงาน

1.5.1 ฮาร์ดแวร์

- 1) เครื่องคอมพิวเตอร์แบบพกพา จำนวน 1 เครื่อง
 - หน่วยประมวลผลกลาง (CPU) Intel (R) Core (TM) i5-2450M CPU @ 2.50 GHz
 - หน่วยความจำหลัก (RAM) 6 GB
 - ฮาร์ดดิสก์ (HDD) 750 GB
- 2) จอคอมพิวเตอร์ จำนวน 1 เครื่อง

1.5.2 ซอฟต์แวร์สำหรับฝ่าย Business Analyst

- 1) ระบบปฏิบัติการ Window 8 (64 bit) และ Window 8.1 (64 bit)
- 2) Microsoft Office 2013
 - Microsoft Word 2013
 - Microsoft Excel 2013
- 3) Oracle SQL Developer
- 4) Internet Explorer (IE)
- 5) Microsoft Paint
- 6) Apache JMeter 2.13
- 7) Notepad++

1.5.3 ซอฟต์แวร์สำหรับฝ่าย Developer

- 1) ระบบปฏิบัติการ Window 8 (64 bits) และ Window 8.1 (64 bit)
- 2) Microsoft Office 2013
 - Microsoft Word 2013
- 3) Oracle SQL Developer
- 4) Internet Explorer (IE)
- 5) Red Hat JBoss Developer Studio 7.1.1GA
- 6) Java Development Kit Version 7 (JDK ver.1.7)

1.5.4 ซอฟต์แวร์สำหรับฝ่าย Tester

- 1) ระบบปฏิบัติการ Window 8 (64 bits) และ Window 8.1 (64 bit)
- 2) Microsoft Office 2013
 - Microsoft Word 2013
 - Microsoft Excel 2013
- 3) Oracle SQL Developer
- 4) Internet Explorer (IE)
- 5) Microsoft Paint

1.5.5 ภาษาที่ใช้

- 1) Java (Java Programming Language)
- 2) SQL (Structure Query Language)
- 3) XML (Extensible Markup Language)
- 4) HTML (Hyper Text Markup Language)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ขั้นตอนการดำเนินงาน

- 1.6.1 ศึกษาขอบเขตโครงการงานสหกิจศึกษา และข้อมูลต่างๆที่เกี่ยวข้อง รวมไปถึงโปรแกรม และอุปกรณ์ที่นำมาใช้
- 1.6.2 ศึกษาหน้าที่และความรับผิดชอบต่างๆของตนเอง
- 1.6.3 ทดสอบและลองใช้งานเว็บแอปพลิเคชันบนอุปกรณ์ที่นำมาใช้งานจริง
- 1.6.4 เก็บข้อมูลจากผู้ทดลองใช้เพื่อนำมาปรับปรุงและพัฒนาเว็บแอปพลิเคชัน
- 1.6.5 รวบรวมและบันทึกปัญหาจากการทำงานจริงของแอปพลิเคชันจากกลุ่มตัวอย่างผู้ใช้งาน
- 1.6.6 ทำความเข้าใจกับปัญหาที่เกิดขึ้นและสามารถอธิบายได้
- 1.6.7 จัดทำรูปเล่มโครงการเตรียมนำเสนอ
- 1.6.8 นำเสนอโครงการงานสหกิจศึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

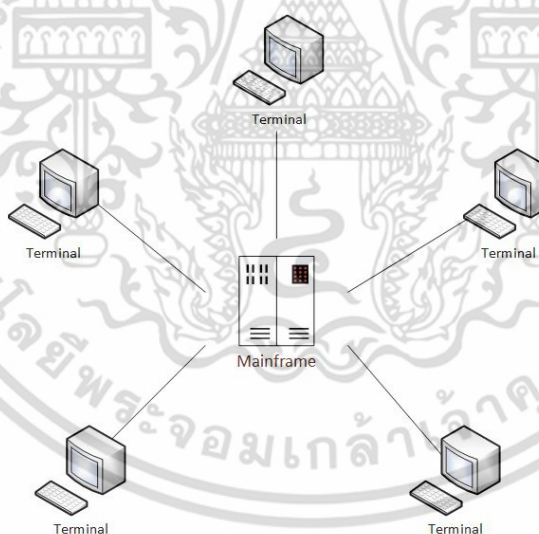
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงความรู้และทฤษฎีที่เกี่ยวข้องกับการโครงการงานสหกิจนี้ โดยผู้จัดทำได้ศึกษารวบรวมข้อมูลและนำความรู้เหล่านี้ไปประยุกต์ใช้ในการวิเคราะห์และพัฒนาโปรแกรม ซึ่งทฤษฎีที่เกี่ยวข้องที่ได้รวบรวมไว้มีดังนี้

2.1 ความเป็นมาของ Web Technology

ก่อนจะกล่าวถึงเทคโนโลยีเกี่ยวกับ Website ทางผู้จัดทำอยากให้ทราบถึงความเป็นมาของสถาปัตยกรรมต่างๆก่อน เนื่องจากว่า เมื่อทราบความเป็นมาแล้ว จะทำให้เข้าใจการทำงานเพิ่มมากขึ้น โดยการเกิด Website มีลำดับการพัฒนา ดังนี้

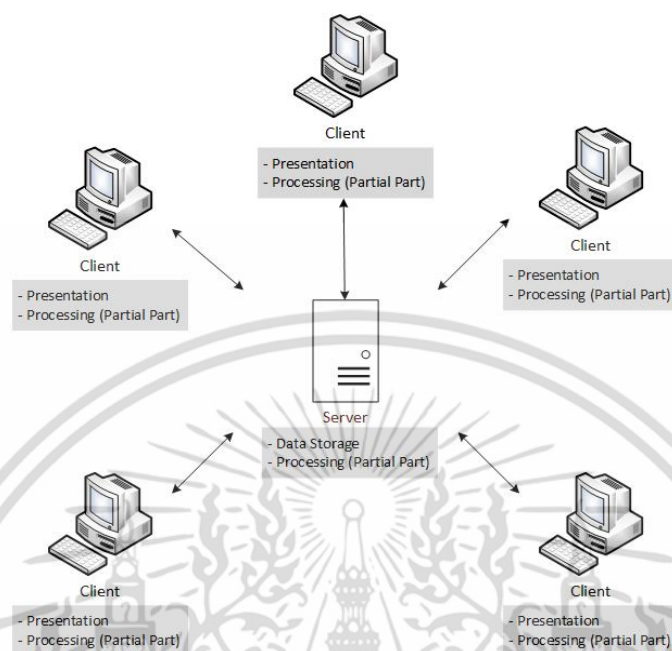
- 1) Mainframe (Centralize): ยุคนี้จะเริ่มจากการที่มี Mainframe เครื่องเดียวเป็นจุดศูนย์กลาง มีหน้าที่จัดการการทำงานต่างๆของระบบ ไม่ว่าจะเป็นการประมวลผล หรือจัดเก็บข้อมูล โดยผู้ใช้งานระบบ (User) สามารถเข้าใช้งานได้ผ่านเครื่อง Terminal เนื่องจากการที่ Mainframe เป็นศูนย์กลางเครื่องเดียวที่จัดการการทำงานต่างๆ จึงทำให้เกิดปัญหาคอขวด (Bottle Neck) และเครื่อง Mainframe มีค่าใช้จ่ายสูง



รูปที่ 2.1 Host Base Architecture

- 2) Client – Server (Distribute): ยุคนี้เป็นยุคที่พัฒนาถัดจากยุค Host Base ซึ่งจะแยกการทำงานระหว่างส่วนประมวลผล และส่วนจัดเก็บข้อมูล โดยที่จะกระจายการทำงานแต่ละส่วนไปให้ทั้งฝั่ง Client และ Server เพื่อไม่ให้ส่วนใดส่วนหนึ่งทำงานมากจนเกินไป ซึ่งฝั่ง Client อาจจะเป็นส่วนที่ประมวลผลข้อมูลบางส่วน และนำเสนอข้อมูล ส่วนฝั่ง Server อาจจะเป็นส่วนที่จัดเก็บข้อมูล และประมวลผลบางส่วนก็ได้ ซึ่งการกระจายงานเช่นนี้จะสามารถทำให้ Software จากแหล่งที่มาต่างๆ สามารถติดต่อสื่อสารกันได้ โดยที่ไม่จำเป็นต้องเป็น Software เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

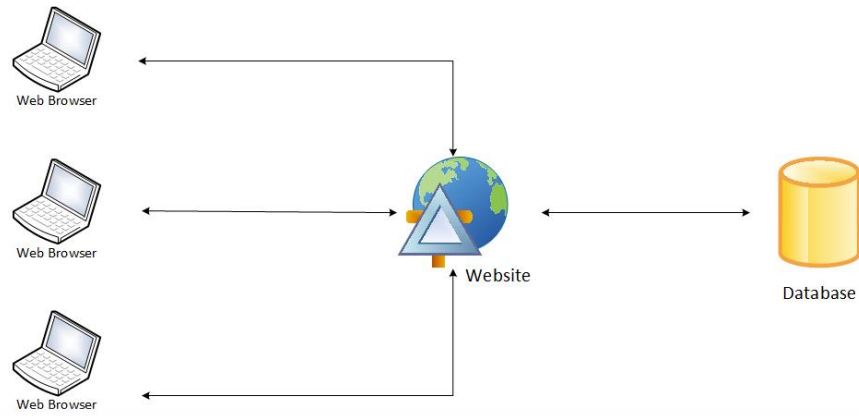
จากแหล่งเดียวกัน แต่ยุคของ Client – Server มีข้อจำกัดคือ เมื่อทำการเปลี่ยนแปลงข้อมูลในแต่ละส่วน จะต้องทำการปรับเปลี่ยนข้อมูลให้ตรงกันทั้งหมด ทำให้เสียระยะเวลา และอาจเกิดความซ้ำซ้อนของข้อมูลได้



รูปที่ 2.2 Client – Server Architecture

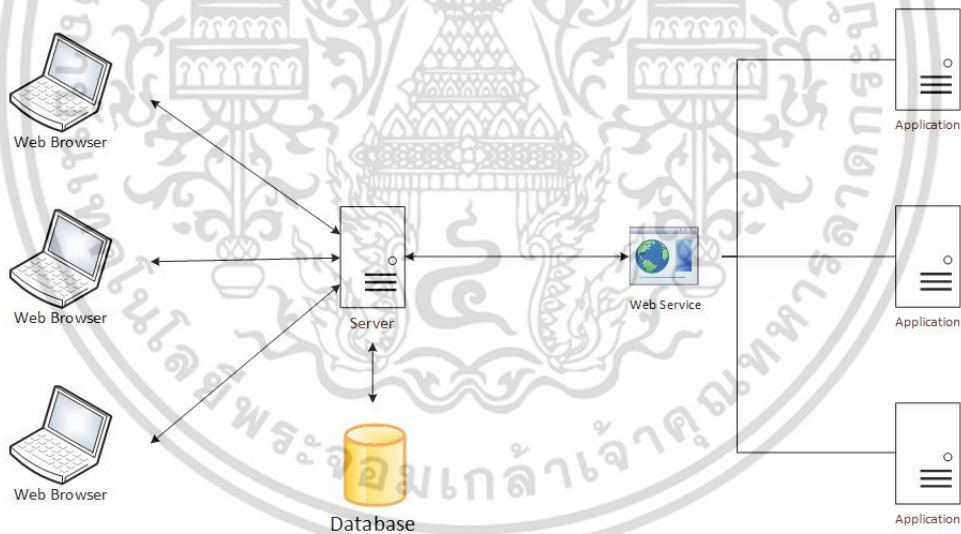
3) Web Technology: เป็นยุคที่มีการแสดงข้อมูลต่างๆผ่านทาง Website สามารถอ่านได้เพียงอย่างเดียว ซึ่ง Website พัฒนาขึ้นโดยใช้ภาษา HTML (Hyper Text Markup Language) ส่งข้อมูลผ่านทาง HTTP Protocol และมีหน้าที่ในการประมวลผลการทำงานต่างๆ โดยที่ผู้ใช้สามารถเข้าใช้งานได้ผ่านทาง Web Browser ผ่านระบบ Internet เมื่อ Website ทำการประมวลผลเรียบร้อยแล้ว ก็จะส่งข้อมูลไปจัดเก็บลงในฐานข้อมูล (Database) ซึ่งยุค Web Technology มีข้อดีคือ ผู้ใช้งานระบบ (User) สามารถเข้าถึงข้อมูลจากแหล่งต่างๆได้ แต่เนื่องจากปัจจุบันเป็นยุคที่ต้องการให้มีการส่งข้อมูลผ่านทาง Website ได้ ไม่เพียงแต่แสดงหน้า Website เพียงอย่างเดียว ซึ่งไม่สามารถแก้ไขได้ จึงทำให้เกิดยุคถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 Web Technology Architecture

4) ยุคที่ 4 นี้เป็นสถาปัตยกรรมที่ใช้ในปัจจุบัน ซึ่งยังไม่มีชื่ออย่างเป็นทางการ ยุคนี้เป็นยุคที่ได้รับ การพัฒนาจากยุค Web Technology ซึ่งเป็นยุคที่มีการแสดงข้อมูลเพียงอย่างเดียว แต่ในยุค ที่ 4 นี้ จะเป็นยุคที่มีการแลกเปลี่ยนข้อมูลการด้วย ทำให้หน้า Website มีการเปลี่ยนแปลงข้อมูลที่ ทันสมัยขึ้นอยู่เสมอ และเป็นยุคที่ทำให้ Application ต่างๆสามารถแลกเปลี่ยนข้อมูลกันได้ โดยไม่จำเป็นต้องเป็น Platform เดียวกัน ซึ่งเป็นที่มาของ Web Service ด้วย



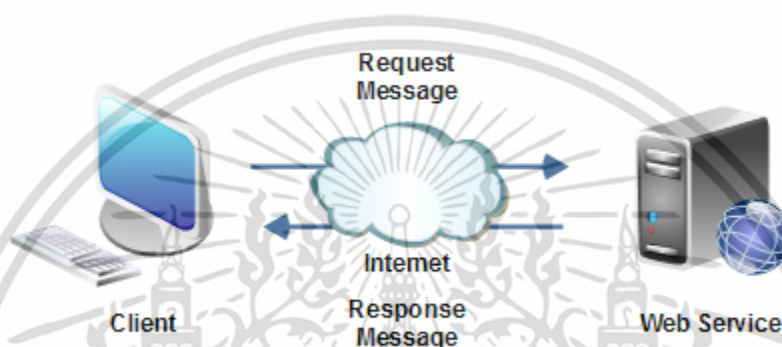
รูปที่ 2.4 Web Technology Architecture ยุคที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Web Services

2.2.1 ความหมายของ Web Service

Web Service คือ Application หรือ Program ที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจาก application อื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการให้บริการ จะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นตัวกลางในการแลกเปลี่ยนคือ XML ทำให้สามารถเรียกใช้ component ใด ๆ ก็ได้ ใน platform ใด ๆ ก็ได้ บน protocol HTTP ซึ่งเป็น protocol สำหรับ World Wide Web อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกัน ระหว่าง application กับ application ในปัจจุบัน



รูปที่ 2.5 Web Service

2.2.2 ประโยชน์ของ Web Service

1) Web Service ช่วยให้การเข้าถึงข้อมูลสารสนเทศจาก Application ที่ต่างกันเป็นไปโดยง่าย โดย Application นั้นๆ สามารถเขียนด้วย Java และรันอยู่บน Sun Solaris Application Server หรืออาจจะเขียนด้วย C++ และรันอยู่บน Windows NT หรืออาจจะเขียนด้วย Perl และรันอยู่บนเครื่อง Linux ซึ่งมาตรฐานของ Web Service ทำให้อินเทอร์เน็ตของ Application เหล่านี้ ถูกอธิบายโดย WSDL และทำให้อยู่ในมาตรฐานของ UDDI หลังจากนั้น จึงสามารถติดต่อสื่อสารถึงกันโดย XML ผ่าน SOAP อินเทอร์เน็ต

2) Web Service สามารถถูกเรียกใช้ภายในองค์กรเองหรือจากภายนอกองค์กร โดยผ่านไฟร์วอลล์ ดังนั้นจึงมีองค์กรใหญ่ๆ มากมาย กำลังพัฒนาระบบที่มีอยู่ของตน ให้เข้ากับ Web Service ซึ่งนับเป็นการลงทุนที่คุ้มค่า เนื่องจาก Web Service สามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กรได้อีกทางหนึ่ง

3) นอกจากนั้น Web Service ยังสามารถใช้ร่วมกับ Web Application โดยส่งผ่านข้อมูลทางอินเทอร์เน็ตได้อีกด้วยซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้าหรือหุ้นส่วน ถึงแม้จะต้องคำนึงถึงระบบรักษาความปลอดภัย และการจัดการรายการของข้อมูลอยู่ก็ตาม แต่ Web Service ได้ใช้มาตรฐานทั่วไปของ internet เรื่องดังกล่าวจึงนับเป็นเรื่องธรรมดาของการสื่อสารผ่านระบบอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 เทคโนโลยีที่ใช้ในการพัฒนา Web Services

XML (The Extensible Markup Language 1.0)

เป็นภาษา Markup ที่เป็น text-based ซึ่งทำให้เป็น มาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตอย่างรวดเร็ว ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ในการสร้าง เว็บไซต์ที่สามารถแสดงผลได้โดยโปรแกรมเว็บเบราว์เซอร์ แต่ XML จะใส่ tags ได้อย่างอิสระ แล้วทำการส่ง XML ชุดนี้ไป ประมวลผลยังแอปพลิเคชันใด ๆ ที่สามารถใช้ข้อมูลใน XML นี้ XML เป็นภาษาที่มีลักษณะเป็น tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกัน ทำให้ข้อมูลระหว่าง tag สามารถนำไปประมวลผลต่อได้ เช่น

```
<ComputerBook>
```

```
  <book>
```

```
    <name>เว็บเซอร์วิส</name>
```

```
    <price>10.00$</price>
```

```
  </book>
```

```
  <book>
```

```
    <name>xml</name>
```

```
    <price>10.00$</price>
```

```
  </book>
```

```
</ComputerBook>
```

จากตัวอย่าง จะเห็นได้ว่ารูปแบบเอกสาร xml ไม่ได้บอกวิธีแสดงผลไว้ แต่เอกสารสามารถสื่อความหมายให้คอมพิวเตอร์เข้าใจได้ และนำค่าไปประมวลผลต่อได้ แต่ถ้าต้องการจะแสดงผลต้องใช้ควบคู่กันกับ HTML



รูปที่ 2.6 XML (The Extensible Markup Language)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Web Application

2.3.1 ความหมายของ Web Application

Web Application คือ การพัฒนาระบบงานบนเว็บ ซึ่งมีข้อดีคือ ข้อมูลต่าง ๆ ในระบบมีการไหลเวียนในแบบOnline ทั้งแบบ Local (ภายในวง LAN) และ Global (ออกไปยังเครือข่ายอินเทอร์เน็ต) ทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time ระบบมีประสิทธิภาพ แต่ใช้งานง่าย เหมือนกับท่านทำกำลังท่องเว็บ ระบบงานที่พัฒนาขึ้นมาจะตรงกับความต้องการกับหน่วยงาน หรือห้างร้านมากที่สุด ไม่เหมือนกับโปรแกรมสำเร็จรูปทั่วไป ที่มักจะจัดทำระบบในแบบกว้าง ๆ ซึ่งมักจะไม่ตรงกับความต้องการที่แท้จริง ระบบสามารถโต้ตอบกับลูกค้า หรือผู้ใช้บริการแบบ Real Time ทำให้เกิดความประทับใจ เครื่องที่ใช้งานไม่จำเป็นต้องติดตั้งโปรแกรมใด ๆ เพิ่มเติมทั้งสิ้น ตัวอย่างระบบงานที่เหมาะสมกับเว็บ แอปพลิเคชัน เช่น ระบบการจองสินค้าหรือบริการต่าง ๆ เช่น การจองที่พัก การจองโปรแกรมทัวร์ การจองแผ่น CD-DVD ฯลฯ ระบบงานบุคลากร ระบบงานแผนการตลาด ระบบการสั่งซื้อแบบพิเศษ ระบบงานในโรงเรียน เช่น ระบบงานวัดและประเมินผล ระบบงานปกครอง ระบบงานห้องสมุด ระบบการลงทะเบียน เช็คเกรด ฯลฯ ระบบงานอื่น ๆ ที่ต้องการนำข้อมูลมา Online ค่าใช้จ่ายในการทำเว็บ แอปพลิเคชัน ปกติจะใช้วิธีการคำนวณจากขอบเขตของระบบงาน และปริมาณของข้อมูลที่ไหลเวียนในระบบ รวมถึงปัจจัยด้านอื่น ๆ ซึ่งทางเว็บโปรแกรมเมอร์จะคำนวณราคาออกเป็นงาน ๆ ไป ซึ่งส่วนใหญ่จะมีค่าใช้จ่ายต่าง ๆ ต่อปีนี้รวมกัน ค่าจัดทำระบบงาน ค่าชื่อโดเมน และ Web Hosting (ในกรณีจะนำระบบออกทางเครือข่ายอินเทอร์เน็ต) ค่าบริการหลังการขาย ค่า Hardware และอุปกรณ์ด้านเครือข่าย เพิ่มเติม อื่น ๆ



รูปที่ 2.7 Web Application

2.3.2 ลักษณะการทำงานของ Web Application

การทำงานของ Web Application นั้นโปรแกรมส่วนหนึ่งจะวางตัวอยู่บน Rendering Engine ซึ่งตัว Rendering Engine จะทำหน้าที่หลักๆ คือนำเอาชุดคำสั่งหรือรูปแบบโครงสร้างข้อมูลที่ใช้ในการแสดงผล นำมาแสดงผลบนพื้นที่ส่วนหนึ่งในจอภาพ โปรแกรมส่วนที่วางตัวอยู่บน Rendering Engine จะทำหน้าที่หลักๆ คือการเปลี่ยนแปลงแก้ไขสิ่งที่แสดงผล จัดการตรวจสอบข้อมูลที่รับเข้ามาเบื้องต้นและการประมวลผลบางส่วนแต่ส่วนการทำงานหลักๆ จะวางตัวอยู่บนเซิร์ฟเวอร์ ในลักษณะ Web Application แบบเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Rendering engine

Rendering engine หรือ ตัววาดหน้าเว็บ เป็นซอฟต์แวร์ที่นำข้อมูลเนื้อหาเว็บในลักษณะ HTML XML ภาพ วิดีโอ ไฟล์ รวมถึงข้อมูลในรูปแบบอื่น มาแสดงและจัดรูปแบบบนมอนิเตอร์แก่ผู้ใช้ โดยตัว rendering engine นี้มักจะใช้กับ Web browser หรือ Application ที่ต้องการแสดงผลข้อมูลเนื้อหาของเว็บ



รูปที่ 2.8 Rendering Engines

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Software Configuration Management (SCM)

2.4.1 ที่มาและนิยามของ Software Configuration Management (SCM)

การเปลี่ยนแปลงเป็นสิ่งที่ไม่อาจหลีกเลี่ยงได้ เมื่อมีการพัฒนาซอฟต์แวร์เกิดขึ้น อาจเนื่องด้วยเหตุผลที่ว่า มีการเปลี่ยนแปลงความต้องการของลูกค้าเอง (Requirement) หรือตัวนักพัฒนานั้นมีความต้องการที่จะพัฒนาหรือปรับปรุงตัวซอฟต์แวร์นั้นขึ้นมาใหม่ และเมื่อเกิดการพัฒนาซอฟต์แวร์ขึ้นย่อมจะทำให้มีผลกระทบต่อชิ้นงานเดิม (Project) ในทุกส่วน ไม่ว่าจะเป็น System Model, Source Code หรือ Document แล้วองค์กรหรือตัวนักพัฒนาซอฟต์แวร์เองนั้นจะมีวิธีการรับมือต่อการเปลี่ยนแปลงที่เกิดขึ้นได้อย่างไร SCM คือคำตอบในการรับมือกับปัญหานี้ คำนิยามของ Software Configuration Management (SCM) " SCM is the discipline of managing and controlling change in the evolution of Software System " From IEEE Standard 1042-1987 SCM คือ ข้อกำหนดเพื่อสร้างมาตรฐานในการจัดการและควบคุมการเปลี่ยนแปลงในส่วนของวิวัฒนาการของการพัฒนางานด้านซอฟต์แวร์

2.4.2 วัตถุประสงค์ของ Software Configuration Management (SCM)

วัตถุประสงค์ของ Software Configuration Management (SCM) เพื่อสร้างมาตรฐานและสามารถบำรุงรักษาความถูกต้องของตัวผลิตภัณฑ์ทางด้านซอฟต์แวร์ในทุก วัฏจักรการทำงาน โดยหลักสำคัญ SCM นั้นตั้งใจที่จะลดความสับสนและข้อผิดพลาดต่างๆที่เกิดขึ้น อันเนื่องมาจากเกิดความแตกต่างในแต่ละเวอร์ชันของซอฟต์แวร์

2.5 ความต้องการ (Requirement)

ความต้องการ ถือเป็นวัตถุประสงค์สำคัญในการผลิตซอฟต์แวร์ เพื่อสร้างข้อกำหนดความต้องการของลูกค้า เพื่อให้ซอฟต์แวร์ที่ถูกพัฒนาขึ้นตรงกับความต้องการที่แท้จริง จำแนกความต้องการด้านซอฟต์แวร์ได้เป็น 2 ระดับ คือ

1. ความต้องการของผู้ใช้ (User Requirement)

แสดงถึงความคาดหวัง ในบริการ หรือการทำงานที่ได้จากระบบและเงื่อนไขที่ระบบจะต้องทำตาม ถือเป็นความต้องการในระดับสูงสุด

2. ความต้องการด้านระบบ (System Requirement)

เป็นการกำหนดการทำงาน ฟังก์ชัน และบริการต่างๆ ของระบบในระดับรายละเอียด และจัดทำเป็นข้อกำหนดหน้าที่ของระบบ (Functional Specification)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 ความต้องการด้านซอฟต์แวร์ (Software Requirement)

มีความสัมพันธ์อย่างมากกับความต้องการของระบบ เพราะเป็นการรวบรวมคุณสมบัติ ด้านเทคนิคของซอฟต์แวร์ที่แสดงถึงคำสั่งและบริการที่ซอฟต์แวร์สามารถทำได้ภายใต้ระบบหนึ่งๆ ความต้องการด้านซอฟต์แวร์ แบ่งออกเป็น 3 ประเภท ได้แก่

1. ความต้องการที่เป็นหน้าที่หลัก (Functional Requirement)

เป็นความต้องการที่เป็นหน้าที่หลัก ซึ่งทำหน้าที่ใดๆ ตามที่กำหนดไว้ในส่วนการทำงานหรือบริการที่ซอฟต์แวร์นั้นควรมี

2. ความต้องการที่ไม่ใช่หน้าที่หลัก (Non- Functional Requirement)

เป็นความต้องการที่ไม่เกี่ยวข้องกับโดยตรงกับหน้าที่ หรือฟังก์ชันหลักของระบบ เช่น ความต้องการด้านผลิตภัณธ์ ความต้องการขององค์กร และความต้องการจากปัจจัยภายนอก เช่น การทำงานร่วมกัน ด้านกฎหมาย และด้านจริยธรรม

3. ความต้องการด้านธุรกิจ (Domain Requirement)

เป็นความต้องการที่เกี่ยวข้องกับงานหลักของระบบธุรกิจ ที่ต้องการซอฟต์แวร์มาสนับสนุน โดยเฉพาะ ซึ่งอาจเป็นเงื่อนไขของฟังก์ชันใดๆ หรือเงื่อนไขที่ใช้คำนวณหาผลลัพธ์ใดๆ ของระบบ

2.5.2 เอกสารความต้องการด้านซอฟต์แวร์ (Software Requirement Document)

เอกสารความต้องการด้านซอฟต์แวร์ (Software Requirement Document) เรียกได้อีกอย่างหนึ่งว่า “ข้อกำหนดความต้องการซอฟต์แวร์” (Software Requirement Specification : SRS) เป็นเอกสารข้อกำหนดความต้องการอย่างเป็นทางการ ที่จะบอกให้ทีมพัฒนาซอฟต์แวร์ทราบว่าต้องพัฒนาอะไรบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์

การควบคุมการเปลี่ยนแปลงเอกสาร					
ชื่อเอกสาร		Software Requirement Specification (SRS)			
เวอร์ชันปัจจุบัน					
วันที่ที่แก้ไขเวอร์ชันล่าสุด					
ผู้จัดทำเอกสารนี้					
เวอร์ชัน	วันที่	หมายเลขเอกสาร Change Request Form	หมายเหตุ	ผู้แก้ไข	รายการที่ทำการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Configuration Control

2.6.1 ความหมายของ Configuration Control

Configuration Control เป็นกิจกรรมที่ทำค่อนข้างบ่อยและเป็นกิจกรรมที่เกี่ยวข้องกับคนจำนวนมาก CI ที่เกี่ยวข้องก็มีจำนวนมาก เพราะฉะนั้น Configuration Control จึงจำเป็นต้องมีการวางแผนเป็นอย่างดี กิจกรรมที่สำคัญได้แก่ การประเมิน การประสานงาน การพิจารณาอนุมัติ หรือไม่อนุมัติ และการดำเนินการเปลี่ยนแปลงชิ้นงาน หลังจากทีชิ้นงานนั้นๆ ได้ผ่านกระบวนการ Configuration Identification อย่างถูกต้อง สมบูรณ์

2.6.2 วัตถุประสงค์ของ Configuration Control

เพื่อควบคุมกระบวนการเปลี่ยนแปลงให้เป็นระเบียบ นอกจากนี้ยังทำเพื่อลดความผิดพลาดของการดำเนินการ และ ให้ติดตามข้อมูลต่างๆ ให้ได้โดยง่าย

2.6.3 ข้อมูลที่จำเป็นต่อการทำ Change Control

- ตัวโครงการ โครงการ วันที่ ชื่อผู้ร้องขอ และความสำคัญ (Priority)
- ตัว CI คำอธิบายถึง CI ที่ต้องการเปลี่ยนแปลง
- คำอธิบายลักษณะการเปลี่ยนแปลงที่ต้องการ
- ถ้าผู้ร้องขอมีแนวทางในการปรับแก้ ก็ให้ระบุมาด้วย
- ผลของการเปลี่ยนแปลง
- ต้องมีการรับรอง พร้อมลายเซ็นกำกับด้วย
- วันที่ทำการดำเนินการ
- วันที่ของการทดสอบและreview

2.6.4 Configuration control process

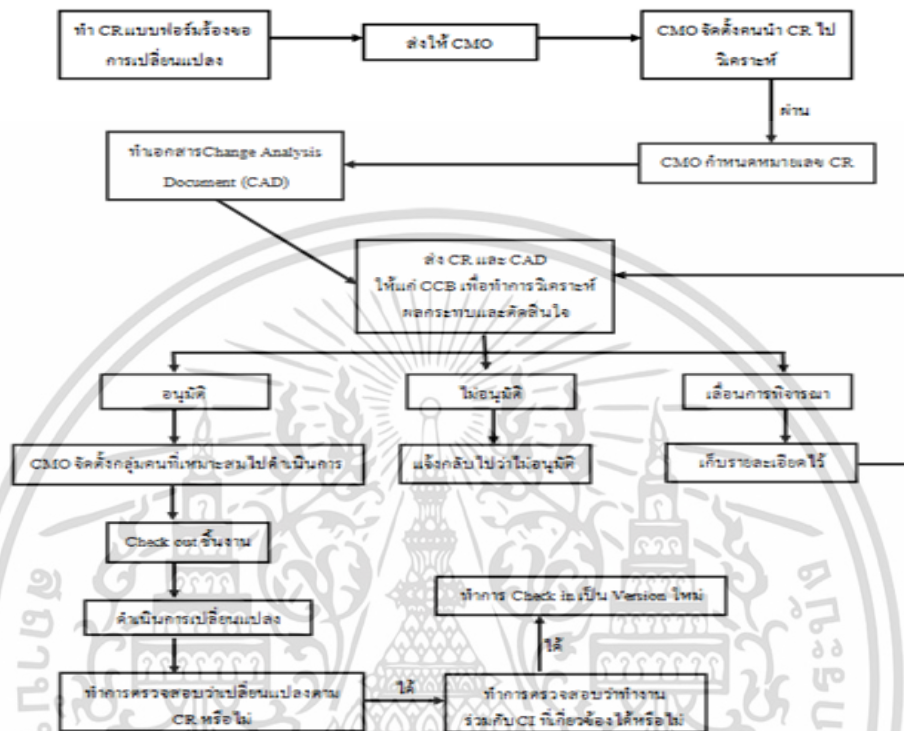
- 1) เมื่อมีการเกิดความต้องการเปลี่ยนแปลง ต้องมีการกรอกแบบฟอร์ม change request (CR) ซึ่งเป็นแบบฟอร์มร้องขอการเปลี่ยนแปลง
- 2) ส่ง CR ไปที่ change manager officer (CMO) รับ change request โดย CMO เป็นกลุ่มคน หรือ บุคคลเดียวก็ได้ ทำหน้าที่ในการตรวจสอบความชัดเจน สมบูรณ์เบื้องต้น
- 3) CMO นำเรื่องมาตรวจสอบเบื้องต้นก่อน และ มีการจัดตั้งกลุ่มคนเพื่อนำ CR ไป วิเคราะห์และประมวลผล ดูว่าสมเหตุผลหรือไม่ จากนั้นก็ให้ทำรายงาน Change 4) Analysis Document เป็นเอกสารรายงานการวิเคราะห์ การร้องขอการเปลี่ยนแปลง
- ส่งรวบรวม CR และ Change analysis document ให้กลุ่มคนที่ทำหน้าที่ในการวิเคราะห์ผลกระทบและตัดสินใจในการดำเนินงานกับการเปลี่ยนแปลงนั้นๆ (CCB)
- 5) CCB ทำการวิเคราะห์และตัดสินใจได้ 3 ลักษณะ ดังนี้

- ปฏิเสธ ว่าการเปลี่ยนแปลงนั้นไม่มีเหตุผลเพียงพอในการดำเนินการ ผู้รับผิดชอบในการเปลี่ยนแปลงต้องแจ้งผู้ทำ CR ทราบถึงเหตุผลของการปฏิเสธการเปลี่ยนแปลงนี้
- อนุมัติ ยอมรับให้มีการเปลี่ยนแปลงตามที่ร้องขอ จากนั้นก็จัดตั้งกลุ่มบุคคล เพื่อไปดำเนินการให้เป็นไปตามที่ร้องขอใน CR นั้น แล้วทำการ check-out ชิ้นงาน ออกจาก library สำหรับ CI ที่ต้องการเปลี่ยนแปลง จากนั้น ก็ทำการเปลี่ยนแปลง และตรวจสอบอีก

เอกสารนี้เป็นเอกสารทบทวนวัสดุสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งหนึ่งว่าเป็นไปตามที่ร้องขอหรือไม่ และสามารถทำงานกับ องค์ประกอบอื่นๆ ได้ จากนั้นก็สร้าง CI นั้นเป็น version ใหม่

- เลื่อนการพิจารณา กรณีการเปลี่ยนแปลงยังไม่ถึงเวลา หรือเลื่อนการดำเนินการไปก่อน ต้องมีการเก็บ detail ถึงการขอเปลี่ยนแปลงใน ฐานข้อมูลเพื่อดำเนินการในภายหลัง



รูปที่ 2.9 Configuration control process

2.6.5 ขั้นตอนในการทำ Configuration Control Process

1) การเตรียมการ และ รับเรื่องการร้องขอเปลี่ยนแปลง (Change initiation)

เนื่องจากการเปลี่ยนแปลงเกิดจากทั้ง internal (tester , developer, Quality Manager) และ external (user) มีบุคคลที่ต้องการเปลี่ยนแปลงมากมาย ดังนั้นจึงต้องมีมาตรฐานในการร้องขอการเปลี่ยนแปลง ต้องมีการกำหนดแบบฟอร์ม change request ให้ผู้ที่ต้องการร้องขอการเปลี่ยนแปลง นำข้อมูลไปวิเคราะห์ ประเมินต่อไปต้องมีการเตรียมการบุคคลที่รับผิดชอบควบคุมในการเปลี่ยนแปลง ซึ่งบุคคลที่ว่าคือ CMO ผู้ที่เป็นคนรับเรื่อง มีหน้าที่คือ

- คอยรับเรื่อง และ scan CR เบื้องต้นว่ามีความถูกต้อง สมบูรณ์ หรือไม่
- CMO เป็นผู้ให้หมายเลข change request
- บันทึกข้อมูลการขอเปลี่ยนแปลงเข้าไปใน ฐานข้อมูล เพื่อการอ้างอิงในภายหลัง
- CMO ทำการปฏิเสธเบื้องต้น หากกรณีเป็นไปได้ที่จะทำการเปลี่ยนแปลง การ reject ได้เลย

2) การจัดกลุ่ม และ จัดระดับความสำคัญของการร้องขอเปลี่ยนแปลง

(Change Classification)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดกลุ่ม หรือ ระดับของการเปลี่ยนแปลง โดยผู้วิเคราะห์หรือประเมินจะเป็นคนกำหนด ดังเกณฑ์ต่อไปนี้

- ความรุนแรงของ CR (severity)
- ความสำคัญ (importance)
- ผลกระทบ (impact)
- ค่าใช้จ่าย (cost)
- อื่นๆ (other)

3) การประเมิน และ วิเคราะห์ผลกระทบต่อ CR นั้นๆ (Change Evaluation or Change analysis)

วิเคราะห์ผลกระทบในด้านต่างๆ ได้แก่ ฟังก์ชันของระบบมีผลกระทบหรือไม่ อย่างไรกับระบบอื่นๆ ค่าใช้จ่าย ตารางเวลาการทำงาน และ สัญญาที่ทำได้กับลูกค้า

ประเมินผลที่อาจเกิดกับซอฟต์แวร์ในด้าน ความปลอดภัย ความเสถียรภาพของระบบ การบำรุงรักษา และการถ่ายโอนจากของเก่ามาเป็นของใหม่

การประเมินจาก change analysis document ถ้าเห็นว่า CR นั้นมีเหตุผลเพียงพอต่อการเปลี่ยนแปลง ก็ทำการ evaluation/analysis ต่อได้

4) การจัดการหรือการตัดสินใจ ดำเนินการต่อการร้องขอการเปลี่ยนแปลง (Change disposition)

- เป็นกระบวนการในการนำผลการวิเคราะห์มาตัดสินใจ
- CCB ต้องรับผิดชอบในการตัดสินใจ โดย CCB ต้องมี power/priority สูง อาจเป็นตัวแทนจาก PM (Project Manager), development manager, Configuration manager, Quality management Manager, Marketing Manager, Customer
- CCB จะตัดสินใจในการร้องขอที่สำคัญ ๆ เท่านั้น ส่วนพวก minor change (bugs fixing) โครงการจะให้ CMO เป็นคนตัดสินใจคนเดียว

5) การดำเนินการเปลี่ยนแปลงให้เป็นไปตามที่ร้องขอ (Change implementation)

- การมอบหมายให้ CMO ไปดำเนินการต่อให้เป็นไปตามที่ร้องขอ
- CMO มอบหมายให้บุคคลหรือกลุ่มบุคคลที่เหมาะสม ไปจัดการให้เป็นไปตาม CR
- ต้องดูว่า การเปลี่ยนแปลงนั้น ใช้เวลาเท่าใด และจัดตารางการทำงาน
- ต้องดูว่า item ใดบ้างที่ได้รับผลกระทบ
- Check-out item ที่เกี่ยวข้อง และเปลี่ยนแปลงสถานะของ item นั้นๆ ว่านำออกด้วยสาเหตุใด
- ดำเนินการตามที่ร้องขอการเปลี่ยนแปลง

6) การทบทวน ตรวจสอบ และ การเปลี่ยนแปลงที่เกิดขึ้น (Change Verification)

เมื่อทำการปรับแก้ CI เสร็จเรียบร้อยแล้ว ให้ integrate CI นั้นกับ CIs อื่นๆ และดูว่าทำงานได้เหมือนเดิมหรือไม่

บันทึกผลการดำเนินการลงในฐานข้อมูล เป็น change history ของ CIs นั้นๆ

7) การทำ baseline ตัวใหม่ ต่อ item ที่ได้รับการเปลี่ยนแปลงเรียบร้อยแล้ว

(Baseline change control)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Configuration Item ที่ผ่านการ approved และพร้อม release ไปยัง phrase ถัดไป หรือส่งมอบได้

2.7 Change Control Board

2.7.1 Change Control Board Team

ถ้าเป็นการทำ process ที่จะต้องมีการ review หรือ approve จะต้องมีการตั้งทีมๆ หนึ่ง ขึ้นมาเพื่อ control และ track change ที่เกิดขึ้นกับ project เรียกทีมนี้ว่า Change Control Board หรือ CCB โดยทั่วไป CCB ก็จะประกอบด้วยสมาชิกในทีมพัฒนาเอง อาจจะเป็น help desk, project manager และ SA ที่จะช่วยกัน filter, tracking และ control change ต่างๆ ซึ่งหน้าที่ของทีมนี้อาจมีดังนี้

1) รับ Change Request จาก user หรือลูกค้า ทำการ submit CR เข้าสู่ระบบเพื่อให้ SA หรือ Developer เข้ามา review ก่อนที่จะทำหรือ planning ต่อไป

2) Review CR ก่อนที่จะ implement CR นั้น CR ควรที่จะต้องถูก review ก่อน เช่น CR นั้นมีความสำคัญหรือไม่ CR นั้นจะเป็นต่อลูกค้าจริงหรือไม่ คำนวณค่าใช้จ่ายหรือไม่ ถ้าทำแล้วกระทบกับใครหรืองานจุดไหน ลูกค้า (คนจ่ายเงิน) รับรู้เรื่องนี้หรือไม่กรณีที่มีค่าใช้จ่ายเพิ่มหรืองานอาจจะ delay ถ้าต้องทำ เป็นต้น การ review ก็จะเป็นการป้องกันไม่ให้ทีมทำงานที่ไม่จำเป็นหรือไม่ควรที่จะต้องทำ และให้แน่ใจว่า software นั้นมี feature ที่มีประโยชน์จริงๆ เท่านั้น

3) Track change request status คอยดูแลสถานะของ CR ว่า CR เหล่านั้นถูก review หรือยัง แก้ไขหรือยัง เสร็จแล้วหรือไม่ ใครดูแลรับผิดชอบแต่ละ status เช่น ถ้าเป็น bug ใครเป็นคน fix, ใครจะเป็นคน test เมื่อ fix แล้ว เป็นต้น

4) Deployment plan กำหนดว่า CR นั้นจะถูก deploy ไปใน version ไหน โดยปกติ Process ของ change management และ CCB นั้นจะขึ้นอยู่กับ methodology ที่แต่ละองค์กรเลือกใช้ เช่นถ้าเป็น eXtreme Programming จะไม่ได้เน้นว่าจะต้องมี change process หรือ CCB เพราะ developer ทำงานแบบ pare developer ถ้าเกิด change ขึ้นมา ก็จะ log ลงไปใน user story ใครที่ดูแล user story นั้นก็ดูแลกันไป หรือในกรณีที่เป็นองค์กรใหญ่ๆ ที่มีแผนก Helpdesk แยกต่างหากไป วิธีการก็จะแตกต่างกันไป อาจจะต้องมีการทำ review สรุปค่าใช้จ่าย เป็นต้น

ตารางที่ 2.2 Change Request Form (CRF)

Project Name		Change Number	
Service			
Requested By		Date of Request	
Presented To		Detail Design Number	
Change Name			
Service /Module / Segment			
Priority			
<input type="checkbox"/> Critical <input checked="" type="checkbox"/> Important <input type="checkbox"/> Desirable			
Description of Change:			
Reason for Change:			
Effect on Deliverables (including a list of any affected deliverables):			
Effect on Organization:			
Effect on Schedule (including Estimated Completion Date for this change):			
Effect on Project Cost:			
Item Description			Man-day
Analysis			
Development (Code and Unit Test)			
Total:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Effect of NOT Approving this Change:
Reason for Rejection (if applicable):

Project Manager

Approved Signature: _____

Rejected Title: _____ Date: _____

User

Approved Signature: _____

Rejected Title: _____ Date: _____



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การทดสอบระบบ (Testing)



รูปที่ 2.10 System Development Life Cycle

เนื่องจากการสร้างระบบในแต่ละครั้งอาจเกิดข้อผิดพลาดต่างๆได้มากมาย จึงจำเป็นต้องมีผู้ทดสอบระบบ เพื่อให้ระบบมีความพร้อมในการใช้งาน และเกิดความสมบูรณ์ถูกต้องที่สุดก่อนที่จะส่งมอบให้กับผู้ใช้งานระบบ (User) โดยจะอยู่ในขั้นตอน Testing ในกระบวนการของ System Development Life Cycle (SDLC) ซึ่งการทดสอบระบบในแต่ละครั้งจะมีขั้นตอนมาตรฐานในการทดสอบ และควรทดสอบโดยผู้ทดสอบหลายคน เพื่อไม่ให้เกิดการสับสน และให้ครอบคลุมทุกๆกรณี โดยจะทดสอบจากเอกสาร Software Requirement หรือเรียกอีกอย่างว่า Program Specification หรือ Detail Design ที่ถูกเขียนขึ้นโดย System Analysis (SA) ซึ่งในเอกสาร Program Specification จะเป็นรายละเอียดเกี่ยวกับความต้องการของผู้ใช้งานระบบ (User) โดยจะใช้ภาษาในการเขียนที่เข้าใจง่าย เรียกว่า ภาษาในระดับ High Level โดยการทดสอบระบบนั้นสามารถแบ่งได้เป็น 2 มุมมอง คือ การทดสอบระบบในมุมมองของ Programmer และการทดสอบระบบในมุมมองของ Tester ซึ่งความแตกต่างสามารถแบ่งได้ตามตารางด้านล่าง

ตารางที่ 2.3 แนวความคิดในการทดสอบระบบระหว่าง Programmer และ Tester

Programmer	Tester
ทดสอบเฉพาะในส่วนของการเขียนโปรแกรม จะไม่ทดสอบในเชิง Business	ทดสอบในเชิง Business ซึ่งมีการรายละเอียดต่างๆเข้ามาเกี่ยวข้อง
ทดสอบเฉพาะชุดคำสั่ง (Code) ที่มีการเปลี่ยนแปลง	ทดสอบการทำงานของระบบอย่างเป็นขั้นตอน และทดสอบในส่วนที่จะเกิดข้อผิดพลาด
ถ้ามีการแก้ไขเล็กน้อย จะไม่สนใจทำการทดสอบ และจะไม่ตรวจสอบว่ามีผลกระทบต่อส่วนอื่นหรือไม่	ทดสอบทั้งในส่วนที่มีการเปลี่ยนแปลง และ ไม่มีการเปลี่ยนแปลง
สร้างข้อมูลเพื่อใช้ในการทดสอบ โดยที่ไม่ต้องเอามาจากระบบอื่นๆ	ทดสอบระบบในทุกๆด้านไม่ว่าจะเป็นเชิงบวก หรือเชิงลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 ความแตกต่างระหว่าง Verification และ Validation

- 1) Verification คือ กระบวนการประเมิน ตรวจสอบ และควบคุมให้ Software ที่ถูกออกแบบ และพัฒนานั้นถูกต้องตรงตาม Requirement ของ User ซึ่งขั้นตอนนี้ต้องทำการตรวจสอบก่อนการพัฒนา Software
- 2) Validation คือ กระบวนการตรวจสอบ Software ว่าถูกพัฒนาได้ตรงตาม Requirement ของ User หรือไม่ ซึ่งขั้นตอนนี้จะทำการตรวจสอบหลังจากได้พัฒนา Software ไปแล้ว

2.8.2 เทคนิคที่ใช้ในการทดสอบระบบ

- 1) Black Box คือ การทดสอบโดยผู้ทดสอบระบบ (Tester) โดยที่จะสนใจเพียงว่า ระบบมี Function การทำงานอย่างไร และผลลัพธ์ที่เกิดขึ้น ไม่สนใจชุดคำสั่ง (Code) ต่างๆในระบบ
- 2) White Box คือ การทดสอบที่ผู้ทดสอบจำเป็นต้องมีความรู้เรื่อง Programming ซึ่งจะทดสอบการทำงาน และการออกแบบระบบต่างๆ

2.8.3 ประเภทของการทดสอบโปรแกรม

- 1) Functional Test คือ จะใช้เทคนิคการทดสอบแบบ Black Box ซึ่งจะทำการทดสอบโดยผู้ทดสอบระบบ (Tester) โดยจะทดสอบว่าระบบนั้นๆสามารถทำงานอะไรได้บ้าง
- 2) Non – Functional Test คือ เป็นการทดสอบประสิทธิภาพ ความถูกต้อง ความสามารถ และความน่าเชื่อถือของระบบ ทั้งทางด้าน Software และ Hardware ซึ่งตัวอย่างของ Non – Function เช่น Performance Test (ทดสอบประสิทธิภาพของระบบ) Load Test (ทดสอบการรองรับปริมาณผู้ใช้งาน) เป็นต้น

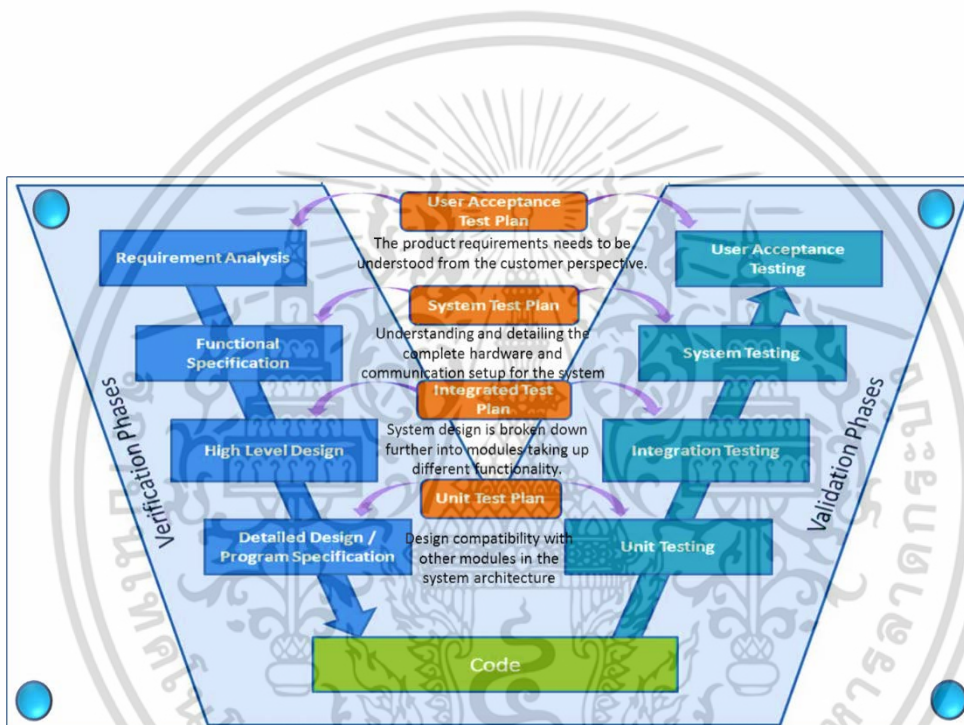
2.8.4 ระดับ Software Testing มี 4 ระดับ

- 1) Unit Test: ทดสอบโดย Developer ซึ่งเป็นการทดสอบเบื้องต้น ขณะที่เขียนโปรแกรมในส่วนของตนเอง เช่น การทดสอบไวยากรณ์ (Syntax) หรือ Format ต่างๆ เป็นต้น เพื่อตรวจสอบผลการทำงานของแต่ละ Module ย่อยๆทั้งหมดของระบบให้ตรงตามที่ออกแบบไว้ ซึ่งเทคนิคที่ใช้ในการทดสอบ คือ White Box และ Black Box
- 2) Integration Test: ทดสอบส่วนของโปรแกรมโดยมี Function หรือ Feature มากกว่า 2 ขึ้นไป แต่โปรแกรมยังไม่ใหญ่มาก ซึ่งเป็นการทดสอบการส่งค่าตัวแปรต่างๆไปยังแต่ละหน้าว่าทำงานถูกต้องหรือไม่
- 3) System Test: เป็นการทดสอบระบบที่ใกล้เคียงกับระบบจริงมากที่สุด ซึ่งจะมีการเรียกใช้งานค่าต่างๆมาจากหลากหลายระบบ เพื่อตรวจสอบ (Verify) การทำงานของระบบว่าทำงานถูกต้องตาม Requirement หรือไม่ โดยสามารถทดสอบได้ทั้งแบบ Functional และ Non-Functional ขึ้นอยู่กับว่าระบบเหมาะสมกับแบบใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) E2E (End-to-End): เป็นการทดสอบก่อนส่งมอบให้ผู้ใช้งานระบบ (User) เรียกอีกอย่างว่า Acceptance Testing

Regression Test: จะเป็นการทดสอบเพื่อดูผลจากการเปลี่ยนแปลงในบางส่วน ซึ่งอาจจะไปกระทบกับส่วนอื่นๆที่ไม่ได้ทำการเปลี่ยนแปลง โดย Regression Test นี้ เปรียบเสมือนการทดสอบอีกครั้งใน Function เดิมที่ได้ทำการทดสอบไปแล้ว เพื่อตรวจสอบว่า Function นั้นๆ ยังสามารถใช้งานได้เช่นเดิม ซึ่งการทำ Regression Test จะเกิดขึ้นก็ต่อเมื่อมี Requirement แจ้งมาว่าให้ทดสอบใหม่ทั้งระบบ



รูปที่ 2.11 V Model

2.8.5 รูปแบบการทดสอบระบบ

- 1) Manual Test: เป็นการทดสอบระบบทีละขั้นตอน เมื่อทดสอบแล้วต้องนำมาเขียนเป็น Test Case เพื่อให้ผู้ทีนำไปอ่านต่อเข้าใจได้ง่าย
- 2) Automated Test: เป็นการทดสอบระบบโดยการรัน Script หรือทดสอบซ้ำๆ เพื่อให้ได้ผลลัพธ์เหมือนเดิม ซึ่งถ้ามีการเปลี่ยนแปลงส่วนใดส่วนหนึ่งของโปรแกรมต้องทำการทดสอบใหม่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.6 ขั้นตอนการจัดการความผิดพลาด (Life Cycle Defect Management)

Defect คือข้อผิดพลาดที่เกิดขึ้นกับระบบ ซึ่งไม่ตรงกับ Program Specification สามารถเรียกได้อีก อย่าง คือ Bug หรือ Issue

- 1) ทดสอบระบบตามกรณีที่มีอยู่ ถ้าเจอ Defect ต้องตอบกลับให้ผู้เขียน โปรแกรมที่รับผิดชอบให้รับรู้ โดยทาง Developer Leader จะทำการ Assign Defect ต่างๆ ให้กับ Programmer เพื่อให้ทำการแก้ไข ซึ่งใน ส่วนนี้จะมี Status เป็น Assign
- 2) เมื่อ Programmer แต่ละคนทำการแก้ไข Defect ของตนเองตามที่ได้รับ มอบหมายเรียบร้อยแล้ว ในส่วนนี้จะมี Status เป็น Resolve
- 3) ในกรณีที่ไมแก้ไข Defect ต่างๆ ที่พบ เนื่องจากสาเหตุใดสาเหตุหนึ่ง ในส่วนนี้จะมี Status เป็น Feedback
- 4) ถ้าการแก้ไขโปรแกรมต้องรอข้อสรุป หรือวิธีการแก้ไขจากผู้ที่เกี่ยวข้อง ในส่วนนี้จะมี Status เป็น Acknowledged
- 5) Defect ต่างๆ ได้รับการแก้ไขอย่างถูกต้องสมบูรณ์ ซึ่งในส่วนนี้จะมี Status เป็น Closed

2.8.7 ความสำคัญของข้อผิดพลาด (Defect Priority)

- 1) Critical: เป็นข้อผิดพลาดที่กระทบต่อระบบมากที่สุด ต้องได้รับการแก้ไข ทันที ไม่เช่นนั้นระบบจะทำงานต่อไม่ได้
- 2) High/ Major: เป็นข้อผิดพลาดที่กระทบต่อระบบมาก แต่ไม่จำเป็นต้อง แก้ไขทันที โดยอาจเกิดจากการกรอกข้อมูลที่ต้องการ แต่ระบบแสดงผลการทำงาน ผิดพลาด เช่น การเกิด Error
- 3) Minor: เป็นข้อผิดพลาดที่กระทบต่อระบบ แต่ระบบยังสามารถ ทำงานต่อได้ เช่น การส่งค่าผิด
- 4) Low: เป็นข้อผิดพลาดที่กระทบต่อระบบน้อยที่สุด เช่น Design ผิดพลาดสีของหน้า Web Page เป็นต้น

2.8.8 ระดับในการทดสอบระบบ

- การทดสอบบนเครื่องผู้พัฒนา หรือเรียกอีกชื่อว่า Dev คือ การทดสอบโดยใช้เครื่องทดสอบที่มีประสิทธิภาพน้อยที่สุด เช่น Ram น้อย เป็นต้น ถ้าทดสอบผ่านจะไปทดสอบในระดับถัดไป
- การทดสอบในระดับ Staging หรือเรียกอีกชื่อว่า UAT คือ การทดสอบโดยให้ผู้ใช้งานระบบ (User)เป็นผู้ทดสอบ ผู้พัฒนาโปรแกรมไม่สามารถทดสอบได้ เครื่องที่ใช้ในการทดสอบมีประสิทธิภาพเพิ่มมากขึ้น ถ้าทดสอบผ่านจะไปทดสอบในระดับถัดไป
- การทดสอบในระดับ Production หรือเรียกอีกชื่อว่า Prod คือ การทดสอบในระดับสุดท้าย ซึ่งเครื่องที่ใช้ในการทดสอบจะมีประสิทธิภาพมากที่สุด เช่น Ram เยอะที่สุด เป็นต้น

หมายเหตุ : การทดสอบในแต่ละระดับต้องมีจำนวนเครื่องในการทดสอบที่เท่ากัน

2.9 Defect Management

Defect Management คือ การจัดการ Defect ที่พบในช่วงของ Phase Testing โดย Tool ที่ผู้เขียนใช้คือ Mantis การนำ Tool เข้ามาใช้ใน Project นั้นๆ เพื่อใช้ Report Defect ที่พบเพื่อแจ้งทีมที่เกี่ยวข้อง และเพื่อใช้ข้อมูลสรุปปัญหาของโปรเจกต์นั้นๆ ว่าจะสามารถ Launch ได้ตาม Plan หรือไม่

2.9.1 Defect

Defect คือ ปัญหาที่พบในการทดสอบระบบ ซึ่งปัญหาเหล่านี้อาจกระทบต่อ Function การทำงานของระบบ เช่น ระบบแสดง Error ต่างๆ หรือ Defect ที่พบอาจจะไม่กระทบกับ Function การทำงาน เช่น การแสดงผลที่อาจเกิดจาก Design หรือ การแสดงผลของข้อความ ซึ่งปัญหาเหล่านี้จะไม่ส่งผลกระทบต่อระบบ

2.9.2 Defect Life Cycle

คือ วงจรการทำงานของ Defect ที่เริ่มตั้งแต่เมื่อพบ Defect แล้ว Assign ให้ทางทีม Developer แก้ไข จนกระทั่ง Developer แก้ไขเสร็จ ซึ่งมีรายละเอียด ดังนี้

2.9.3 Defect Status

เป็นการบอกถึงสถานะของ Defect ที่พบ Status ที่ใช้ปัจจุบันจะมีอยู่ 5 status ดังนี้

- Assign: เมื่อพบ defect จะ Assign ไปให้ทาง Team Develop ที่รับผิดชอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Resolve: เมื่อ Developer แก้ไข Defect ที่พบ และ upload Code ขึ้น Test Environment
- Feedback: ไม่แก้ไข Defect ที่พบเนื่องจากสาเหตุใดๆ
- Acknowledged: การแก้ไขโปรเจคค ต้องรอข้อสรุป หรือ solution จากผู้ที่เกี่ยวข้องกับโปรเจค
- Closed: Defect ได้รับการแก้ไขถูกต้อง

2.9.4 Defect Type

เป็นการแบ่งประเภทของ Defect ที่พบ ว่าเกิดจากส่วนใดในการทำงาน สามารถแบ่งออกได้เป็น 6 ประเภท

- Requirement: เป็น Defect ที่เกิดจากการแก้ไข Requirement ของทาง Business โดยไม่แจ้งทีมที่เกี่ยวข้อง (Design, Developer, Tester)
- Coding: เป็น Defect ที่เกิดจากการ Coding ของทาง Developer ที่ไม่ตรวจสอบในส่วนนั้นๆ
- Graphic Design: เป็น Defect ที่เกิดจากการ Design ที่ไม่รองรับกับ Browser ต่างๆ หรือ เมื่อ developer นำ Design มาประกอบกับ Code แล้วทำให้การแสดงผลไม่ถูกต้อง
- Tester: เป็น Defect ที่เกิดจากความเข้าใจผิด หรือเกิดจากความผิดพลาดของ Tester
- Data Test: เป็น Defect ที่เกิดจาก Test data อาจไม่มีใน Environment หรือระบบอาจไม่ Support data ส่วนนี้
- Other: ข้อจำกัดของระบบ, ข้อจำกัดของ Environment

2.9.5 Defect Severity

เป็นการระบุความรุนแรงของ Defect ที่พบใน Test Phase โดยยึดตาม Standard Severity Code แบ่งออกเป็น 4 ระดับ ดังนี้

- Critical: Defect ที่ไม่สามารถทดสอบโปรแกรมในส่วนของ Function นั้นต่อได้
- High: Defect ที่เกิดจากการใส่ข้อมูลถูกต้อง แต่ระบบแสดงผลผิดพลาด เช่น Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Medium: ระบบจะแสดงผลถูกต้องเมื่อใส่ข้อมูลถูกต้อง แต่เมื่อใส่ข้อมูลไม่ถูกต้อง ระบบจะแสดงผลผิดพลาด เช่น Filed ที่มีการ Validate ผล เมื่อใส่ ค่าว่าง, อักขระพิเศษ (‘, ” %, &) และ Script ที่มีผลต่อการแสดงผลของระบบ ๆ จะแสดงผลผิดพลาด
- Low: Defect ที่เกิดจากการแสดงผลของข้อความ หรือ เรื่องของการ Design ซึ่ง Defect เหล่านี้จะไม่มีผลกระทบต่อการทำงานของระบบ

2.9.6 Defect Tracking

สำหรับ Tool ที่ใช้เป็นตัวช่วยในการจัดการกับ Defect ที่พบ คือ Jira ใช้ในการ log Defect ที่พบ เพื่อแจ้งทีมที่เกี่ยวข้องให้เข้าไปดูรายละเอียดของ Defect และแก้ไขในส่วนที่รับผิดชอบ ข้อมูล Defect เหล่านี้จะใช้ในการสรุปปัญหาของระบบว่าจะสามารถ Launch ได้ตาม Plan หรือไม่

2.10 Jira

Jira คือโปรแกรมที่ช่วยจัดการงานต่างๆ ใน Project ใน Software Development project ส่วนใหญ่ก็จะประกอบไปด้วยงานหลายๆ งาน งานพัฒนาฟังก์ชันงานต่างๆ หรืองานทดสอบฟังก์ชันงานที่ได้พัฒนาขึ้น ซึ่ง Jira ช่วยในการติดตามงานต่างๆ และยังสามารถแบ่งงานออกเป็นเฟสต่างๆ หรือที่เรียกว่าเป็นแต่ละรีลีส (release) นอกจากนี้ยังสามารถสร้าง release notes ที่จะใช้เป็นเอกสารร่วมกับการ release



รูปที่ 2.12 Jira

2.10.1 ประเภทของงานใน Jira

งานในแต่ละงานใน Jira แบ่งเป็นแต่ละประเภท ตัวอย่างเช่น

- Bug ปัญหาการทำงานของโปรแกรม
- Improvement งานที่จะต้องพัฒนาให้ดีขึ้น
- New feature ความสามารถในการทำงานต้องการจะเพิ่มเข้ามาในระบบ
- Task งานทั่วไปที่จะต้องทำให้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในการใช้งาน Jira สามารถปรับแต่งประเภทของงานได้ตามที่ต้องการ ไม่จำเป็นต้องใช้ตามนี้ ตัวอย่างการปรับแต่งประเภทของงานนะค่ะ

2.10.1.1 Project management type

- Task หมายถึงงานทั่วไป หรือปัญหาที่ต้องได้รับการแก้ไขในProject
- Risk งานที่ทำเพื่อป้องกันปัญหาที่จะเกิดขึ้นในอนาคต
- New Feature หมายถึง ความสามารถในการทำงานที่ต้องการให้มีการเปลี่ยนแปลง

2.10.1.2 Analysis and Design type

- Story งานที่เกี่ยวข้องกับการวิเคราะห์ระบบ เช่นการทำ Use Case, Activity Diagram
- Model งานที่เกี่ยวข้องกับการออกแบบระบบ เช่น การทำ Class Diagram, Database Diagram

2.10.1.3 Development type

- Code หมายถึงงานพัฒนาโปรแกรม

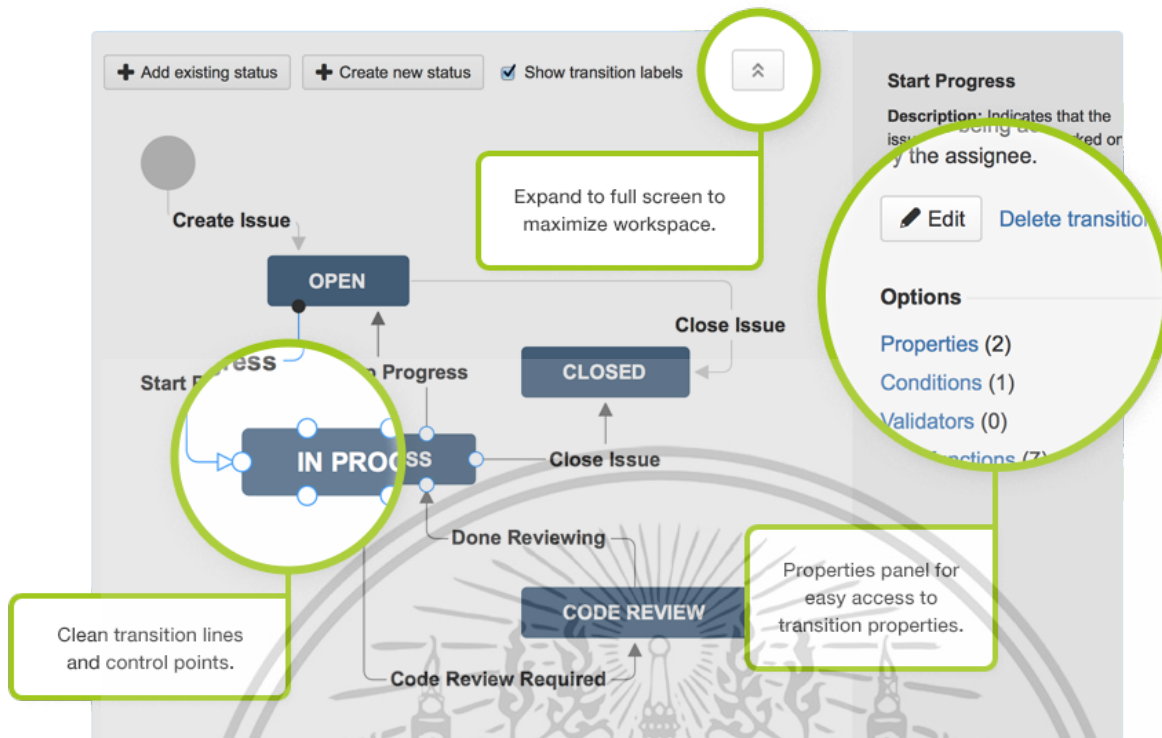
2.10.1.4 Testing type

- Test Event หมายถึงการทำ Test Script
- Test Execution หมายถึงการตรวจสอบการทำงานของโปรแกรม โดย Test Script ที่ได้ทำจาก Test Event
- Bug ในกรณีที่ตรวจสอบแล้วพบข้อบกพร่องในการทำงาน ก็จะต้องสร้างเป็น Defect เพื่อส่งกลับไปให้นักพัฒนาระบบแก้ไข

2.10.1.5 Deployment type

- Release Task เมื่อทดสอบและแก้ไขแล้ว ก็จะต้อง release หรือติดตั้งโปรแกรมบน server ซึ่งอาจจะเป็น development server หรือ production server ก็แล้วแต่ เรียกว่าเป็น Release Task

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Jira Screen

2.10.2 ความสำคัญ (Priority)

- Blocker เป็นงานที่ถ้าไม่ทำให้เสร็จ ก็จะกันไม่ให้งานอื่นๆ สำเร็จได้ คือเป็นงานที่ block ชาวบ้านนั่นเอง
- Critical เป็นงานที่มีความสำคัญมาก ซึ่งทำให้การทำงานในฟังก์ชันใดๆ ไม่สามารถทำงานได้
- Major เป็นงานที่มีผลกระทบกับการทำงานโดยรวม
- Minor เป็นงานที่กระทบกับการทำงานแต่น้อย ซึ่งโปรแกรมยังสามารถทำงานต่อ
- Trivial เช่นการแก้ไขคำผิด หรืองานใดๆ ที่ไม่มีผลต่อการทำงานของฟังก์ชัน

งานแต่ละงานใน Jira จะเก็บรายละเอียดดังกล่าวข้างต้นแล้วยังเก็บ หัวข้อ, รายละเอียดของงาน, ความคิดเห็นที่โต้ตอบไปมาเกี่ยวกับงานนั้นๆ และยังสามารถ attach file รายละเอียดหรือข้อมูลเพิ่มเติมในงานแต่ละงานได้ และเก็บเวลาที่ใช้ในการทำงานของแต่ละในงานหนึ่งๆ ได้ เพื่อใช้ในการประเมินงาน และเป็นสถิติของโปรเจคต่อไป

Jira ยังมีความสามารถอื่นๆ ที่ช่วยในการทำงานของโปรเจคได้ ดังจะเห็นว่าหลายๆ โปรเจค รวมทั้ง open source project ก็ยังนำโปรแกรมตัวนี้มาช่วยในการทำงาน เนื่องจากเป็น web application ทำให้ user ไม่ต้องลำบากในการลงโปรแกรม และยังเข้าถึงได้ง่าย

2.11 ระบบบริหารความสัมพันธ์ลูกค้า (Customer Relationship Management หรือ CRM)

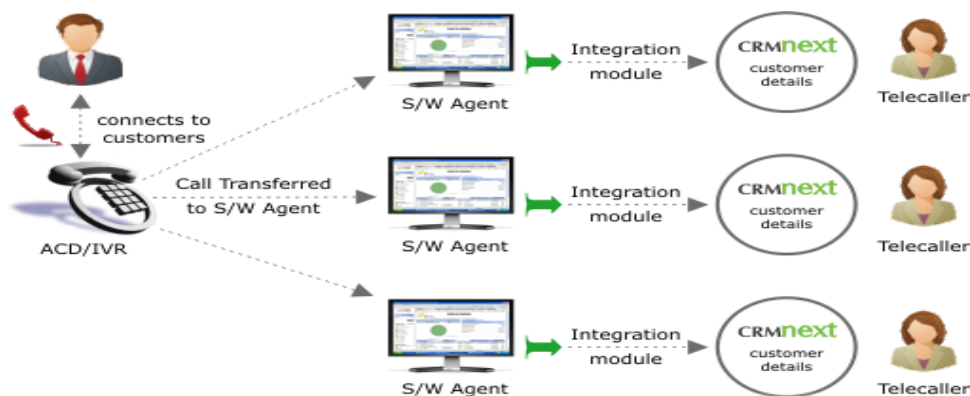
CRM ย่อมาจาก Customer Relationship Management คือ การบริหารความสัมพันธ์กับลูกค้า ซึ่งให้ลูกค้ามีความรู้สึกผูกพันกับสินค้า บริการ หรือองค์กร เมื่อลูกค้ามีความผูกพันในทางที่ดี แล้ว ลูกค้าก็จะไม่เปลี่ยนใจไปจากสินค้า หรือบริการ ทำให้มีฐานลูกค้าที่มั่นคง และนำมาซึ่งความมั่นคงขององค์กร ดังนั้น การที่จะเข้าใจลูกค้าได้นั้น จะต้องอาศัยการสังเกตพฤติกรรมของลูกค้า แล้วนำมาวิเคราะห์หาความเกี่ยวข้องระหว่างพฤติกรรมของลูกค้า กับกลยุทธ์ทางการตลาดขององค์กร ซึ่งกระบวนการทำงานของระบบ CRM มี 4 ขั้นตอนดังนี้

- 1) Identify คือ การเก็บข้อมูลว่าลูกค้าของบริษัทเป็นใคร เช่น ชื่อลูกค้า ข้อมูลสำหรับติดต่อกับลูกค้า เป็นต้น
- 2) Differentiate คือ การวิเคราะห์พฤติกรรมของลูกค้าแต่ละคน และจัดแบ่งลูกค้าออกเป็นกลุ่มตามประเภทลูกค้า
- 3) Interact คือ การมีปฏิสัมพันธ์กับลูกค้าเพื่อเรียนรู้ความต้องการของลูกค้า และเพื่อสร้างความพึงพอใจให้กับลูกค้าในระยะยาว
- 4) Customize คือ การนำเสนอสินค้า หรือบริการที่มีความเหมาะสมเฉพาะตัวกับลูกค้าแต่ละคน

นอกจากนั้นแล้ว CRM ยังมีความสัมพันธ์เกี่ยวกับเทคโนโลยีสารสนเทศในด้านการนำซอฟต์แวร์มาใช้ในการพัฒนาเพื่อดึงดูดลูกค้า สร้างความสัมพันธ์กับลูกค้า เช่น Website เป็นต้น ซึ่ง CRM software จะช่วยในการจัดการฐานข้อมูลลูกค้า วิเคราะห์ข้อมูล และเป็นช่องทางในการติดต่อกับลูกค้า โดย CRM software มักแบ่งออกเป็น 3 ส่วนดังนี้

- 1) Operational CRM เป็นซอฟต์แวร์ Front Office ที่ใช้จัดการกระบวนการทางธุรกิจที่เกี่ยวข้องกับลูกค้า ไม่ว่าจะเป็น Sales, Marketing หรือ Service เช่น การจัดการข้อมูลติดต่อลูกค้า การเสนอราคา การบริหารฝ่ายขาย การเก็บข้อมูลพฤติกรรมลูกค้า ระบบบริการลูกค้า เป็นต้น
- 2) Analytical CRM ระบบวิเคราะห์ข้อมูลลูกค้าที่รวบรวมได้จากส่วน Operational CRM หรือจากแหล่งอื่นๆ เพื่อแบ่งลูกค้าออกเป็นกลุ่ม และค้นหากลุ่มลูกค้าเป้าหมายที่บริษัทสามารถนำเสนอสินค้า หรือบริการเพิ่มเติมได้
- 3) Collaborative CRM ระบบช่วยสนับสนุนในการติดต่อปฏิสัมพันธ์กับลูกค้าผ่านทางช่องทางต่างๆ เช่น ติดต่อส่วนตัว จดหมาย แฟกซ์ โทรศัพท์ เว็บไซต์ E-Mail เป็นต้น รวมถึงช่วยจัดการทรัพยากรที่บริษัทมี คือพนักงาน กระบวนการทำงาน และฐานข้อมูล (Database) เพื่อนำไปให้บริการแก่ลูกค้า และช่วยรักษาฐานลูกค้าของบริษัทได้ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ระบบบริหารความสัมพันธ์ลูกค้า (CRM Software)

2.11.1 ประโยชน์ของ CRM ต่อธุรกิจ

- 1) CRM ช่วยเพิ่มความสามารถในการให้บริการลูกค้าได้ดีขึ้น เช่น ใช้ Website ในการให้ข้อมูลเกี่ยวกับสินค้า วิธีการใช้สินค้า และให้ความช่วยเหลือแก่ลูกค้าได้ตลอด 24 ชั่วโมง เป็นช่องทางให้ลูกค้าแนะนำติชมต่อบริการของบริษัทได้ง่าย ช่วยให้ลูกค้าสามารถ customize ความต้องการของตนเองได้ทันที เป็นต้น
- 2) CRM ช่วยพัฒนาความสัมพันธ์ระหว่างบริษัทกับลูกค้าให้ดีขึ้น ช่วยให้บริษัทรู้ความสนใจ ความต้องการ และพฤติกรรม การซื้อของลูกค้า ทำให้บริษัทสามารถนำเสนอสินค้าที่เหมาะสม กับลูกค้าได้ และช่วยให้บริษัทสามารถให้บริการหลังการขายแก่ลูกค้าตามที่ลูกค้าต้องการได้ การสร้างความสัมพันธ์กับลูกค้าในระยะยาวจะช่วยเพิ่มความจงรักภักดี (loyalty) ที่ลูกค้ามีต่อบริษัท ลดการสูญเสียลูกค้า ลดต้นทุนการตลาด เพิ่มรายได้จากการที่ลูกค้าซื้อซ้ำหรือแนะนำให้คนรู้จักซื้อสินค้าของบริษัท และหมายถึงกำไรของบริษัทที่เพิ่มขึ้น

2.11.2 การเพิ่มประสิทธิภาพของเจ้าหน้าที่ Call Center ด้วยระบบ CRM

ระบบ CRM (Customer Relationship Management Software) สามารถเพิ่มประสิทธิภาพของเจ้าหน้าที่ที่ทำงานในศูนย์บริการลูกค้า (Call Center) ได้ เนื่องจากเป็นระบบที่รวบรวมข้อมูลของลูกค้า และกิจกรรมต่างๆ ที่ลูกค้าได้ติดต่อมายัง Call Center เพื่อให้เจ้าหน้าที่ได้ค้นหาข้อมูลเพิ่มเติม โดยไม่ต้องขอข้อมูลซ้ำจากลูกค้า หากองค์กรมีระบบ CRM ซึ่งได้การออกแบบให้ทำงานเข้ากับระบบ call center แล้ว ก็จะสามารถเพิ่มอัตราการแก้ปัญหาในสายแรก (Increase First Call Resolution Rate) ลดระยะเวลาในการคุยสายกับลูกค้า (Reduce Call Time) โดยระบบ CRM จะมีระบบปฏิบัติการเชื่อมต่อระหว่าง call center และระบบของหน่วยงานอื่นๆ สามารถค้นหาข้อมูลแบบ real-time ทำให้เกิดความรวดเร็วในการแก้ปัญหา และพัฒนาความพึงพอใจแก่ลูกค้าได้ เพิ่มโอกาสในการขายได้มากขึ้น (Improve Up Sell and Cross Sell) และสามารถพัฒนาระบบการติดต่อของเจ้าหน้าที่ในองค์กรให้มีประสิทธิภาพดียิ่งขึ้น (Improve Call Center Workflows) ปัญหาที่มักจะพบในการทำงานของเจ้าหน้าที่เกี่ยวกับการทำงานบนจอแสดงผล (Monitor) ในกรณีที่เจ้าหน้าที่จำเป็นต้องเปิดหน้าจอแสดงผลหลายๆจอ พร้อมกัน เพื่อไว้เพื่อใช้ในการแก้ปัญหาให้กับลูกค้า ป้อนข้อมูลการให้บริการพร้อมทั้งต้องส่งข้อมูลออกไปยังพนักงานอื่นเพื่อหาข้อมูล ตรวจสอบสถานะและจัดการข้อมูลพื้นฐานอื่นๆด้วยหน้าจอแสดงผลของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ CRM นั้นจะมีความยืดหยุ่นสูง โดยสามารถออกแบบเครื่องมือระบบจอแสดงผลที่จะดึงเอาข้อมูลจากฐานข้อมูลลูกค้า ใน Application ของ CRM และข้อมูลจาก Application ต่างๆใน call center มาแสดงบนจอแสดงผลที่มีการจัดการไว้ได้ ซึ่งจะช่วยให้เจ้าหน้าที่ เข้าสู่ขั้นตอนการทำงานในส่วนของแต่ละคนได้ในเวลาน้อยที่สุด ทั้งหมดนี้จะช่วยให้เจ้าหน้าที่ทำงานได้รวดเร็วขึ้น พัฒนาการบริการลูกค้าและพนักงาน call center ยังลดความผิดพลาดจากการทำงานได้อีกด้วย

2.12 ฐานข้อมูล (Database)

Database หรือ ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์กัน จำเป็นว่าข้อมูลทั้งหมดจะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกัน หรือแยกเก็บหลายๆแฟ้มข้อมูล

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่างๆที่เกี่ยวข้องกันไว้ด้วยกันโดยมีความสัมพันธ์ระหว่างข้อมูลต่างๆชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้ม และให้ผู้ใช้สามารถใช้งาน หรือจัดการกับข้อมูลได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้ และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวก และมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

2.12.1 ประโยชน์ของฐานข้อมูล

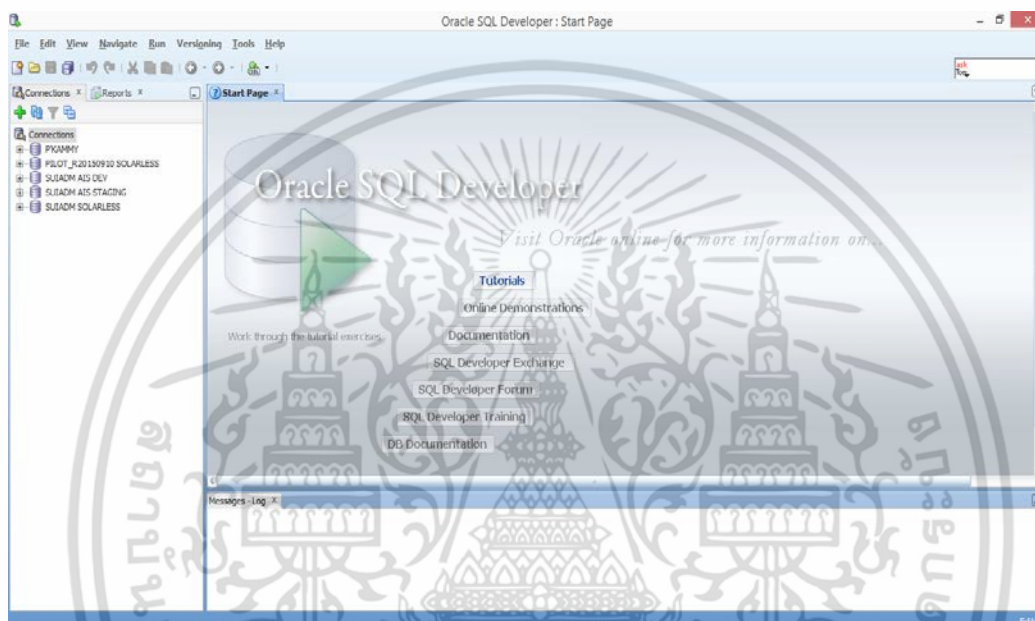
- 1) ลดการเก็บข้อมูลที่ซ้ำซ้อน เนื่องจากข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลายๆที่ เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ความซ้ำซ้อนของข้อมูลลดน้อยลง
- 2) รักษาความถูกต้องของข้อมูล เนื่องจากฐานข้อมูลมีเพียงฐานข้อมูลเดียว ในกรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายที่ในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุกที่ที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้องเหมือนกันโดยอัตโนมัติด้วยระบบจัดการฐานข้อมูล
- 3) การป้องกัน และรักษาความปลอดภัยให้กับข้อมูลทำได้ง่ายสะดวก การป้องกัน และรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้น ซึ่งก่อให้เกิดความปลอดภัย (security) ของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 เครื่องมือ หรือโปรแกรมที่ใช้งาน (Tools)

2.13.1 โปรแกรม Oracle SQL Developer

Oracle SQL Developer เป็นโปรแกรมฟรีแวร์จาก Oracle ใช้สำหรับจัดการฐานข้อมูล Oracle โดยเฉพาะ เช่นเดียวกับ MySQL Workbench ของ MySQL สำหรับผู้ที่ต้องการใช้ Oracle SQL Developer จำเป็นต้องติดตั้ง Java Developer Kit เป็นอันดับแรก จึงจะสามารถใช้งานได้ โปรแกรม Oracle SQL Developer (สามารถดูการใช้งานได้ในส่วนของภาคผนวก ค.)

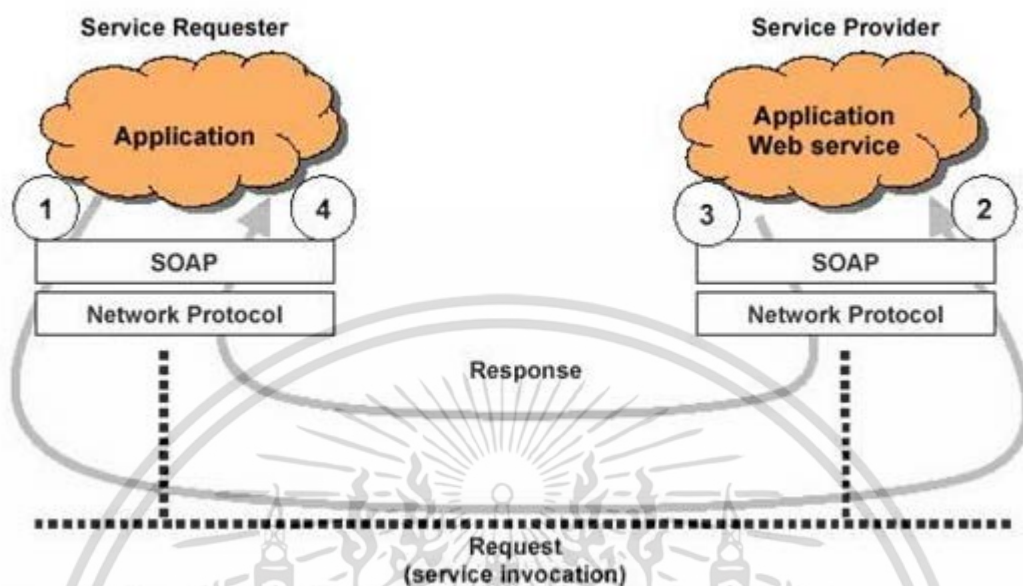


รูปที่ 2.15 ตัวอย่างโปรแกรม Oracle sqlDeveloper

2.13.2 SOAP (Simple Object Access Protocol)

SOAP (Simple Object Access Protocol) เป็น XML-based โปรโตคอล (lightweight protocol) และใช้ HTTP เป็นโปรโตคอลร่วม สำหรับการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้ กำหนดเมสเซจจิงโปรโตคอล (Messaging Protocol) ระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) เช่น ผู้ขอบริการสามารถติดต่อแลกเปลี่ยนข้อมูลกับผู้ให้บริการโดยใช้ RMI (Remote Method Invocation) ตามวิธีการของ โปรแกรมแบบออปเจ็ค บริษัทไมโครซอฟท์, ไอบีเอ็ม, ล็อตัส, ยูสเซอร์แลนด์ (UserLand) และ ดีเวลลอปเปอร์เมนเตอร์ (DeveloperMenter) ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัทเข้าร่วมและ จัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น SOAP ได้กำหนดรูปแบบพื้นฐานของการสื่อสารแบบกระจายขึ้นโดย การพัฒนา SOA แม้ว่า SOA จะไม่ได้กำหนดเมสเซจจิงโปรโตคอล (Messaging Protocol) ไว้ แต่ SOAP ได้ถูกกำหนด ให้เป็น Services-Oriented Architecture Protocol เรียบร้อยแล้ว เนื่องจากมันได้ถูกใช้ในการพัฒนา SOA อย่างแพร่ หลายแล้วนั่นเอง จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโปรโตคอล ที่ทำงานกับโปรโตคอลอื่นหลายชนิด การพัฒนาก่อนุญาตให้ทำได้อย่าง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ผ่านการอนุญาต ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิสระตามแพลตฟอร์มระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา



รูปที่ 2.16 Simple Object Access Protocol

2.13.2.1 จุดประสงค์ของ SOAP

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส ต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่ในเครื่องอื่นโดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่าง Object ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM, EJB หรือ CORBA นั้นไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอล HTTP (เครือข่ายที่ปัจจุบันใช้งานแพร่หลาย และสะดวกมากที่สุดได้แก่เครือข่าย Internet ซึ่งให้การสื่อสารผ่านโปรโตคอล HTTP เป็นส่วนใหญ่) เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้นต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย ยกเว้น CORBA ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy Server ด้วยเนื่องจากโดยปกติ เซอร์เวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอล HTTP ออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอกดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำบริการเว็บเซอร์วิส ก็คือให้ทำงานอยู่บน โปรโตคอล HTTP ไปเลย ซึ่ง SOAP นอกจากจะทำงานบนโปรโตคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.2.2 โครงสร้างของ SOAP

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

- 1) SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
- 2) SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้
- 3) SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส



รูปที่ 2.17 SOAP Envelope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเอกสาร SOAP

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <!-- ข้อมูลในส่วนของ Header -->
    <i:local xmlns:i="http://www.i3t.or.th/ws/">
      <i:currency>Bath</i:currency>
    </i:local>
  </soap:Header>
  <soap:Body>
    <!-- ข้อมูลในส่วนของ Body -->
    <GetPrice>
      <Item>Rose</Item>
      <Quantity>100</Quantity>
    </GetPrice>
  </soap:Body>
  <soap:Fault>
    <!-- ข้อมูลของ SOAP ในกรณีมีข้อผิดพลาด จาก SOAP Node -->
  </soap:Fault>
</soap:Envelope>

<soap:Fault>
  <faultcode>Client</faultcode>
  <faultstring>Invalid Request</faultstring>
</soap:Fault>

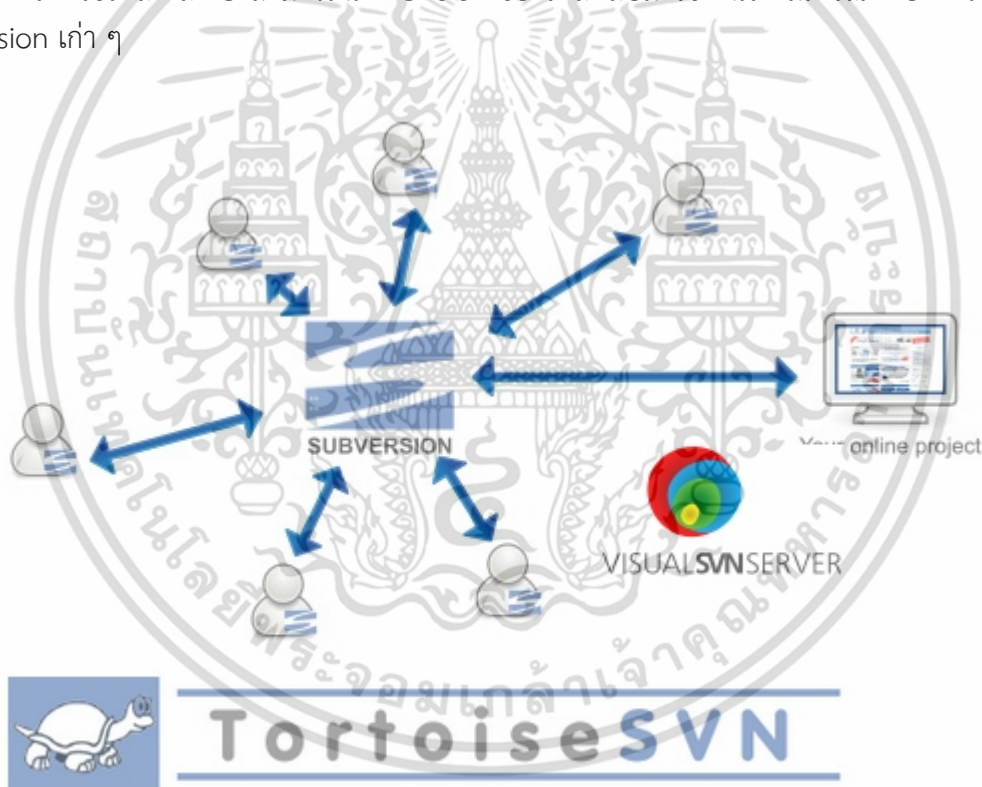
<soap:Fault>
  <faultcode>Client</faultcode>
  <faultstring>Invalid Request</faultstring>
</soap:Fault>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.3 SVN (Subversion)

SVN คือ โปรแกรมที่ทำหน้าที่จัดการกับ Version Control ของไฟล์ต่าง ๆ ไม่ว่าจะเป็นการจัดการกับเอกสารพวก Document , ไฟล์รูปภาพ และที่สำคัญที่สุดก็คือ จัดการเกี่ยวกับ Source Code ของโปรแกรม โดยหน้าที่ของ SVN คือจัดเก็บไฟล์ต่าง ๆ เหล่านั้นไว้ในคลัง แล้วแยกไฟล์นั้นเป็นเวอร์ชันต่าง ๆ โดยเรียกกันว่า Revision ซึ่งเวอร์ชันของไฟล์ที่จัดเก็บนั้นจะเกี่ยวข้องกับ User หลาย ๆ คนที่เข้ามาใช้งาน เช่น นาย A ทำเอกสารเสร็จ แล้วใช้ SVN จัดเก็บ จะถูกนับเป็น Revision : 1 และเมื่อนาย B นำไฟล์นี้ไปแก้ไขแล้วนำมาจัดเก็บลงใน SVN อีกครั้งก็จะถูกนับเป็น Revision: 2 และการแก้ไขต่อ ๆ ไปก็จะนับ Revision ไปเรื่อย ๆ ซึ่งประโยชน์ของมันก็คือโดยปกติแล้ว SVN จะมี Server ทำหน้าที่จัดเก็บไฟล์ ฉะนั้นไฟล์ต่าง ๆ จะถูกจัดเก็บไว้บน Server และเรียกใช้งานผ่าน Protocol : TCP/HTTP ฉะนั้น SVN Server ที่ทำหน้าที่จัดเก็บไฟล์ จะเป็นตัวกลางในการแลกเปลี่ยน Version ของไฟล์ ซึ่งวิธีนี้จะเป็นประโยชน์มากในการป้องกันไฟล์หาย และจะได้ไฟล์ล่าสุดเสมอ เมื่อทำการ Checkout หรืออัปเดตจาก SVN Server และประโยชน์อื่น ๆ ของ SVN เช่น ดู Log หรือ History ของการแก้ไขไฟล์ และ ยังสามารถนำ Revision ของไฟล์กลับมาใช้งานได้ ในกรณีที่ต้องการกลับไปใช้ Versionเก่า ๆ

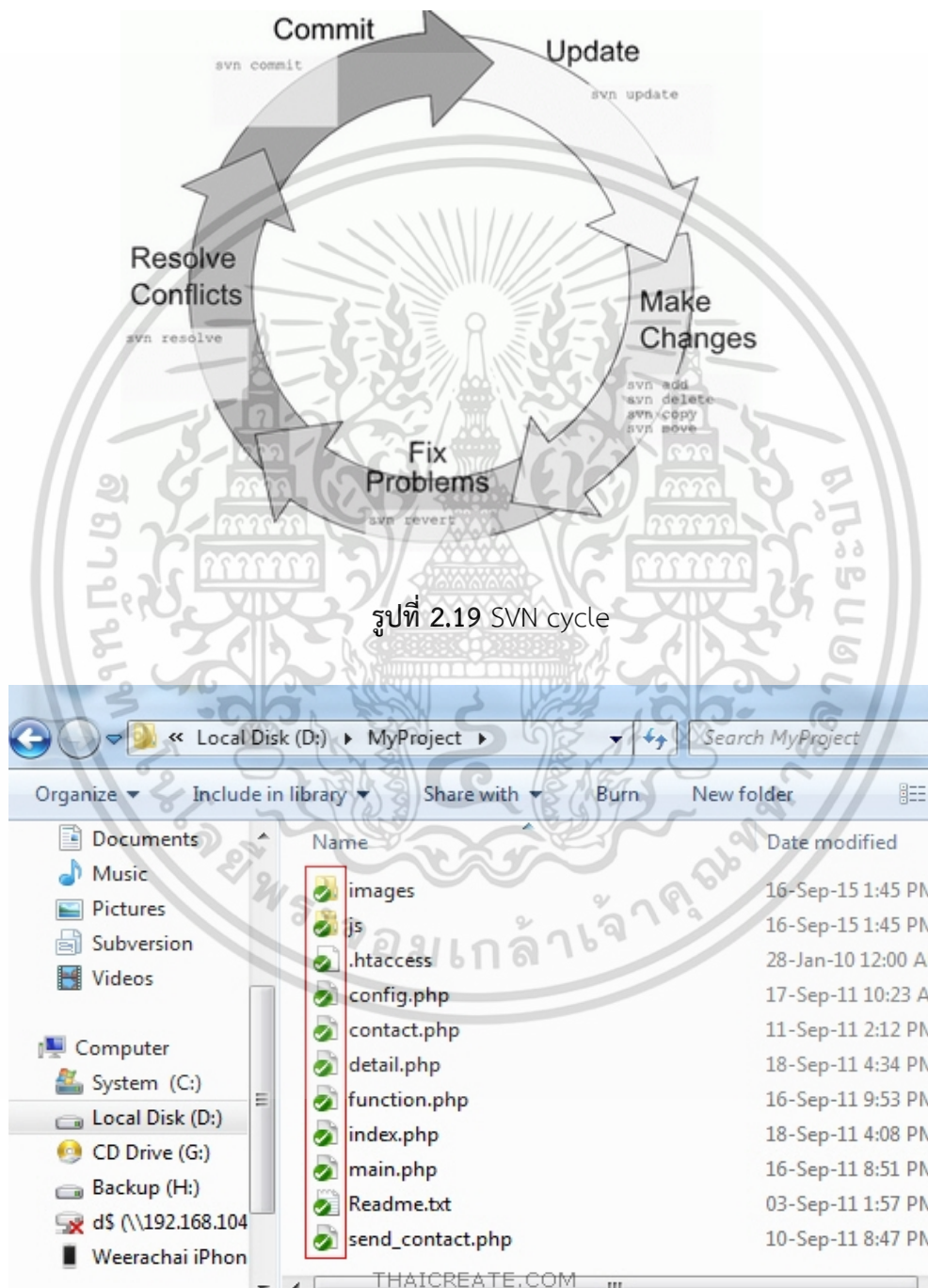


THAICREATE.COM

รูปที่ 2.18 SVN (Subversion)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

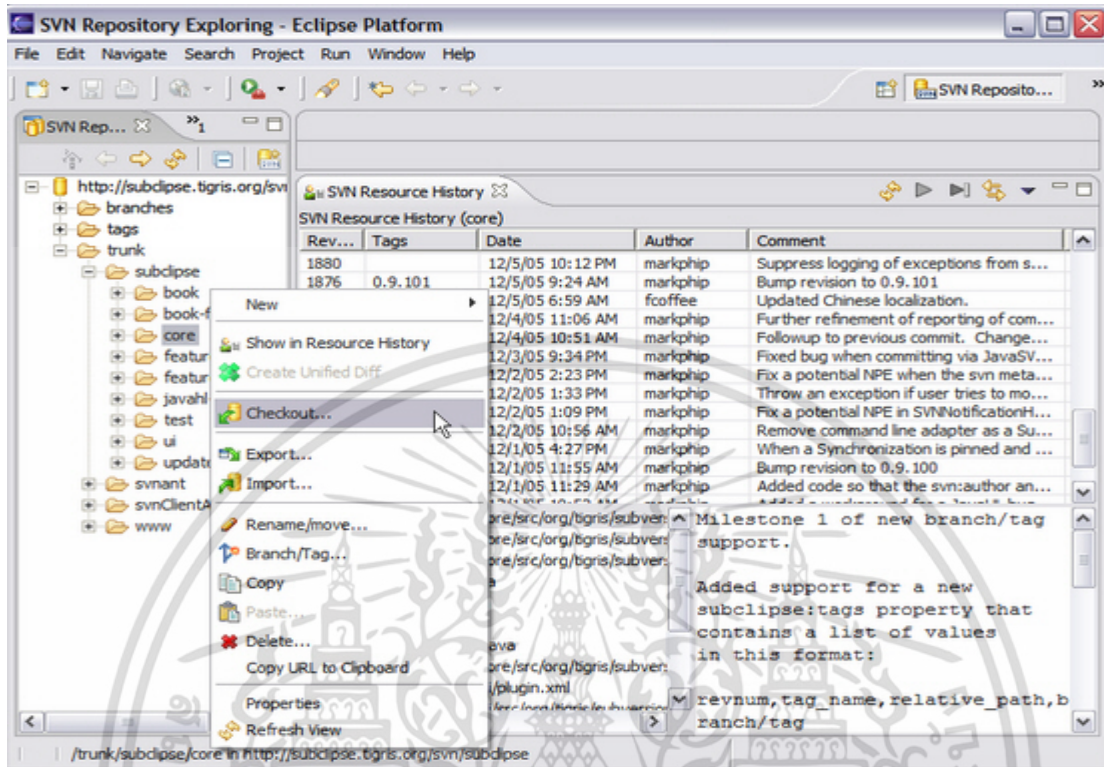
ในปัจจุบัน SVN ได้รับความนิยมอย่างมาก โดยเฉพาะโปรเจก Open Source ในต่างประเทศก็ใช้ SVN เป็น Source Version Control เข้ามาจัดการกับการทำงานเป็นทีม และในเมืองไทยในหลาย ๆ บริษัทก็ใช้กันเกือบทุก ๆ ที่ที่มีการพัฒนาโปรแกรมที่มีหลาย ๆ คนเข้ามาเกี่ยวข้อง ส่วนหนึ่งเพราะ SVN เป็น Open Source ที่สามารถใช้งานได้ฟรี โปรแกรมมีขนาดเล็ก อีกทั้งยังมี Plugin อีกมากมายที่รองรับบน IDE ร่วมกับการพัฒนาโปรแกรมหลาย ๆ ประเภท เช่น PHP , Java หรือจะเป็นโปรแกรมบน .Net Application ที่ได้รับความนิยมมากในปัจจุบัน



รูปที่ 2.20 Screenshot SVN

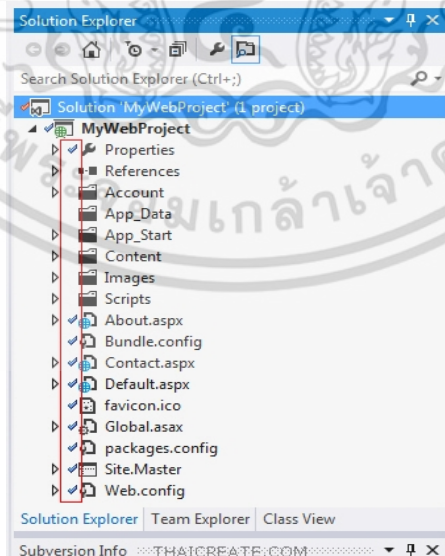
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SVN จัดเก็บไฟล์ได้ทุกประเภท ไม่ว่าจะเป็นไฟล์ Source Code ของ PHP, HTML, CSS และอื่น ๆ เกือบทุกประเภท



รูปที่ 2.21 จัดเก็บไฟล์ใน SVN

SVN มี Plugin ใช้กับโปรแกรม Eclipse ไว้เขียน PHP, Java, Android และอื่น ๆ อีกมากมาย

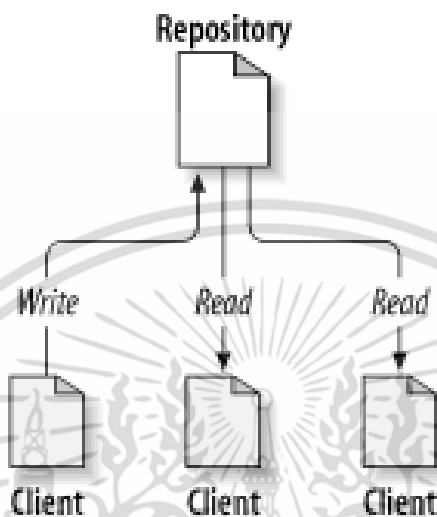


รูปที่ 2.22 Plugin ที่รองรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มี Plugin รองรับบน Visual Studio ทั้ง Version เก่า ๆ เช่น 2005, 2008 , 2010 และ รุ่นใหม่ ๆ เช่น 2012,2013, 2015

2.13.3.1 Repository



รูปที่ 2.23 Repository

Repository เป็นเหมือนศูนย์กลางในการเก็บข้อมูลซึ่งทุกคนจะใช้ร่วมกัน ดังนั้นหากมีการแก้ไขข้อมูลใน Repository โดยสมาชิกคนหนึ่ง คนอื่นก็จะเห็นการแก้ไขนั้นๆด้วย ถ้าพูดให้ง่ายก็คือมันเป็น Shared Folder เหมือนที่เห็นใน Network

Repository มีความสามารถในการเก็บ "สถานะ" เก่าๆย้อนหลังไว้ได้ด้วย ดังนั้นหากมีสมาชิกเผลอลบไฟล์บางไฟล์ออกไปจาก Repository ก็ยังสามารถกู้กลับมาได้โดยง่าย โดยการย้อนไป

สถานะแต่ละสถานะก็จะมีหมายเลขกำกับด้วย ซึ่งจะเรียกว่า Revision Number ซึ่งทำการแก้ไขเข้าไปหลายๆครั้ง Revision Number ก็จะมีเพิ่มมากขึ้น และสามารถดึงข้อมูลจาก Revision ใดๆใน Repo ก็ได้ Revision ล่าสุดใน Repo จะมีชื่อเรียกพิเศษเรียกว่า Head Revision

2.13.3.2 คำสั่งใน SVN

1) Checkout

ขั้นแรกต้องทำการดึงข้อมูลที่ต้องการออกมาจาก Repository ก่อน ข้อมูลมักจะเป็น Source Code และ Project File ทั้งหลายนั้นแหละ กระบวนการนี้เรียกว่าการ Checkout ซึ่งต้องระบุตำแหน่ง Repository และ Path ใน Repository ที่ต้องการ Checkout ออกมา หลังจาก Checkout ออกมาแล้ว ก็จะได้เป็น Copy อยู่บนเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Commit

หลังจากที่ Checkout ออกมาแล้ว หากทำการแก้ไขไฟล์บางไฟล์ และต้องการส่งผลการแก้ไขกลับไป Repo ก็สามารถทำได้โดยการ Commit สิ่งแก้ไขก็จะถูกส่งไปที่ Repository ทันที หากมีสมาชิกคนอื่นมา Checkout หลังจากนี้ ก็จะได้รุ่นที่แก้ไขไปแล้วด้วย

3) Update

ถ้ามีคนแก้ไขไฟล์อื่นแล้ว Commit กลับเข้าไปใน Repo การแก้ไขนั้นย่อมยังไม่ส่งผลใดๆกลับมาที่ Copy ที่อยู่บนเครื่องทำงาน

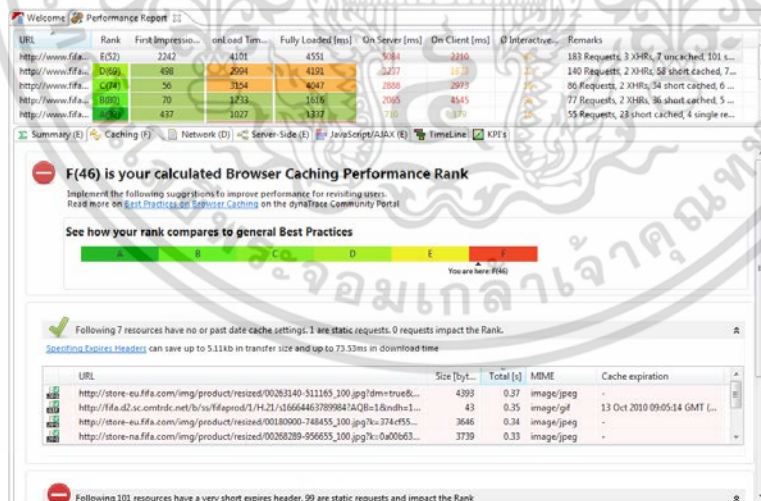
ดังนั้นต้องใช้คำสั่ง Update เพื่อให้ผลการแก้ไขนั้นลงมาบนเครื่องด้วย

4) Add, Delete

ในความเป็นจริงไม่ได้ทำการแก้ไขไฟล์อย่างเดียว มีการ เพิ่ม ลบ ไฟล์เข้าไปอยู่เสมอ เช่นตอนสร้าง Class ใหม่ เป็นต้น การเพิ่มไฟล์เข้าไปทำได้โดยใช้คำสั่ง Add และในทางกลับกัน การลบไฟล์ออกใช้คำสั่ง Delete

2.13.4 DynaTrace AJAX Edition

เป็นเครื่องมือฟรีที่จะแก้ปัญหาและตรวจสอบการปรับปรุงประสิทธิภาพในเว็บไซต์ สามารถดูการไหลของข้อมูลและ Monitor ปัญหาต่างๆที่เกิดขึ้นภายในเว็บไซต์ เป็นผลการดำเนินงาน Front-end ของเว็บไซต์



รูปที่ 2.24 หน้าจอ DynaTrace AJAX Edition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

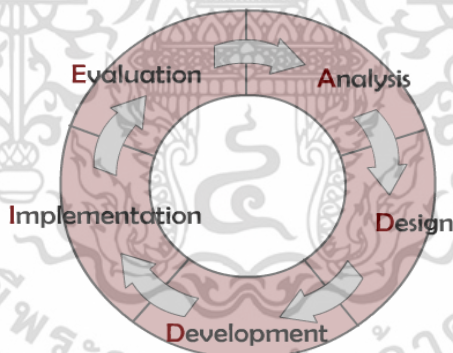
บทที่ 3

วิธีการดำเนินงานวิจัย

จากการที่ข้าพเจ้าได้เข้าร่วมการวิเคราะห์ ออกแบบ และดำเนินการเริ่มใช้งานระบบสำหรับ ศูนย์บริการข้อมูลลูกค้า เป็นระบบประเภท Web Application ประกอบการทำงานกับ Web Service ต่างๆ กับทางบริษัท Fusionware Company Limited ซึ่งมีการนำหลักสถาปัตยกรรมเชิงบริการ (Service Oriented Architecture หรือ SOA) มาใช้เป็นมาตรฐานในการพัฒนาระบบขนาดใหญ่ และมีการทำงานที่ซับซ้อน เพื่อทำให้เกิดความสะดวก สบายต่อการใช้งาน และนำหลักการของระบบบริหารความสัมพันธ์ลูกค้า (Customer Relationship Management หรือ CRM) มาใช้ในการพัฒนา Website ให้ดึงดูดลูกค้า สร้างความสัมพันธ์กับลูกค้า และวิเคราะห์ข้อมูลต่างๆของลูกค้า โดยส่วนที่ข้าพเจ้าได้ร่วมพัฒนาสามารถแบ่งได้เป็น 4 ส่วน คือ ส่วนของการตรวจสอบและดูแลการทำงานของระบบในช่วง Go-Live (Monitor) ส่วนของการติดตามปัญหา (Defect Tracking) ส่วนการทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request) และการทำเอกสารส่งมอบ (Delivery)

3.1 การตรวจสอบและดูแลการทำงานของระบบในช่วง Go-Live (Monitor)

ช่วง Go-Live หรือ การขึ้นใช้งานจริงของระบบ คือ ช่วงที่มีการนำระบบที่พัฒนาเสร็จแล้วขึ้นไปให้ ผู้ใช้งานระบบ (Agent) ใช้งานจริง และจำเป็นต้องมีผู้ดูแลระบบคอยตรวจสอบการทำงานของระบบ และบันทึกปัญหาที่เกิดขึ้นในการทำงานจริง เพื่อนำมาปรับปรุงและแก้ไขในส่วนต่างๆที่มีการผิดพลาด



รูปที่ 3.1 Go-Live Cycle

ขั้นตอนการตรวจสอบและดูแลการทำงานของระบบในช่วง Monitor

- 1) ดูแลการใช้งานระบบของผู้ใช้งานระบบ (Agent) ว่ามีข้อผิดพลาดหรือปัญหาหรือไม่
- 2) แนะนำการใช้งานบางส่วนและตอบปัญหาให้กับผู้ใช้งานระบบ (Agent)
- 3) ผู้ดูแลระบบจะคอยตรวจสอบประสิทธิภาพของระบบเมื่อใช้งานจริง และบันทึกใส่ไฟล์ Performance
- 4) เมื่อเกิดข้อปัญหาขึ้นผู้ใช้งานระบบ (Agent) และผู้ดูแลระบบจะคอยบันทึกปัญหาต่างๆ
- 5) ผู้ใช้งานระบบ (Agent) จะส่งที่บันทึกให้กับผู้รวบรวม
- 6) ผู้รวบรวมจะรวบรวมปัญหาทั้งหมดที่เกิดขึ้นจากระบบใส่ไฟล์ Summary Ticket
- 7) ผู้ดูแลระบบ จะมีบันทึกปัญหาทั้งหมดและนำไปใส่ไฟล์ Summary Defect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การติดตามปัญหา (Defect Tracking)

เมื่อเกิดปัญหาขึ้นในช่วงของการขึ้น Go-Live จะมีการประชุมและวิเคราะห์ปัญหาระหว่าง ผู้พัฒนา และผู้ใช้งานระบบ เพื่อหาแนวทางการแก้ไขปัญหาต่างๆ เมื่อหาต้นเหตุและสิ่งที่เป็นปัญหาได้แล้วจะนำมาแก้ไขโดยผู้พัฒนา และจะมีการใช้เครื่องมือมาช่วยในการติดตามปัญหาต่างๆ ซึ่งเครื่องมือที่ระบบนี้ใช้ก็คือ Jira ซึ่งเป็น Web Application

ขั้นตอนการติดตามปัญหา (Defect Tracking)

- 1) เมื่อวิเคราะห์และสรุปการประชุมได้รายละเอียดของปัญหาแล้วทำการใส่รายละเอียดต่างๆลงในไฟล์ Summary Defect
- 2) นำรายละเอียดของปัญหาต่างๆมาลงใน Create ลงใน Web Jira สามารถอธิบาย Text Box ต่างๆ ได้ดังนี้
- 3) เมื่อ Create ปัญหาต่างๆลงใน Jira แล้ว หัวหน้าทีมพัฒนา หรือ Senior Developer ก็ sẽ นำปัญหาต่างๆ ไปจัดการและมอบหมายให้กับ Developer แก้ไข
- 4) เมื่อปัญหาถูกแก้ไขเสร็จแล้ว ก็จะถูกทดสอบด้วยทีมทดสอบ หรือ Tester
- 5) เมื่อผ่านการทดสอบจากฝั่งของผู้พัฒนาแล้ว จะถูกนำไปทดสอบอีกที จากฝั่งผู้ใช้งานระบบ ก่อนนำปัญหาต่างๆที่ถูกแก้ไขแล้วขึ้นใช้งานจริง
- 6) ในการเปลี่ยนแปลงสถานะต่างๆของปัญหา จะถูก Update ในไฟล์ Summary Defect
- 7) เมื่อเกิดปัญหา หรือ ทีมพัฒนาต้องการข้อมูลเพิ่มเติม จะจัดให้มีการประชุมเพื่อวิเคราะห์

3.3 การทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request)

เมื่อเกิดปัญหาซึ่งไม่ได้เกิดจากความผิดพลาดของการทำงานของระบบ แต่เป็นปัญหาที่มาจากความต้องการของผู้ใช้งาน ความต้องการ (Requirement) เริ่มแรกจะมีความต้องการตั้งต้นมาจากผู้ใช้งานเพื่อให้ผู้พัฒนาระบบสามารถพัฒนาได้ตามความต้องการที่ตกลงกันได้ แต่เมื่อผู้ใช้ต้องการเพิ่มบางอย่างลงไปในระบบ แต่ไม่ได้มีการตกลงไว้ตั้งแต่แรก จะถูกเก็บเป็น การเปลี่ยนแปลงความต้องการ หรือ Change Request

บางปัญหาที่เกิดขึ้น เกิดจากการสื่อสารไม่ตรงกันระหว่าง ผู้ใช้งานและผู้พัฒนาระบบ หรือ อาจเกิดจากองค์ประกอบต่างๆที่มีอยู่ของระบบไม่เพียงพอ จึงต้องการองค์ประกอบบางตัวเพิ่ม แต่ไม่อยู่ใน เอกสาร Requirement แรกเริ่ม ปัญหาทั้งหมดนี้จึงต้องมีการสร้างเอกสารการเปลี่ยนแปลงความต้องการขึ้นมา เพื่อเป็นการตกลงกันระหว่างผู้ใช้งานและผู้พัฒนา

ขั้นตอนการทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request)

- 1) เมื่อวิเคราะห์หรือประชุมแล้วว่า ปัญหาเหล่านั้นคือ การเปลี่ยนแปลงความต้องการของผู้ใช้งาน ก็ จะมีการสร้างเอกสารขึ้นมาเพื่อเป็นตัวกลางในการตกลง (Change Request)
- 2) ผู้ทำเอกสาร จะส่งผลการวิเคราะห์ที่ได้จากการประชุมไปให้ทางฝั่งของทีมพัฒนา ได้แก้ไขระบบ ตามที่ผู้ใช้งานต้องการ
- 3) ผู้ทำเอกสาร จะมีการทำเอกสารขึ้นมาอีกฉบับ คือ Change Request Tracking จะใช้ในการติดตามสถานะของ เอกสาร Change Request แต่ละฉบับ
- 4) Change Request Tracking สามารถอธิบาย column

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก็เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) เมื่อปัญหาถูกแก้ไขและผ่านการทดสอบจากทางฝั่งผู้พัฒนาเรียบร้อยแล้ว จะถูกทดสอบจากทางผู้ใช้งาน
- 6) เมื่อผ่านการทดสอบทั้งหมด ผู้จัดทำเอกสารจะส่งเอกสารการเปลี่ยนแปลงความต้องการ (Change Request) เพื่อให้ผู้ใช้งาน เช่นอนุมัติ เพราะว่าการเปลี่ยนแปลงความต้องการเหล่านี้ ไม่ได้อยู่ในความต้องการตั้งต้น จึงต้องมีค่าใช้จ่ายเพิ่มในการพัฒนา
- 7) เมื่อทางฝั่งผู้ใช้งานอนุมัติแล้ว จะนำเอกสารการเปลี่ยนแปลงความต้องการทั้งหมด มาไว้ด้วยกัน เพื่อใช้ในการเรียกค่าใช้จ่ายต่อไป ในเอกสารจะประกอบด้วย

3.4 การทำเอกสารส่งมอบ (Delivery)

ก่อนที่จะมีการพัฒนา หรือการเริ่มการทำงานในส่วนการDevelopment จะมีการสรุป ช่วงการทำงานต่างๆของระบบ หรือ Phaseการทำงาน ภายในก็จะมี Feature หรือความสามารถในการทำงานของระบบต่างๆ ที่ผู้ใช้งานระบบต้องการให้เกิดขึ้น จะมีการทำเอกสารรวบรวม Feature ทั้งหมดไว้และแบ่งแยกไปในช่วงการทำงานแต่ละช่วง

เมื่อการ Go-Live เสร็จสิ้นแล้ว ระบบจะยังถูกใช้งานต่อ อยู่ในช่วงการรับประกันการใช้งาน หลังจากนั้นเมื่อมีปัญหาเกิดขึ้นจากการทำงานผิดพลาดของระบบ ทีมพัฒนาจะรับรู้และแก้ไขในทรัพยากรของทีมพัฒนาและจะถูกนำขึ้นมา Deploy ลงการใช้งานจริงในช่วงเวลาที่มีการกำหนดไว้ รวมถึงการนำ Feature ที่มีกำหนดใช้งาน ณ ช่วงเวลานั้น โดยจะมีเอกสารส่งมอบคอย เช็สถานะของFeature ที่ถูกนำขึ้นใช้งาน และส่งให้ทางฝั่งของผู้ใช้งานได้รับรู้

เอกสารส่งมอบ หรือ Delivery Document คือ เอกสารที่จะรวบรวม Feature หรือระบบการทำงานย่อยต่างๆ ในช่วงของการทำงาน (Phase) นั้นๆ เพื่อให้ผู้ใช้งานระบบ ได้ทราบถึงช่วงเวลา การนำระบบย่อยนั้นขึ้นใช้งานจริง สถานะของFeature และปัญหาภายใน Feature แต่ละระบบ เอกสารส่งมอบจะถูกนำส่งให้กับทางผู้ใช้งาน หลังจากช่วงที่การรับประกันระบบเสร็จสิ้นแล้ว

3.4.1 ขั้นตอนการทำเอกสารส่งมอบ (Delivery)

- 1) ก่อนเริ่มพัฒนา ผู้ใช้งานและทีมพัฒนา ประชุมลงความเห็นและสรุป Feature ต่างๆ ทีมพัฒนาจะทำเอกสารส่งมอบ ซึ่งจะรวบรวม Feature ต่างๆไว้หมด ในแต่ละ Phase
- 2) ก่อนที่ระบบจะถูกนำขึ้นใช้งานจริง Go-Live จะมีการรวบรวมสถานะ Feature ต่างๆ ที่จะถูกนำขึ้นไปพร้อมกับระบบในรอบการ Deploy นั้นๆ สถานะจะแบ่งเป็น
 - Pass: Feature ถูกลงความเห็นที่สามารถใช้งานจริงได้แล้วในระบบ
 - Cancel: Feature นั้น ถูกยกเลิกในการทำงานจริง เพราะเหตุผลต่างๆ
 - Fixing: Feature นั้น อยู่ระหว่างการแก้ไข และระบุ จำนวนปัญหา
 - Retest: Feature ที่จะถูกทดสอบจากuser หลังจากการแก้ไขเสร็จแล้ว
- 3) เมื่อเกิดปัญหาต่างๆ เกิดจากความผิดพลาดของระบบ และมีความเกี่ยวข้องกับ Featureต้องมีการประชุม ระหว่างทางผู้ใช้งานและทีมพัฒนา เพื่อหาสาเหตุและวิธีการแก้ไข
- 4) ทีมพัฒนาต้องนำปัญหาที่เกิดขึ้นมาแก้ไข และสร้าง Issue ขึ้นมาภายใน Jira ด้วย ซึ่งจะทำให้ Developer รับรู้ปัญหาที่เกิดขึ้น สามารถแก้ไข และUpdateได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) เมื่อสถานะต่างๆถูก Update ลงเอกสาร และจบช่วงของการรับประกันระบบ ในแต่ละPhase จะมีการส่งเอกสารส่งมอบในPhase การทำงานนั้นๆ ต่อผู้ใช้งาน
- 6) เมื่อการทำงานของระบบสามารถใช้งานได้ตามที่ผู้ใช้งานต้องการ และไม่มีปัญหาที่เกิดจากความผิดพลาดของระบบ ทีมพัฒนาจะนำเอกสารส่งมอบทั้งหมด รวบรวมและ ส่งมอบให้กับฝั่งของผู้ใช้งานระบบ

No	Feature	Feature List	Status	Target Fix Date

ตารางที่ 3.1 แบบฟอร์ม เอกสารส่งมอบ (Delivery) ส่วนของ Feature List

อธิบาย หัวข้อของเอกสาร

- No.: ลำดับของการทำงานย่อยใน Feature
- Feature: ลำดับของ Feature
- Feature List: ชื่อของ Feature นั้น
- Status: สถานะของแต่ละ Feature
- Pass: Feature ถูกลงความเห็นว่าเป็นสามารถใช้งานจริงได้แล้วในระบบ
- Cancel: Feature นั้น ถูกยกเลิกในการทำงานจริง เพราะเหตุผลต่างๆ
- Fixing: Feature นั้น อยู่ระหว่างการแก้ไข และระบุ จำนวนปัญหา
- Retest: Feature ที่จะถูกทดสอบจากuser หลังจากการแก้ไขเสร็จแล้ว
- Target Fix Date: วันที่แก้ไขเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

จากที่ได้กล่าวไว้ในบทที่ 3 ว่า มีการแบ่งการทำงานออกเป็น 4 ส่วน คือ ส่วนของการตรวจสอบและดูแลการทำงานของระบบในช่วง Go-Live (Monitor) ส่วนของการติดตามปัญหา (Defect Tracking) ส่วนการทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request) และการทำเอกสารส่งมอบ (Delivery) ดังนั้นจึงนำมาสรุปผลการดำเนินการได้ ดังนี้

4.1 ผลการดำเนินการของการตรวจสอบและดูแลการทำงานของระบบในช่วง Go-Live (Monitor)

4.1.1 สร้างโฟลเดอร์ (Folder) และเตรียมไฟล์ข้อมูลที่ใช้ในบันทึกผลการทำงานและปัญหาต่างๆ ในช่วง Go-Live

ในขั้นตอนนี้จะเป็นการบันทึกผลของการทำงานและปัญหาต่างๆ หลังจากที่ได้นำระบบขึ้นใช้งานจริง ทำให้สามารถใช้งาน Feature ต่างๆ ได้ตามที่กำหนดไว้ แต่อาจจะมีปัญหาหรือทรัพยากรของระบบเกิดความขัดข้องเพราะการใช้งานจริง ระบบถูกใช้งานจาก User จริงๆ ที่มีจำนวนมาก จึงอาจจะทำให้เกิดปัญหาต่างๆ ได้ จำเป็นต้องมีทีมจากผู้พัฒนามาบันทึกผลการทำงานและเก็บรายละเอียดต่างๆ จะได้รับรู้และสามารถแก้ไขได้

4.1.2 ผลการดำเนินการ หลังจากการประชุม

ไฟล์ MOM (Minutes of Meeting)

: เมื่อทราบถึงปัญหาต่างๆ ที่เกิดขึ้น ซึ่งต้องมีการจัดประชุมเพื่อทำการ วิเคราะห์และหาทางแก้ไข ปัญหา ซึ่งทาง User ต้องเข้าร่วมการประชุมด้วย ระหว่างการประชุมต้องมีการทำเอกสาร MOM (Minutes of Meeting) เป็นเอกสารที่ใช้บันทึกเนื้อหาในการประชุม

4.1.3 ผลการดำเนินงาน หลังจากการวิเคราะห์และหาทางแก้ไขปัญหาต่างๆ

ไฟล์ที่รวบรวมปัญหาต่างๆ ที่ได้รับการวิเคราะห์และหาทางทีมผู้พัฒนาพร้อมแก้ไข Summary Defect

: เมื่อทำการวิเคราะห์ปัญหาเสร็จแล้ว ทางทีมพัฒนาจะรับทราบถึงปัญหาและทำการรวบรวมปัญหาที่ต้องแก้ไขใส่ไฟล์ Summary Defect เพื่อสะดวกต่อการทราบถึงความคืบหน้าในการแก้ไขปัญหา และรู้ประสิทธิภาพในการทำงานของทีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการดำเนินการของการติดตามปัญหา (Defect Tracking)

ต่อเนื่องจากบทที่ 3 ที่ทำการนำรายละเอียดต่างๆของปัญหาลง Web Jira เพื่อทำการ Defect Tracking หรือการติดตามสถานะการแก้ไขปัญหาต่างๆ ของทีมพัฒนา เพื่อที่จะได้ทราบถึงปัญหาต่างๆที่แก้ไขเสร็จแล้ว เพื่อจะนำมาวิเคราะห์และวางแผนการทำงานต่อไป

4.3 ผลการดำเนินการของการทำเอกสารการเปลี่ยนแปลงความต้องการ (Change Request)

ต่อเนื่องจากบทที่ 3 ที่มีการทำเอกสารเปลี่ยนแปลงความต้องการ (Change Request) จึงต้องทำไฟล์เพื่อติดตามสถานะ การแก้ไข และยังสามารถบอกรายละเอียดของ CR ต่างๆได้

4.4 ผลการดำเนินการของการทำเอกสารส่งมอบ (Delivery DOC.)

ต่อเนื่องจากบทที่ 3 ที่มีการทำเอกสารส่งมอบ (Delivery DOC.) เมื่อสร้างไฟล์เสร็จแล้วก็จะมีการประชุมร่วมกับฝั่ง User เพื่อนำเอกสารให้กับฝั่ง User ได้รับรู้และเซ็นยินยอม ในแต่ละส่วนก็ได้อธิบายไปในบทที่ 3 แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

1) องค์กร

- สามารถเพิ่มประสิทธิภาพขององค์กรในการให้บริการและดูแลลูกค้า
- เพิ่มเครื่องมือในการทำงานของพนักงาน
- เพิ่มผลกำไรขององค์กรที่ได้จากการบริการลูกค้า

2) ผู้ใช้งาน (Agent)

- ผู้ใช้สามารถลดเวลาในการรอการประมวลผลของข้อมูลมาแสดงบนหน้าจอ เพื่อความสะดวกในการให้บริการแก่ลูกค้า
- ผู้ใช้สามารถลดเวลาในการศึกษาระบบ และใช้ระยะเวลาในการเข้าถึงระบบลดน้อยลง
- ผู้ใช้สามารถดูรายงานต่างๆ เพื่อประกอบการตัดสินใจในการบริหารจัดการงานได้
- ลดความผิดพลาดซึ่งเกิดจากตัวของผู้ใช้งาน (Agent) ลง จากการทำงานแบบ Manual
- สามารถเพิ่มประสิทธิภาพในการทำงานและทำงานได้ตรงตามเป้าหมาย
- ลดความซับซ้อนของหน้าจอการทำงาน ทำให้ผู้ใช้ไม่สับสน

3) ผู้ศึกษาและวิจัย

- ได้เรียนรู้การทำงานในสถานการณ์จริง
- ได้รู้จักแก้ไขปัญหาในการทำงานร่วมกับผู้อื่น
- เพิ่มทักษะในการสื่อสารกับผู้อื่น
- เพิ่มคุณลักษณะของตนเอง เช่น การตรงต่อเวลา ความรับผิดชอบ ความอดทน
- เพิ่มความสามารถในการใช้เครื่องมือต่างๆ ที่ใช้ในการทำงานจริง
- ได้วิเคราะห์การทำงานของระบบ สามารถเข้าใจหลักการคิดของระบบได้ง่ายขึ้น
- สามารถเข้าใจหลักการคิดในเชิงธุรกิจ
- สามารถเข้าใจความต้องการของผู้ใช้งานระบบ และอธิบายให้ผู้อื่นได้

5.2 ข้อเสนอแนะ

- 1) ในการทำงานจริงต้องคิดอย่างถี่ถ้วนและรอบคอบที่สุด เพราะถ้าเกิดความผิดพลาด อาจนำมาซึ่งความเสียหายมากมายที่เกิดขึ้นในระบบ
- 2) ในการทดสอบจำเป็นต้องใช้ข้อมูลที่เหมือนจริงมากที่สุด เพื่อผลการทดสอบที่แม่นยำ
- 3) เนื่องจากความต้องการอาจมีการเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นในการพัฒนาต่อควรมีการตรวจสอบความถูกต้องอีกครั้งหนึ่ง
- 4) ในความเป็นจริง ชีวิตการทำงานต้องแบ่งส่วนงานกันทำ และต้องทำงานเป็นทีม เราจะทำตามใจไม่ได้โดยเด็ดขาด ซึ่งการตัดสินใจอะไรก็ตาม ต้องได้ความเห็นชอบจากทีมเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[b8%a3%e0%b8%b0%e0%b8%aa%e0%b8%b4%e0%b8%97%e0%b8%98%e0%b8%b4%e0%b8/](http://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2055-database-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A323.html)

[10] Database คืออะไร.

สืบค้นเมื่อ 27 เมษายน 2559 แหล่งที่มา :

<http://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2055-database-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A323.html>

[11] SOAP (Simple Object Access Protocol).

สืบค้นเมื่อ 27 เมษายน 2559 แหล่งที่มา :

<http://55206600055.sci.dusit.ac.th/work9/index.html>

[12] SVN (Subversion).

สืบค้นเมื่อ 27 เมษายน 2559 แหล่งที่มา :

<http://www.thaicreate.com/tutorial/svn-subversion-version-control.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

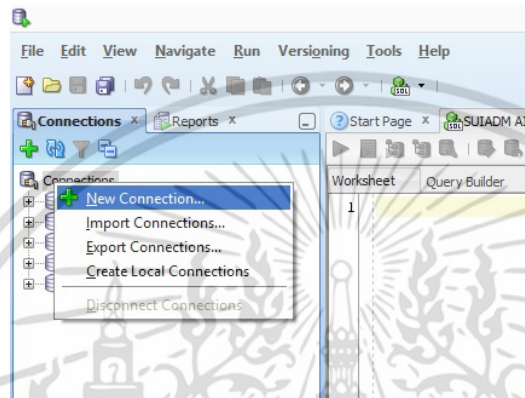
ภาคผนวก ก

การเริ่มต้นการใช้งาน

โปรแกรม Oracle SQLdeveloper

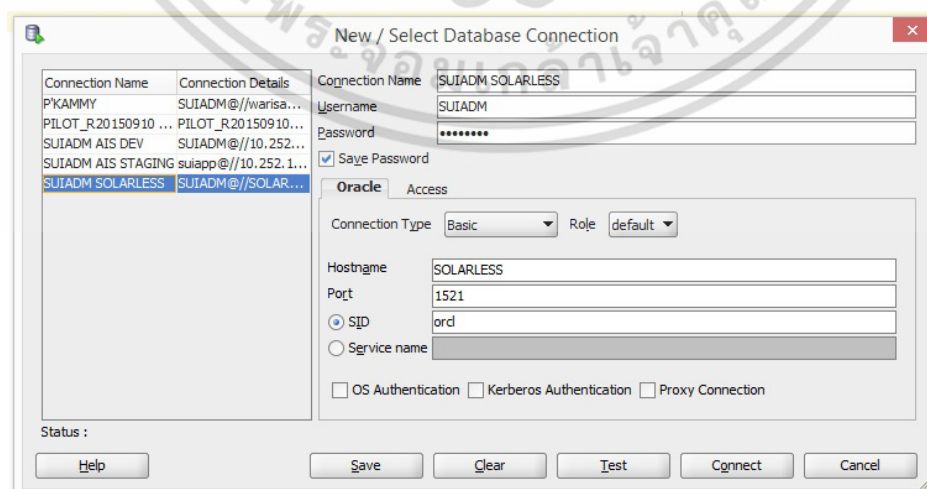
ก.1 การเพิ่ม Connection ในฐานข้อมูล

- คลิกขวาที่ Connections แล้วเลือก New Connection



รูปที่ ก.1 การเพิ่ม Connection ในฐานข้อมูล

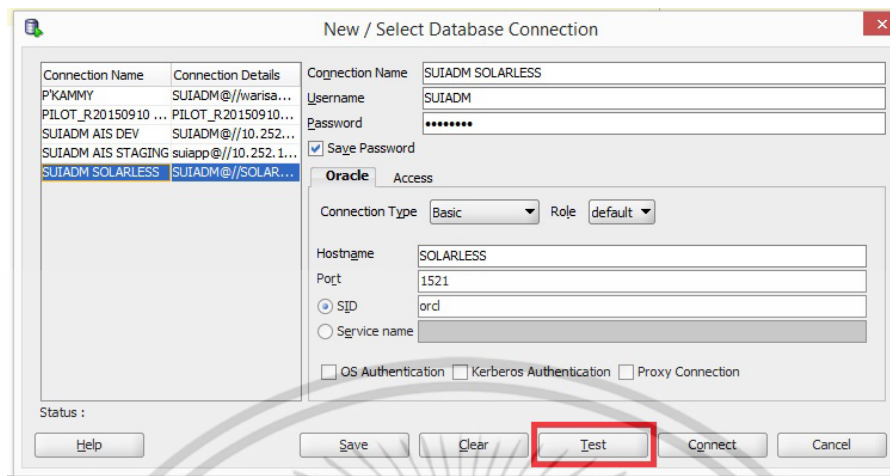
- จะมีหน้าต่างปรากฏขึ้นมาให้กรอกข้อมูล โดยที่
 - Connection Name: เป็นชื่อของ Connection ใหม่ ตั้งชื่ออะไรก็ได้
 - Username: ชื่อของผู้ใช้งานระบบ (Agent) ฐานข้อมูลที่ต้องการเชื่อมต่อ
 - Password: รหัสของผู้ใช้งานระบบ (Agent) ฐานข้อมูลที่ต้องการเชื่อมต่อ
 - Hostname: Hostname ที่ต้องการเชื่อมต่อ
 - Port: ที่อยู่ของฐานข้อมูลที่ต้องการเชื่อมต่อ
 - SID: ชื่อ Service Name



รูปที่ ก.2 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]

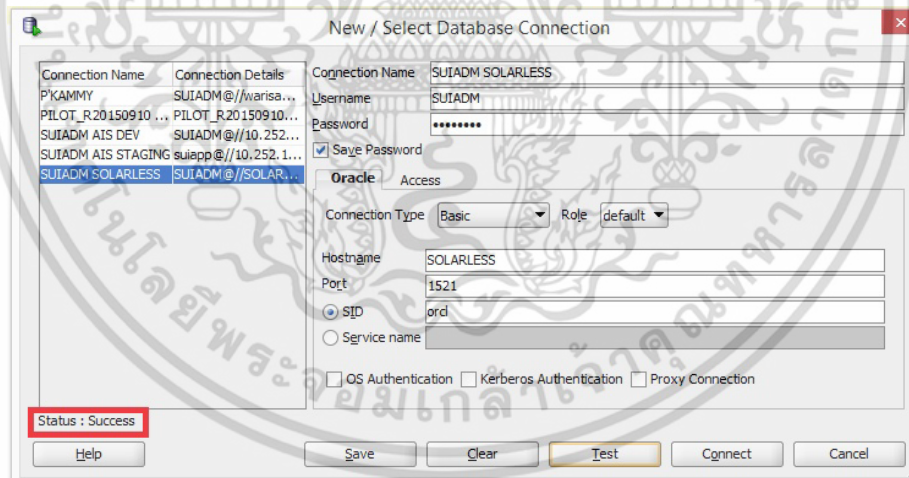
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นให้ทำการคลิก Test เพื่อทดสอบการเชื่อมต่อกับ Connection ใหม่



รูปที่ ก.3 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]

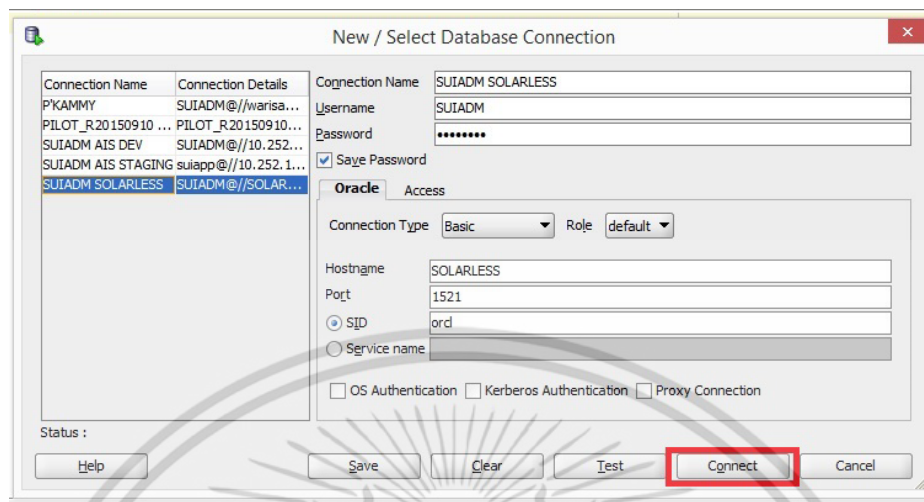
- ถ้าเชื่อมต่อสำเร็จจะมีข้อความ Success ปรากฏในส่วนของ Status



รูปที่ ก.4 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]

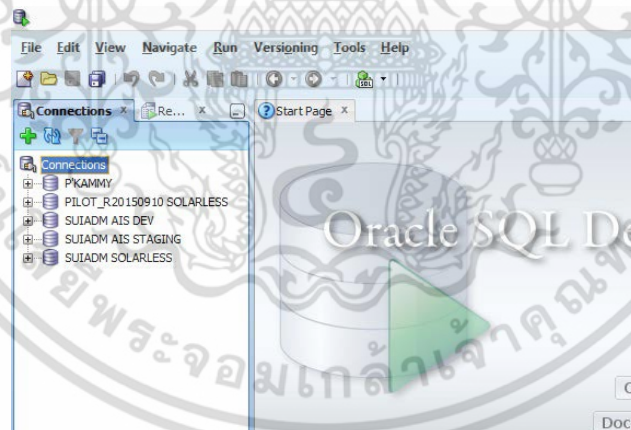
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิก Connect เพื่อทำการสร้าง Connection ใหม่



รูปที่ ก.5 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]

- จะได้ Connection (Schema) ใหม่ ตามต้องการ

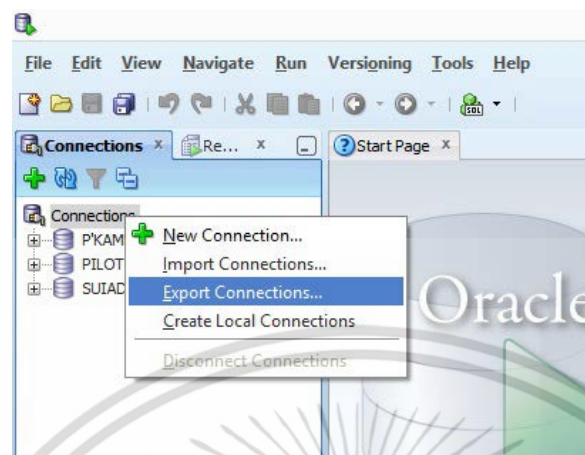


รูปที่ ก.6 การเพิ่ม Connection ในฐานข้อมูล [ต่อ]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

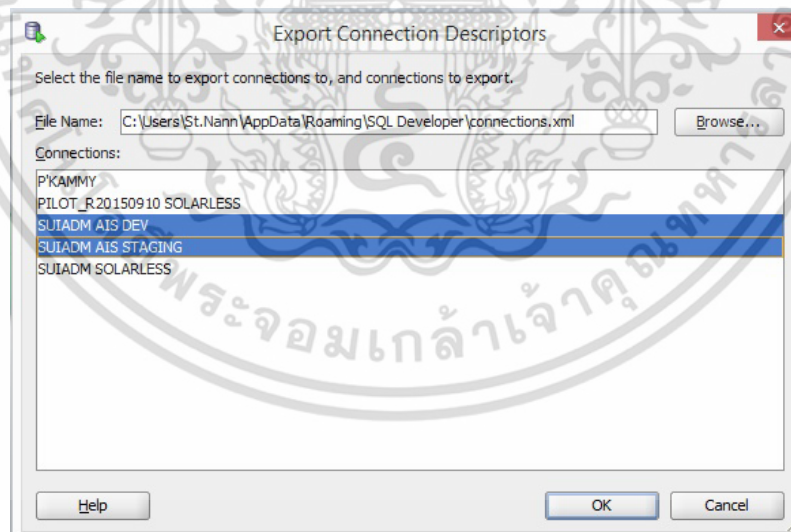
ก.2 การ Export Schema ฐานข้อมูลไปใช้งานในเครื่องอื่น

- คลิกขวาที่ Connections แล้วเลือก Export



รูปที่ ก.7 การ Export Schema ในฐานข้อมูล

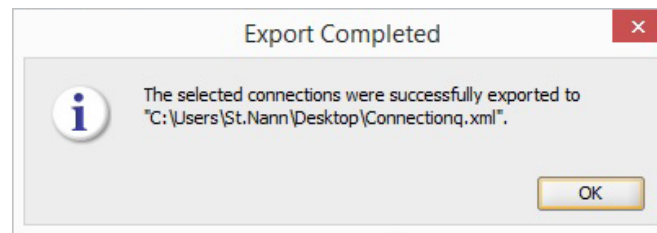
- จะมีหน้าต่างปรากฏขึ้นมา เพื่อให้เลือก Schema ที่ต้องการ Export และให้เลือกที่อยู่ที่จะเก็บไฟล์ ซึ่งการ Export ในรูปแบบนี้ จะได้ไฟล์นามสกุล .xml เมื่อเลือกเรียบร้อยแล้วให้คลิก OK



รูปที่ ก.8 การ Export Schema ในฐานข้อมูล [ต่อ]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

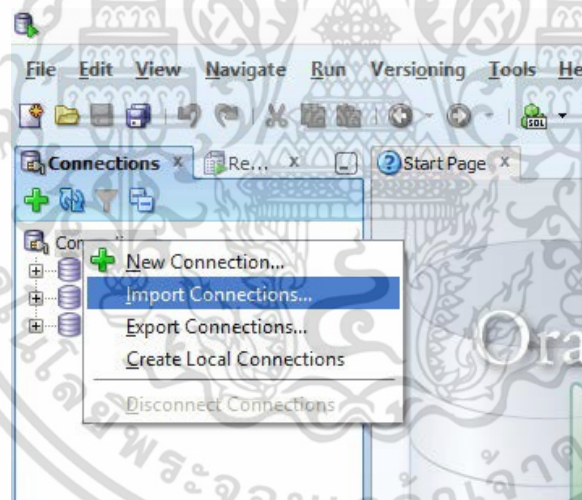
- จากนั้นจะมีกล่องข้อความแจ้งขึ้นมาว่า ทำการ Export เรียบร้อยแล้ว ดังรูป



รูปที่ ก.9 การ Export Schema ในฐานข้อมูล [ต่อ]

ก.3 การ Import Schema ฐานข้อมูลที่ได้จากเครื่องอื่น

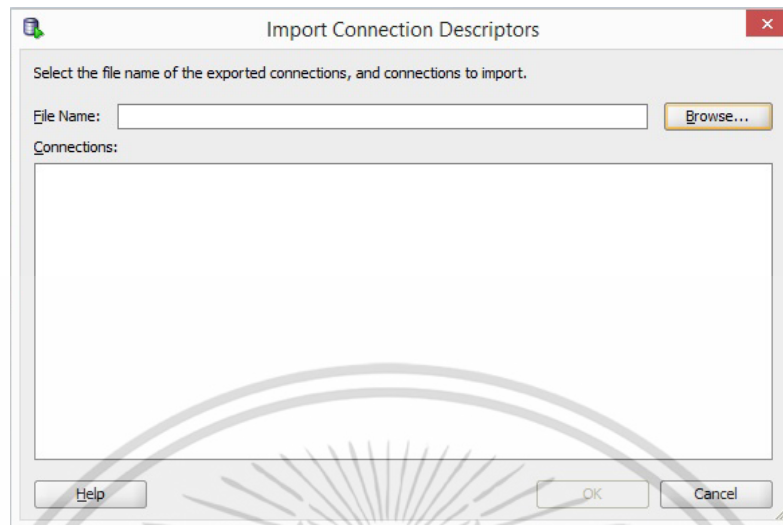
- คลิกขวาที่ Connections แล้วเลือก Import



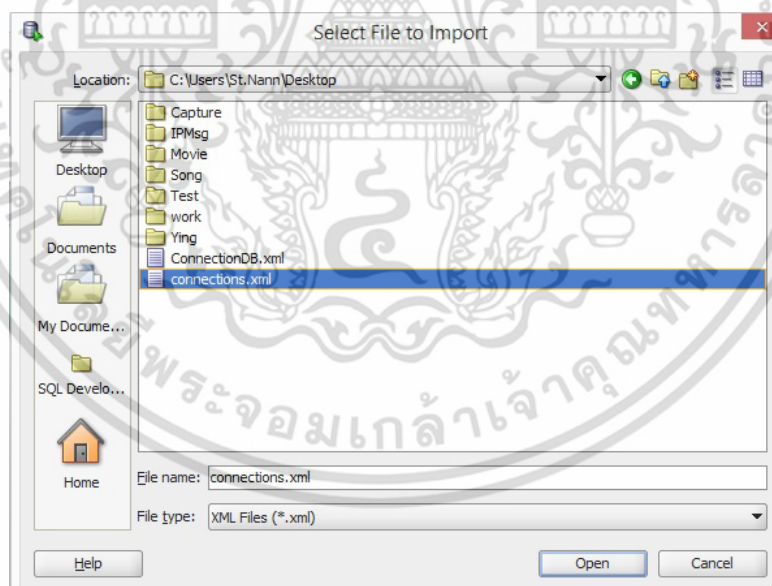
รูปที่ ก.10 การ Import Schema ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จะมีหน้าต่างปรากฏขึ้นมา ให้ทำการเลือกไฟล์ .xml ที่ทำการ Export มาก่อนหน้า



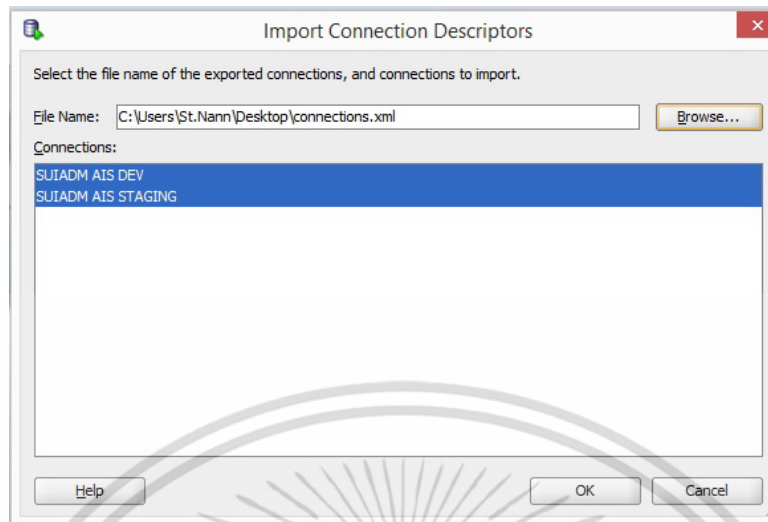
รูปที่ ก.11 การ Import Schema ในฐานข้อมูล [ต่อ]



รูปที่ ก.12 การ Import Schema ในฐานข้อมูล [ต่อ]

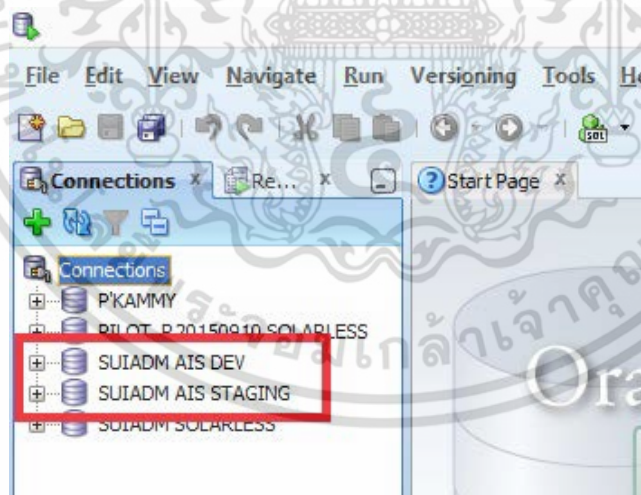
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากนั้นคลิก OK



รูปที่ ก.13 การ Import Schema ในฐานข้อมูล [ต่อ]

- จะได้ Schema ตามต้องการ



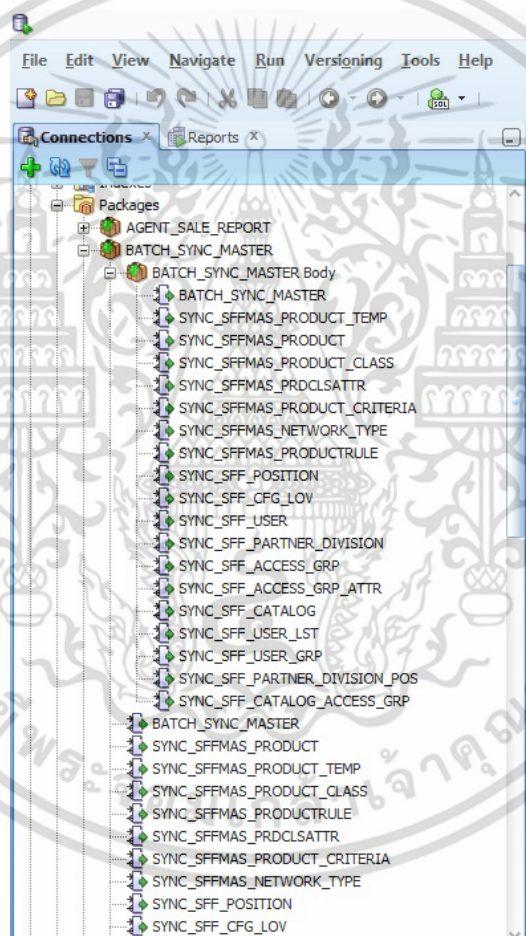
รูปที่ ก.14 การ Import Schema ในฐานข้อมูล [ต่อ]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.4 การสร้าง Package

เนื่องจากระบบที่ทำนี้เป็นระบบที่มีขนาดใหญ่ ซึ่งมีการทำตาราง (Tables) ต่างๆไว้ก่อนแล้ว ดังนั้น ในส่วนนี้จึงขอกว่าแต่การสร้าง Package เพื่อนำไปใช้งานต่อในส่วนอื่นๆ

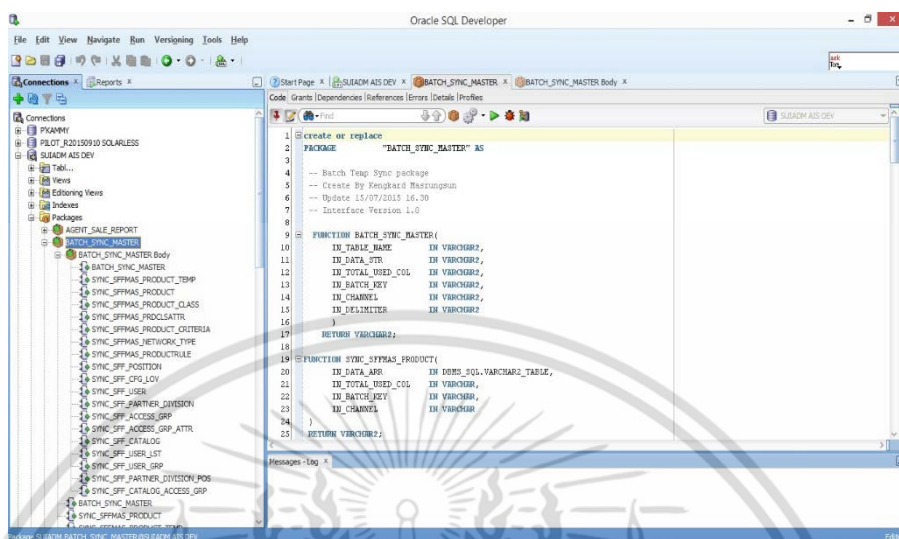
- หลังจากการสร้าง Connection ใหม่ หรือ Import Schema เข้ามาเรียบร้อยแล้ว ให้ทำการคลิกเครื่องหมายบวก (+) ที่หน้า Schema เพื่อขยายออก แล้วเลือก Packages ซึ่งจะประกอบด้วยหลายๆ Package ให้เลือกชื่อ Package ที่ต้องการ ในที่นี้ Package ชื่อ BATCH_SYNC_MASTER โดยภายใต้ชื่อ Package ที่เลือกนั้น จะมีส่วนที่เรียกว่า HEAD และ BODY



รูปที่ ก.15 การ Package ในฐานข้อมูล

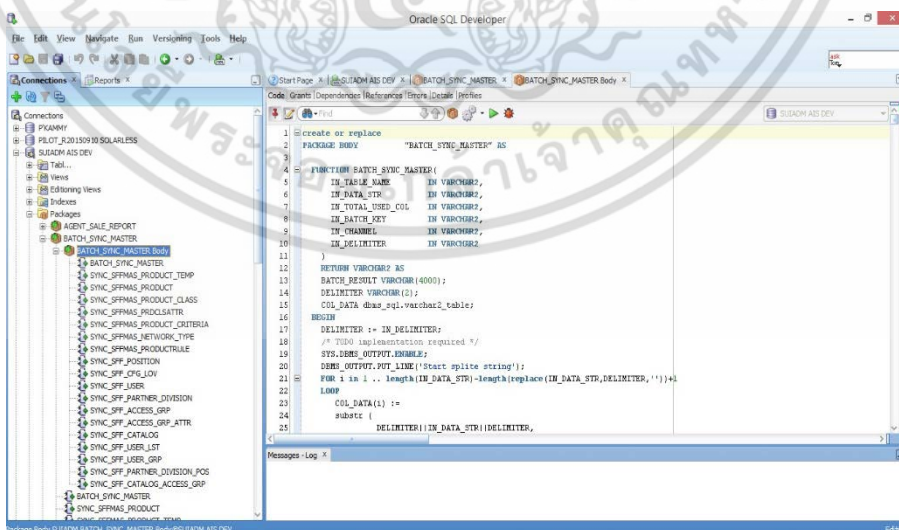
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ให้คลิกที่ส่วน HEAD ซึ่งก็คือ ส่วนที่เป็นชื่อ Package (BATCH_SYNC_MASTER) ซึ่งส่วนนี้จะเป็นส่วนที่ประกาศชื่อ Package และตัวแปรต่างๆ (สามารถดู Code ได้ในภาคผนวก ง.)



รูปที่ ก.16 การ Package ในฐานข้อมูล [ต่อ]

- เมื่อทำการสร้างในส่วน HEAD เรียบร้อยแล้ว ให้ทำการสร้างส่วน BODY ซึ่งจะเป็นส่วนที่ทำการกระบวนกรต่างๆเกี่ยวกับข้อมูล เช่น การเพิ่มข้อมูล (Insert), การเปลี่ยนแปลงข้อมูล (Update) เป็นต้น โดยการคลิกที่ BATCH_SYNC_MASTER body เมื่อทำการเขียน Code เรียบร้อยแล้วให้คลิก Save



รูปที่ ก.17 การ Package ในฐานข้อมูล [ต่อ]

เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้