

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง
ระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์

SOFTWARE DEVELOPMENT EFFORT ESTIMATION SYSTEM



H005936

โดย



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

อพ.

ค 5865

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

หมู่.....

9561

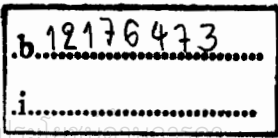
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ทะเบียน.....

05936

ภาคเรียนที่ 2 ปีการศึกษา 2551

เดือนปี ๕ 3 0 11 2553



ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOFTWARE DEVELOPMENT EFFORT ESTIMATION SYSTEM



**A SYSTEM DEVELOPMENT PROJECT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2/2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2009

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	ระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์
นักศึกษา	นายธิดิษฐ์ วงศ์ปิยะ
รหัสนักศึกษา	50066535
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	รศ.ดร.อาริต ธรรมโน
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร. ธนารัตน์ ชลิตาพงศ์

บทคัดย่อ

ระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ พัฒนาขึ้นเพื่อช่วยในการบริหารโครงการในด้านการประมาณภาระงานหรือการประเมินขนาดของโครงการ ซึ่งการทำงานของระบบนั้นประกอบไปด้วยการบันทึกข้อมูลโครงการ การประมาณภาระงาน การจัดการข้อมูลตั้งต้นของระบบ การปรับปรุงค่าพารามิเตอร์ต่างๆที่ใช้ในแบบจำลองในการประมาณภาระงาน รวมถึงสร้างรายงานเพื่อใช้สำหรับบริหารโครงการ ซึ่งระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นี้จะช่วยให้การประมาณภาระงานและการบริหารโครงการมีประสิทธิภาพเพิ่มมากขึ้น สร้างมาตรฐานการประมาณภาระงานของบริษัท ตลอดจนช่วยลดปัญหาการส่งมอบงานล่าช้าเกินกำหนด โดยระบบจะถูกพัฒนาในรูปแบบของเว็บแอปพลิเคชัน ด้วยภาษาโปรแกรม JAVA และ JSP และจัดเก็บข้อมูลบนฐานข้อมูล MySQL สำหรับการประมาณภาระงานนั้นได้นำแบบจำลอง COCOMO II มาประยุกต์ใช้ในการประมาณภาระงาน และจำนวนบุคคลากรที่ใช้ในการพัฒนาซอฟต์แวร์ของโครงการ.

Title	Software Development Effort Estimation System
Student	Mr. Thitichai Wongpiya
Student ID.	50066535
Degree	Master of Science
Programme	Information Science
Academic Year	2008
Advisor	Assoc. Prof. Dr. Arit Thammano
Co-Advisor	Asst. Prof. Dr. Thanarat Chalidabhongse

ABSTRACT

The Software Development Effort Estimation System is developed to help project managers in estimating numbers of resources needed and schedule of the project. This system is combined with many functions such as project information management, effort estimation, model's parameter calibration and summary report for project manager. The benefits of this information system are helping to increase standard and efficiency of effort estimation including, decrease the problems of software overdue delivery. Then, the system is designed on web based application with JAVA and JSP while using MySQL as a database management system (DBMS). Moreover, we apply a well-known COCOMO II effort estimation model to estimate effort needed as well as the time required for projects.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้สำเร็จได้ ด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร. ธนรัตน์ ชลิตาพงศ์ และ รศ.ดร. อาริต ธรรมโน ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้าขอขอบพระคุณอย่างยิ่งที่ท่านได้ให้ความอนุเคราะห์ด้วยดีเสมอมา จนกระทั่งพัฒนาโครงการนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณอาจารย์คณะเทคโนโลยีสารสนเทศ และอาจารย์ต่างคณะ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่เข้ามาให้ความรู้ ตลอดจนอาจารย์ท่านอื่นที่มา จากต่างสถาบัน ที่ได้ให้ความรู้กับข้าพเจ้า

ขอขอบคุณบริษัท โมทีฟ เทคโนโลยี จำกัด (มหาชน) ที่อนุญาตให้ลาหยุดงานในช่วงสอบ และให้ออกก่อนเวลาเลิกงานเพื่อเรียนในตอนเย็น รวมถึงเพื่อนร่วมงานที่สนับสนุนกันตลอดมา

ขอบคุณเพื่อนๆ พี่ๆ รุ่น 23.2 และเพื่อนไอที ปร่าจินบุรีทุกคนที่มีส่วนสนับสนุน ให้กำลังใจและเป็นที่ยกย่องในทุกๆเรื่อง

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ข้าพเจ้าขอระลึกในพระคุณและขอกราบขอบพระคุณมา ณ ที่นี้

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ธิดิษฐ์ วงศ์ปิยะ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมา.....	1
1.2 วัตถุประสงค์ในการพัฒนาระบบ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนในการดำเนินโครงการ.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 เทคโนโลยีที่เกี่ยวข้อง.....	5
2.1 ยูเอ็มแอล (UML).....	5
2.1.1 ยูสเคสไดอะแกรม (Usecase Diagram).....	5
2.2.2 คลาสไดอะแกรม (Class Diagram).....	5
2.2 ไฮเบอร์เนท (Hibernate).....	8
2.3 แบบจำลอง COCOMO II.....	9
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	15
3.1 การศึกษาระบบงานปัจจุบัน.....	15
3.1.1 รูปแบบการทำงานปัจจุบัน.....	15
3.1.2 ปัญหาของระบบการทำงานปัจจุบัน.....	16
3.2 การวิเคราะห์และออกแบบระบบงานใหม่.....	17
3.2.1 ยูสเคสไดอะแกรม (Usecase Diagram).....	17
3.2.2 คลาสไดอะแกรม (Class Diagram).....	28

สารบัญ (ต่อ)

หน้า

3.2.3 ซีควেনซ์ไดอะแกรม (Sequence Diagram).....	30
3.2.4 แผนภาพความสัมพันธ์ระหว่างเอนทิตี (E-R Diagram)	37
3.2.5 พจนานุกรมข้อมูล	39
บทที่ 4 การออกแบบหน้าจอการทำงาน	59
4.1 การออกแบบหน้าจอ.....	59
4.2 การออกแบบหน้าจอการจัดการข้อมูลของผู้ดูแลระบบ	61
4.2.1 ข้อมูลภาษาโปรแกรม.....	61
4.2.2 ข้อมูลประเภทธุรกิจ	62
4.2.3 ข้อมูลลูกค้า	64
4.2.4 ข้อมูลผู้ใช้งานระบบ	65
4.3 การออกแบบหน้าจอการปรับปรุงพารามิเตอร์	66
4.3.1 การปรับปรุงค่าของแบบจำลอง COCOMO II	66
4.3.2 การปรับปรุงค่าของแบบจำลองฟังก์ชันพอยต์	69
4.3.3 การปรับปรุงค่าของแบบจำลองอ็อบเจกต์พอยต์	70
4.4 การออกแบบหน้าจอการบันทึกข้อมูลโครงการ	70
4.4.1 การบันทึกข้อมูลโครงการ	70
4.4.2 รายละเอียดโครงการ	71
4.4.3 การประเมินภาระงาน.....	72
4.4.4 การคำนวณขนาดของโครงการ โดยวิธีฟังก์ชันพอยต์	72
4.4.5 การคำนวณขนาดของโครงการ โดยวิธีอ็อบเจกต์พอยต์.....	72
4.4.6 การคำนวณค่าปัจจัยที่ส่งผลต่อขนาดโครงการ	74
4.4.7 การคำนวณค่าปัจจัยที่ส่งผลต่อต้นทุนของโครงการ	74
4.4.8 การบันทึกสภาพแวดล้อมของโครงการ	74
4.4.9 การบันทึกข้อมูลผู้มีส่วนร่วมในโครงการ	74
4.4.10 การบันทึกประเด็นที่เกิดขึ้นในโครงการ	76
4.4.11 การบันทึกเอกสารแนบของโครงการ.....	77

สารบัญ (ต่อ)

หน้า

4.5 การออกแบบหน้าจอรายงาน	77
4.5.1 รายงานผลการดำเนินโครงการ	78
4.5.2 รายงานสรุปปัจจัยที่มีผลในการดำเนินงาน	78
4.5.3 รายงานสรุปผลการประเมินภาระงาน	78
4.5.4 รายงานเปรียบเทียบการประเมินภาระงานต่อขนาดของโครงการ	79
4.5.5 รายงานเปรียบเทียบการประเมินระยะเวลาต่อขนาดของโครงการ	80
บทที่ 5 บทสรุป	81
5.1 สรุปโครงการพัฒนาระบบงาน	81
5.2 ผลการดำเนินการพัฒนาระบบ	81
5.3 ข้อจำกัดและข้อเสนอแนะ	82
บรรณานุกรม	83
ประวัติผู้เขียน	84

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงสัญลักษณ์ที่ปรากฏในยูสเคสไคอะแกรม	6
2.2 ตารางแสดงสัญลักษณ์ที่ปรากฏในคลาสไคอะแกรม.....	7
3.1 รายละเอียดของการจัดการข้อมูลผู้ใช้ระบบ	18
3.2 รายละเอียดของการจัดการข้อมูลตั้งต้นของระบบ	20
3.3 รายละเอียดของการปรับปรุงค่าพารามิเตอร์ของแบบจำลอง.....	21
3.4 รายละเอียดของการจัดการข้อมูลโครงการ	22
3.5 รายละเอียดของการประเมินภาระงาน	23
3.6 รายละเอียดของการคำนวณขนาดโครงการ.....	24
3.7 รายละเอียดของการจัดการข้อมูลโครงการแก่นอกระบบ	26
3.8 รายละเอียดของการเรียกดูรายงาน.....	27
3.9 T_DEVELOPMENT_LANGUAGE ข้อมูลภาษาโปรแกรมที่ใช้ในการพัฒนาระบบ	39
3.10 T_BUSINESS_TYPE_LKUP ข้อมูลประเภทธุรกิจของโครงการ	39
3.11 T_CUSTOMER ข้อมูลลูกค้าที่ว่าจะจ้างให้พัฒนาซอฟต์แวร์.....	39
3.12 T_USER ข้อมูลผู้ใช้ระบบ	40
3.13 T_POSITION ข้อมูลตำแหน่งงาน	40
3.14 T_FUNCTION_POINT ข้อมูลค่าน้ำหนักของฟังก์ชันพอยต์ของแบบจำลอง ในการคำนวณหาขนาดของโครงการ	41
3.15 T_OBJECT_POINT ข้อมูลค่าน้ำหนักของอ็อบเจกต์พอยต์ของแบบจำลองในการ คำนวณหาขนาดของโครงการ.....	42
3.16 T_SCALE_DRIVER ข้อมูลแรงขับเคลื่อนด้านขนาดของโครงการ	43
3.17 T_COST_DRIVER ปัจจัยขับเคลื่อนด้านต้นทุนที่แปรผันต่อขนาดของโครงการ	44
3.18 T_PROJECT ข้อมูลของโครงการพัฒนาซอฟต์แวร์ที่นำมาประเมินภาระงาน	47
3.19 T_PROJECT_CONTACT ข้อมูลของผู้ที่มีส่วนร่วมในโครงการ	48
3.20 T_PROJECT_ENVIRONMENT ข้อมูลสภาพแวดล้อมที่ใช้ในโครงการ.....	49
3.21 T_PROJECT_ATTACHMENT ข้อมูลเอกสารที่แนบเป็นข้อมูลประกอบในโครงการ.....	49
3.22 T_PROJECT_ISSUE ข้อมูลประเด็นที่เกิดขึ้นในการดำเนินโครงการ.....	50
3.23 T_PROJECT_ISSUE_COMMENT ข้อมูลคำอธิบายประเด็นที่เกิดขึ้นในโครงการ	51

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.24 T_PROJECT_FUNCTION_POINT ข้อมูลค่าฟังก์ชันพอยต์ของโครงการ	51
3.25 T_PROJECT_OBJECT_POINT ข้อมูลค่าอีอบเจกต์พอยต์ของโครงการ	53
3.26 T_PHASE_SCALE_FACTOR ข้อมูลมาตราส่วนของปัจจัยที่มีผลต่อโครงการ (Scale Factor).....	54
3.27 T_PROJECT_COST_DRIVER ข้อมูลปัจจัยขับเคลื่อนด้านต้นทุนที่แปรผันต่อขนาด ของโครงการ	55
3.28 T_ESTIMATION_RESULT ข้อมูลผลการประเมินภาระงานของโครงการ	58



สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างการเขียนโปรแกรมโดยใช้ไฮเบอร์เนทและ Annotation	8
2.2 ภาพจำลองการทำงานของ COCOMO II.....	10
2.3 คำนวณน้ำหนักในแต่ละประเภทฟังก์ชันของฟังก์ชันพอยต์	13
2.4 คำนวณน้ำหนักในแต่ละประเภทอีอบเจกต์ของอีอบเจกต์พอยต์	13
3.1 ยูสเคสไดอะแกรม	18
3.2 คลาสไดอะแกรม.....	28
3.3 ซีเควนซ์ไดอะแกรมของการเพิ่มข้อมูลภาษาโปรแกรม	31
3.4 ซีเควนซ์ไดอะแกรมของการแก้ไขข้อมูลภาษาโปรแกรม	31
3.5 ซีเควนซ์ไดอะแกรมของการเพิ่มประเภทธุรกิจ.....	32
3.6 ซีเควนซ์ไดอะแกรมของการแก้ไขข้อมูลประเภทธุรกิจ	32
3.7 ซีเควนซ์ไดอะแกรมของการเพิ่มและแก้ไขข้อมูลผู้ใช้ระบบ.....	33
3.8 ซีเควนซ์ไดอะแกรมของการปรับปรุงพารามิเตอร์ของ COCOMO II	34
3.9 ซีเควนซ์ไดอะแกรมของการปรับปรุงพารามิเตอร์ของฟังก์ชันพอยต์	34
3.10 ซีเควนซ์ไดอะแกรมของการบันทึกข้อมูลโครงการ	35
3.11 ซีเควนซ์ไดอะแกรมของการคำนวณขนาดโครงการ	36
3.12 ซีเควนซ์ไดอะแกรมของการประมาณภาระงาน	36
3.13 แผนผังแสดงความสัมพันธ์ระหว่างเอนทิตี.....	38
4.1 หน้าจอล็อกอินเข้าสู่ระบบ	60
4.2 เมนูการทำงานของระบบ	60
4.3 รายการข้อมูลภาษาโปรแกรมในระบบ	61
4.4 บันทึกข้อมูลภาษาโปรแกรม	62
4.5 แก้ไขข้อมูลภาษาโปรแกรม	62
4.6 รายการประเภทธุรกิจ	63
4.7 เพิ่มข้อมูลประเภทธุรกิจ	63
4.8 แก้ไขข้อมูลประเภทธุรกิจ	63
4.9 รายการข้อมูลลูกค้า	64
4.10 เพิ่มข้อมูลลูกค้า.....	64

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.11 รายการข้อมูลผู้ใช้ระบบ.....	65
4.12 เพิ่มข้อมูลผู้ใช้ระบบ.....	65
4.13 แก้ไขข้อมูลผู้ใช้ระบบ.....	66
4.14 หน้าจอปรับปรุงค่าพารามิเตอร์ที่ใช้ในแบบจำลอง COCOMO II.....	67
4.15 หน้าจอการปรับปรุงค่า Scale Factor.....	68
4.16 หน้าจอแสดงรายการ Cost Driver.....	69
4.17 หน้าจอปรับปรุงค่าน้ำหนักของฟังก์ชันพอยต์.....	69
4.18 หน้าจอปรับปรุงค่าน้ำหนักของอ็อบเจกต์พอยต์.....	70
4.19 หน้าจอบันทึกข้อมูล โครงการ.....	71
4.20 หน้าจอแสดงรายละเอียด โครงการ.....	71
4.21 หน้าจอการประเมินภาระงาน.....	72
4.22 หน้าจอการคำนวณขนาดของ โครงการด้วยวิธีฟังก์ชันพอยต์.....	73
4.23 หน้าจอการคำนวณขนาดของ โครงการด้วยวิธีอ็อบเจกต์พอยต์.....	73
4.24 หน้าจอการคำนวณค่าปัจจัยที่ส่งผลกระทบต่อขนาดของ โครงการ.....	74
4.25 หน้าจอการคำนวณค่าปัจจัยที่ส่งผลกระทบต่อต้นทุนของ โครงการ.....	75
4.26 หน้าจอบันทึกสภาพแวดล้อมของ โครงการ.....	76
4.27 หน้าจอบันทึกข้อมูลผู้มีส่วนร่วมในโครงการ.....	76
4.28 หน้าจอบันทึกประเด็นที่เกิดขึ้นในโครงการ.....	77
4.29 หน้าจอบันทึกเอกสารแนบของโครงการ.....	77
4.30 ตัวอย่างรายงานผลการดำเนิน โครงการ.....	78
4.31 ตัวอย่างรายงานสรุปปัจจัยที่มีผลในการดำเนินงาน.....	78
4.32 ตัวอย่างรายงานสรุปผลการประเมินภาระงาน.....	79
4.33 ตัวอย่างรายงานเปรียบเทียบการประเมินภาระงานต่อขนาดของ โครงการ.....	79
4.34 ตัวอย่างรายงานเปรียบเทียบการประเมินระยะเวลาต่อขนาดของโครงการ.....	80

บทที่ 1

บทนำ

1.1 ความเป็นมา

จากการที่บริษัทรับพัฒนาซอฟต์แวร์แห่งหนึ่ง ซึ่งดำเนินธุรกิจทางด้านการให้คำปรึกษาด้านระบบสารสนเทศ และ พัฒนาซอฟต์แวร์เพื่อตอบสนองการขยายตัวของหน่วยงานทั้งภาครัฐบาลและบริษัทเอกชน โดยในปัจจุบันบริษัทรับพัฒนาซอฟต์แวร์ดังกล่าวได้เติบโตขึ้นอย่างรวดเร็ว มีผู้ว่าจ้างพัฒนาซอฟต์แวร์เพิ่มขึ้นเป็นจำนวนมาก ทางบริษัทจึงมีความต้องการพัฒนาระบบสารสนเทศเพื่อช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ในแต่ละโครงการ เพื่อให้การบริหารโครงการมีประสิทธิภาพสูงขึ้น รวมถึงทำให้ลูกค้ามีความพึงพอใจต่อโครงการและมีความเชื่อถือในบริษัทเพิ่มมากขึ้น

ปัญหาที่พบมากในการพัฒนาซอฟต์แวร์คือ การที่ผู้พัฒนาซอฟต์แวร์ไม่สามารถส่งมอบงานได้ภายในระยะเวลาที่กำหนด หรือส่งภายในระยะเวลาที่กำหนดแต่ซอฟต์แวร์ไม่มีคุณภาพ และไม่ปฏิบัติตามความต้องการของผู้ใช้ระบบ ส่งผลให้ผู้ใช้งานเกิดความไม่มั่นใจต่อซอฟต์แวร์ที่พัฒนาขึ้นรวมถึงมีความไม่มั่นใจในบริษัทผู้พัฒนาซอฟต์แวร์ นอกจากนั้นการส่งมอบงานไม่ทันตามกำหนดเวลา ยังทำให้ค่าใช้จ่ายของโครงการเพิ่มสูงขึ้นตามระยะเวลาที่เกินกำหนด ซึ่งสาเหตุของปัญหาดังกล่าวอาจเกิดขึ้นได้จากหลายปัจจัยด้วยกัน เช่น ความเข้าใจความต้องการของระบบคลาดเคลื่อนจากผู้ให้ข้อมูล การพัฒนาระบบโดยใช้เทคโนโลยีที่ผู้พัฒนาไม่ชำนาญ หรือ ความซับซ้อนทางด้านธุรกิจของลูกค้า การที่จะลดปัญหาดังกล่าวผู้ทำหน้าที่ในการวางแผนโครงการจึงควรที่จะสามารถประเมินขนาดของโครงการให้สอดคล้องและใกล้เคียงกับความเป็นจริงมากที่สุด เพื่อให้ได้ระยะเวลาดำเนินโครงการที่เหมาะสม

ในปัจจุบันการประมาณภาระงานหรือการประเมินขนาดของโครงการนั้น บริษัทใช้ประสบการณ์ของผู้บริหารโครงการเป็นเครื่องมือในการประมาณภาระงาน โดยผู้บริหารโครงการจะได้รับรายละเอียดเบื้องต้นของโครงการจากฝ่ายขาย เอกสารสัญญาว่าจ้าง ข้อตกลงในการพัฒนาโครงการ (TOR) รวมถึงการเข้าไปพบผู้ประสานงานโครงการของผู้ว่าจ้าง เพื่อรวบรวมข้อมูลแล้วนำมาประเมินขนาดของโครงการ

จากลักษณะการทำงานดังกล่าวพบว่าโดยส่วนใหญ่มีการประมาณภาระงานต่ำกว่าความเป็นจริงในหลายโครงการ เพื่อที่จะแก้ไขปัญหาดังกล่าวนี้ บริษัทรับพัฒนาซอฟต์แวร์ดังกล่าว จึงมีความต้องการที่จะพัฒนาระบบสารสนเทศเพื่อใช้สำหรับช่วยประมาณภาระงานที่ใช้ในโครงการพัฒนาซอฟต์แวร์ เพื่อรองรับการบริหารโครงการให้สามารถประเมินระยะเวลาในการพัฒนา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ รวมถึงสามารถประเมินจำนวนบุคลากรและค่าใช้จ่ายที่จะเกิดขึ้นในการพัฒนาซอฟต์แวร์ได้อย่างมีประสิทธิภาพยิ่งขึ้น

สำหรับการพัฒนาระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นถูกกำหนดให้พัฒนาขึ้นในลักษณะของเว็บแอปพลิเคชัน ด้วยภาษาโปรแกรม JAVA และ JSP ที่ทำงานบนฐานข้อมูล MySQL และได้เลือกใช้วิธีการประมาณภาระงานที่ใช้แบบจำลองทางคณิตศาสตร์ (Algorithmic/Parametric Model) เพื่อคำนวณหาระยะเวลาที่ต้องใช้ในโครงการ ซึ่งจากการพิจารณาแล้วทางบริษัทเลือกที่จะใช้แบบจำลอง COCOMO II เป็นเครื่องมือในการช่วยประมาณภาระงาน

1.2 วัตถุประสงค์ในการพัฒนาระบบ

วัตถุประสงค์ของการวิเคราะห์และออกแบบระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์มี ดังนี้

1. เพื่อช่วยในการประมาณภาระงานหรือขนาดของโครงการในการพัฒนาซอฟต์แวร์ เพื่อให้การบริหารโครงการเป็นไปอย่างมีประสิทธิภาพมากขึ้น
2. เพื่อช่วยลดปัญหาการส่งมอบงานเกินกำหนดจากการประเมินระยะเวลาที่ต้องใช้ในโครงการผิดพลาด
3. เพื่อกำหนดมาตรฐานการประมาณภาระงานของบริษัท
4. เพื่อใช้ในการควบคุมบุคลากรที่ต้องใช้ในโครงการ รวมถึงควบคุมค่าใช้จ่ายในการพัฒนาซอฟต์แวร์
5. เพื่อช่วยในการวิเคราะห์และวางแผนการบริหารโครงการ รวมถึงแนวทางการดำเนินงานของบริษัท

1.3 ขอบเขตของโครงการ

ในการพัฒนาโครงการระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นี้ได้ศึกษาวิเคราะห์ และพัฒนาระบบขึ้น โดยตั้งอยู่บนสมมติฐานของบริษัทรับพัฒนาซอฟต์แวร์แห่งหนึ่ง ซึ่งจะมีขอบเขตการพัฒนาระบบให้ระบบมีความสามารถครอบคลุมการทำงาน ดังนี้

1. ระบบที่พัฒนาขึ้นสามารถบันทึกและแก้ไขข้อมูลรายละเอียดของโครงการ รวมถึงสามารถบันทึกประเด็นต่างๆที่เกิดขึ้นในการดำเนินงาน และแนบไฟล์เอกสารที่เกี่ยวข้องกับโครงการเพื่อเก็บเป็นข้อมูลสำหรับอ้างอิงในการทำงานได้
2. ระบบที่พัฒนาขึ้นสามารถนำเข้าข้อมูลของโครงการเก่าที่อยู่นอกระบบมาบันทึกเข้าสู่ระบบเพื่อใช้เป็นข้อมูลเชิงสถิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ระบบที่พัฒนาขึ้นสามารถประมาณค่าภาระงานในรูปแบบของระยะเวลาของโครงการ และ ค่าใช้จ่ายที่จะเกิดขึ้น ตามแบบจำลองที่กำหนด (COCOMO II) ได้

4. ระบบที่พัฒนาขึ้นสามารถเพิ่ม แก้ไข หรือ ลบข้อมูลตั้งต้นของระบบได้ เช่น ข้อมูลประเภทของโครงการ ข้อมูลประเภทธุรกิจของลูกค้า หรือ ภาษาโปรแกรมที่ใช้ในการพัฒนาซอฟต์แวร์ เป็นต้น

5. เพื่อให้การประมาณการมีความใกล้เคียงกับความเป็นจริงมากขึ้น ระบบที่พัฒนาขึ้นจะต้องสามารถปรับปรุงค่าพารามิเตอร์ที่ใช้ในแบบจำลองที่ใช้สำหรับประเมินภาระงานของโครงการได้ และระบบจะต้องเก็บประวัติของการปรับปรุงค่าพารามิเตอร์ทุกครั้งที่มีการปรับปรุงข้อมูล

6. ระบบสามารถสร้างรายงานแสดงข้อมูลของโครงการ หรือ ข้อมูลสถิติอื่นๆ เพื่อใช้ในการบริหารโครงการ และ สนับสนุนการทำงานของบริษัทได้

1.4 ขั้นตอนในการดำเนินโครงการ

ขั้นตอนในการดำเนินโครงการพัฒนาระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ มีขั้นตอน ดังนี้

1. กำหนดหัวข้อในการศึกษา เป้าหมายของการศึกษา ขอบเขต วัตถุประสงค์ และ ผลที่คาดว่าจะได้รับ

2. ศึกษาขั้นตอนการทำงานของทางบริษัทและสำรวจความต้องการของระบบใหม่หรือระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ จากผู้บริหาร โครงการที่ปฏิบัติงานอยู่ในปัจจุบัน

3. วิเคราะห์ถึงปัญหาและความต้องการของระบบจากข้อมูลที่ได้มา เพื่อหาแนวทางการพัฒนาระบบให้ตอบสนองต่อความต้องการของผู้ใช้และให้เกิดประโยชน์ต่อการทำงานสูงสุด

4. ศึกษาเทคโนโลยี แบบจำลอง และทฤษฎีต่างๆที่เกี่ยวข้อง เพื่อเลือกเทคโนโลยีที่เหมาะสมในการนำมาประยุกต์ใช้กับระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์

5. ออกแบบระบบงาน ฐานข้อมูล และหน้าจอให้เหมาะสมสำหรับการใช้งานของผู้ใช้ระบบ

6. พัฒนาระบบตามที่ได้ออกแบบไว้ และทดสอบระบบร่วมกับผู้บริหารโครงการของบริษัท รวมถึงนำระบบไปทดลองใช้งานจริง

7. สรุปผลจากข้อคิดเห็นและคำแนะนำในการทดสอบและทดลองใช้งานระบบ ตลอดจนจัดทำเอกสารประกอบการพัฒนาระบบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการพัฒนาระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ มีดังนี้

1. ช่วยในการบริหารโครงการด้านการประมาณระยะเวลาและค่าใช้จ่ายที่ใช้ในโครงการให้มีประสิทธิภาพมากขึ้น
2. ช่วยลดปัญหาการส่งมอบงานล่าช้า
3. ช่วยควบคุมทรัพยากร และ ค่าใช้จ่ายที่จะเกิดขึ้นของโครงการ
4. ช่วยกำหนดทิศทางการทำงานประมาณภาระงานของบริษัทให้เป็นไปในทิศทางเดียวกัน
5. สร้างความน่าเชื่อถือในการประมาณขนาดของโครงการสำหรับนำเสนอต่อผู้ว่าจ้าง



บทที่ 2

เทคโนโลยีที่เกี่ยวข้อง

ในการพัฒนาโครงการระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นี้ ได้ใช้ ทฤษฎีที่เกี่ยวข้องหลายเรื่องด้วยกัน ทั้งการศึกษาถึงเทคโนโลยีและเครื่องมือที่นำมาใช้ในการ วิเคราะห์และออกแบบ รวมถึงการพัฒนาระบบงาน ดังนั้นในบทนี้จึงขอกกล่าวถึงรายละเอียดของ เทคโนโลยีที่ใช้ในโครงการ ซึ่งมีรายละเอียด ดังนี้

2.1 ยูเอ็มแอล (UML)

ปัจจุบันแนวคิดของการพัฒนาซอฟต์แวร์กำลังเปลี่ยนไปเป็นการพัฒนาในรูปแบบของ อ็อบเจกต์ โอเรียนเต็ด จึงเป็นที่มาของการพัฒนาแบบจำลองขึ้นเพื่อใช้สำหรับสนับสนุนการพัฒนา ซอฟต์แวร์แบบอ็อบเจกต์ โดยที่ UML (Object Management Group, 2007) เป็นภาษาที่ใช้ในการ กำหนด (Specifying) แสดงภาพ (Visualizing) สร้าง และ บันทึกสิ่งต่างๆที่เกิดขึ้นจากการพัฒนา ซอฟต์แวร์ รวมทั้งยังมีความสามารถในการแสดงภาพจำลองการทำงานเชิงธุรกิจ หรือ ระบบอื่นๆที่ ไม่ใช่การทำงานของซอฟต์แวร์ได้อีกด้วย ซึ่งการใช้แบบจำลองเหล่านี้จะสามารถช่วยให้การ ออกแบบและพัฒนาระบบขนาดใหญ่ที่มีความซับซ้อน สามารถทำได้อย่างมีประสิทธิภาพภายใต้ เงื่อนไขของระยะเวลาและงบประมาณที่กำหนดไว้

ในโครงการนี้ได้นำแบบจำลองของ UML มาใช้ในการวิเคราะห์และออกแบบระบบ จึง ขออธิบายแบบจำลองที่ใช้งาน 2 แบบจำลอง ดังนี้

2.1.1 ยูสเคสไดอะแกรม (Use Case Diagram)



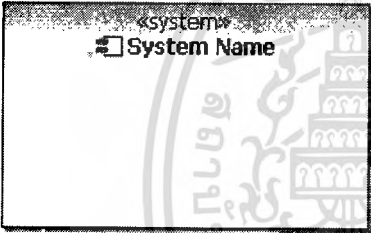
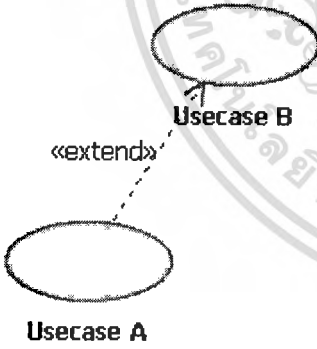
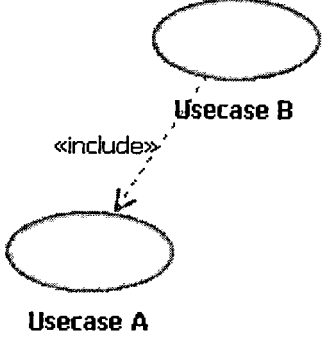
เป็นแผนภาพที่แสดงขอบเขตการทำงานของระบบ และความต้องการของผู้ใช้งานระบบ โดยระบุผู้ใช้งานระบบ (Actor) กิจกรรมที่มีในระบบ (Usecase) และขอบเขตของระบบ (System Boundary) ที่มาจากความต้องการของผู้ใช้งานระบบ โดยยูสเคสไดอะแกรม มีวัตถุประสงค์หลัก เพื่อระบุขอบเขต ส่วนประกอบและหน้าที่ของระบบ สัญลักษณ์ที่ใช้การแสดงยูสเคสไดอะแกรมมี ลักษณะ ดังตารางที่ 2.1

2.1.2 คลาสไดอะแกรม (Class Diagram)

เป็นแผนภาพที่แสดงถึงคลาสทั้งหมดที่มีอยู่ในระบบรวมถึงความสัมพันธ์ของคลาส โดย แต่ละคลาสจะประกอบไปด้วยแอททริบิวต์ (Attribute) ที่เป็นการบ่งชี้คุณลักษณะของคลาส และ

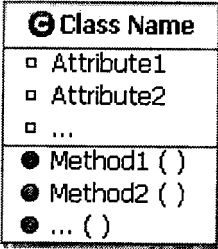
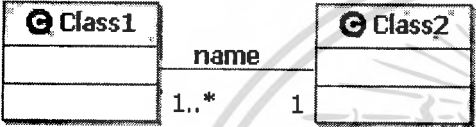
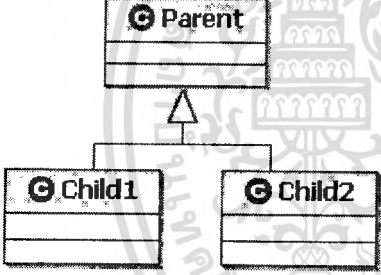
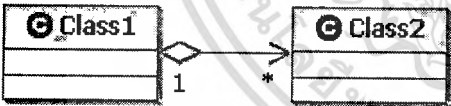
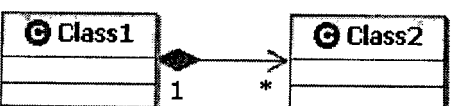
โอเปอเรชัน (Operation) หรือ เมธอด (Method) ที่บอกพฤติกรรมการทำงานของคลาส สัญลักษณ์ที่ใช้แสดงคลาสที่ปรากฏอยู่ในคลาสไดอะแกรมมีลักษณะ ดังตารางที่ 2.2

ตารางที่ 2.1 ตารางแสดงสัญลักษณ์ที่ปรากฏในยูสเคสไดอะแกรม

สัญลักษณ์	ชื่อเรียก	คำอธิบาย
 Actor	Actor	แสดงถึงผู้ใช้งาน หรือกลุ่มผู้ใช้งานที่เกี่ยวข้องข้องกับระบบ ซึ่งจะแสดงชื่อของบทบาทของผู้ใช้งานอยู่ภายใต้สัญลักษณ์
 Usecase Name	Use Case	แสดงถึงฟังก์ชันการทำงานของระบบ
	System Boundary	แสดงถึงขอบเขตของระบบ
	Extend	แสดงความสัมพันธ์ระหว่าง Use Case แบบมีทางเลือก (Optional) หัวลูกศรจะชี้ไปยัง Use Case ที่เป็น Use Case หลัก เช่น Use Case A extends → Use Case B หมายความว่า Use Case A อาจถูกเรียกทำงานโดย Use Case B หรือไม่ก็ได้
	Include	แสดงความสัมพันธ์ระหว่าง Use Case แบบ Use Case หนึ่งเรียกใช้โดยอีก Use Case หนึ่ง หัวลูกศรจะชี้ไปที่ Use Case ที่ถูกเรียกใช้ เช่น Use Case B include → Use Case A หมายความว่า เมื่อ Use Case B ทำงานจะมีการเรียกใช้ Use Case A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ตารางแสดงสัญลักษณ์ที่ปรากฏในคลาสไดอะแกรม

สัญลักษณ์	ชื่อเรียก	คำอธิบาย
	Class	แสดงถึงคลาส และสมาชิกภายในคลาส โดยแบ่งการแสดงผลเป็นสามส่วน ส่วนแรกแสดงถึงชื่อคลาส ส่วนที่สองแสดงถึงแอทริบิวต์ภายในคลาส และส่วนสุดท้าย แสดงถึงเมธอดที่เป็นการทำงานของคลาสนั้น
	Association Relationship	แสดงถึงความสัมพันธ์ระหว่างคลาส โดยมี Multiplicity แสดงจำนวน Instance ของคลาสหนึ่งที่เกี่ยวข้องกับอีกคลาสหนึ่ง
	Generalization Relationship	แสดงถึงความสัมพันธ์ระหว่างคลาสแบบสืบทอดคุณสมบัติ โดยจะมีคุณสมบัติเหมือนกับคลาสหลัก (Super class) และสามารถเพิ่มเติมเป็นของคลาสตัวเองได้
	Aggregation Relationship	แสดงถึงความสัมพันธ์ของการเป็นส่วนประกอบของอีกคลาสหนึ่ง แต่ไม่ขึ้นต่อกัน โดยจะเรียกคลาสหลักว่า Whole และเรียกส่วนประกอบว่า Part โดยแยกการทำงานเป็นอิสระจากกัน
	Composition Relationship	แสดงถึงความสัมพันธ์ของคลาสหนึ่งที่มีส่วนประกอบเป็นอีกคลาสหนึ่ง โดยจะเรียกคลาสหลักว่า Whole และเรียกส่วนประกอบว่า Part เช่น Class1 มี Class2 เป็นส่วนประกอบ เมื่อ Class1 ถูกลบ Class2 จะถูกลบตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่สามารถเผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไฮเบอร์เนท (Hibernate)

ในการพัฒนาซอฟต์แวร์ด้วยหลักของอ็อบเจกต์นั้น มีความพยายามที่จะมองส่วนประกอบต่างๆในการพัฒนาซอฟต์แวร์ให้อยู่ในรูปของวัตถุ ไม่เว้นแม้กระทั่งฐานข้อมูลเอง โดยมีแนวคิดที่จะนำฐานข้อมูลที่มีโครงสร้างข้อมูลอยู่ในรูปแบบของตาราง (Table) และข้อมูลที่เก็บในฐานข้อมูลเป็นเรคคอร์ด (Record) มาอยู่ในรูปแบบของอ็อบเจกต์ คือ ให้คลาสทำหน้าที่แทนตารางในฐานข้อมูล โดยเป็นตัวแสดงรายละเอียดของตาราง และ Instance ของคลาสทำหน้าที่แทนข้อมูลแต่ละเรคคอร์ด ซึ่งแนวคิดดังกล่าวเรียกว่า Object-Relational Mapping (ORM)

Hibernate (Red Hat Middleware, 2008) เป็นเครื่องมือ ORM ชนิดหนึ่งเพื่อช่วยให้ผู้พัฒนาระบบสามารถพัฒนาระบบได้สะดวกยิ่งขึ้น โดยการนำโครงสร้างของฐานข้อมูลมานำเสนออยู่ในรูปแบบของอ็อบเจกต์ และเป็นเครื่องมือสำหรับจัดการกับการทำงานของฐานข้อมูลหรืออาจกล่าวได้ว่าไฮเบอร์เนทเป็นตัวกลางระหว่างซอฟต์แวร์ที่พัฒนาขึ้น และ ฐานข้อมูลที่ใช้ในการเก็บข้อมูล

สำหรับโครงการนี้ได้ใช้ Hibernate ร่วมกับเทคโนโลยี Java Persistence API (JPA) ซึ่งเป็นมาตรฐานที่บริษัท Sun Micro System กำหนดคุณลักษณะ (Specification) ขึ้นเพื่อใช้งานร่วมกับ Java เวอร์ชัน 5.0 ที่เพิ่มความสามารถในการกำหนด Annotation ในการสร้างคลาส หรือเมธอด ดังนั้น Hibernate จึงนำข้อดีของ Annotation มาใช้เพิ่มความสามารถในการทำงาน โดยผู้พัฒนาซอฟต์แวร์สามารถระบุ Annotation ของตาราง หรือ แอทริบิวต์ ได้ในจาวาได้เลย โดยไม่ต้องไปกำหนดค่าเพิ่มเติมเหมือนในรุ่นก่อนหน้านี้ เช่น ตัวอย่างในรูปที่ 2.1

```
import javax.persistence.*;

@Entity
@Table ( name="TB_USER" )
public class User {

    @Id GeneratedValue
    @Column ( name="USER_ID" )
    private Long user_id;

    @Column ( name="USER_NAME" )
    private String user_name;

    /** Setter methods */
    /** Getter methods */
}
```

รูปที่ 2.1 ตัวอย่างการเขียน โปรแกรมโดยใช้ไฮเบอร์เนทและ Annotation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างเป็นการทำการ Mapping ตาราง TB_USER เข้ากับคลาส User โดยการกำหนด Annotation @Table และ Mapping คอลัมน์ในตาราง TB_USER เข้ากับ แอทริบิวต์ของคลาส User ด้วย @Column โดยมีคุณสมบัติอื่นเพิ่มเติมเช่น @Id GeneratedValue เป็นการระบุให้ Hibernate รู้ว่าคอลัมน์ USER_ID นั้นเป็นคอลัมน์ที่เป็นไพรมารีคีย์ (PK) และที่ค่าของ USER_ID จะเพิ่มขึ้นโดยอัตโนมัติเมื่อมีการเพิ่มข้อมูลในตาราง TB_USER

2.3 แบบจำลอง COCOMO II

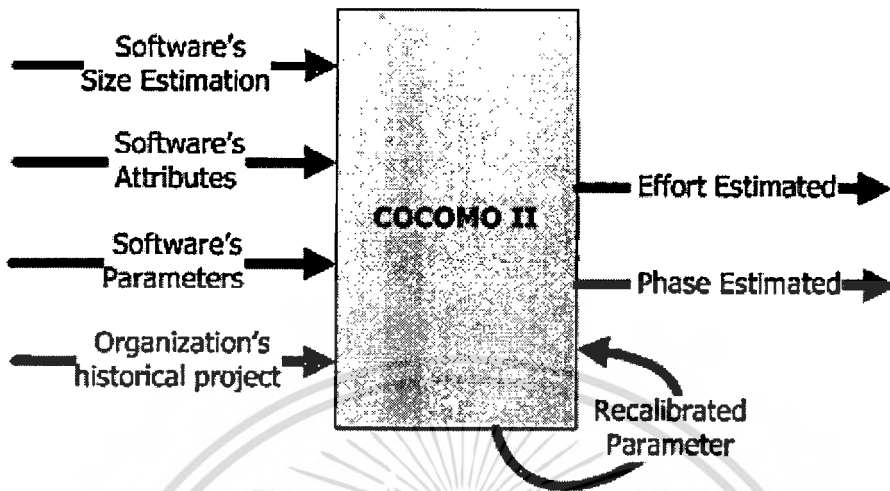
การประมาณภาระงาน เป็นกระบวนการที่ทำขึ้นเพื่อประเมินระยะเวลาในการพัฒนาโครงการ รวมถึงประมาณทรัพยากรที่ต้องใช้ในโครงการ ซึ่งถือเป็นจุดเริ่มต้นที่สำคัญในการพัฒนาซอฟต์แวร์ โดยการประมาณภาระงานนั้นสามารถแบ่งกลุ่มของการประมาณค่าภาระงานออกเป็น 3 กลุ่ม ดังนี้

1. การใช้แบบจำลองทางคณิตศาสตร์เพื่อคำนวณหาระยะเวลาที่ต้องใช้ในโครงการ (Algorithmic/Parametric Model)
2. การตัดสินใจประเมินระยะเวลาโดยผู้เชี่ยวชาญ (Expert Judgment)
3. การวิเคราะห์โดยเปรียบเทียบกับข้อมูลโครงการเก่า (Analogy Model)

สำหรับในโครงการนี้จะเลือกใช้วิธีการประมาณภาระงานโดยใช้แบบจำลองทางคณิตศาสตร์ซึ่งเป็นวิธีการที่ได้รับความนิยมและยอมรับใช้งานกันอย่างแพร่หลาย และมีความเหมาะสมกับลักษณะการทำงานของบริษัทมากที่สุด เนื่องจากการประมาณภาระงานลักษณะเดิมของบริษัท เป็นการประมาณการโดยผู้บริหารโครงการ ซึ่งถือว่ายังมีความคลาดเคลื่อนอยู่มาก และทางบริษัทเองก็ยังไม่มีการจัดเก็บข้อมูลโครงการเก่าที่พัฒนาแล้วมากพอที่จะนำมาประมาณโดยการเปรียบเทียบ ดังนั้น การเลือกใช้วิธีการประมาณทางคณิตศาสตร์จึงเป็นวิธีที่เหมาะสมที่สุดสำหรับโครงการนี้ โดยการประมาณภาระงานโดยใช้แบบจำลองทางคณิตศาสตร์เองก็มีการพัฒนาแบบจำลองขึ้นมาเป็นจำนวนมาก เช่น COCOMO II (Boehm, 1997) SLIM (Kirk, 2000) หรือ SEER (Galorath, 2008) เป็นต้น แต่สำหรับการวิเคราะห์และออกแบบระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นี้เลือกใช้แบบจำลอง COCOMO II เป็นเครื่องมือในการประมาณภาระงาน

COCOMO (Constructive Cost Model) เป็นแบบจำลองที่ใช้สำหรับประมาณภาระงานหรือขนาดของโครงการในการพัฒนาซอฟต์แวร์ ถูกพัฒนาขึ้นโดย Dr. Barry Boehm (Boehm, 1997) ตั้งแต่ปีค.ศ. 1981 และพัฒนาต่อเนื่องมาจนถึงปัจจุบันคือ COCOMO II แนวความคิดในการพัฒนาแบบจำลองนั้น ก็เพื่อใช้ประเมินระยะเวลาในการพัฒนาซอฟต์แวร์ โดย Dr. Barry Boehm ได้รวบรวมข้อมูลของโครงการต่างๆ และนำเอาความแตกต่างของแต่ละโครงการ ลักษณะเฉพาะ

ของโครงการ หรือ ปัจจัยอื่นๆ มาคำนวณออกมาเป็นตัวเลขทางคณิตศาสตร์และสถิติ เพื่อสร้างเป็นแบบจำลองสำหรับประมาณค่าภาระงานในการพัฒนาซอฟต์แวร์



รูปที่ 2.2 ภาพจำลองการทำงานของ COCOMO II

รูปที่ 2.2 เป็นภาพแสดงการทำงานของแบบจำลอง COCOMO II โดยการทำงานจะเริ่มจากการบันทึกข้อมูลต่างๆ ของโครงการ เช่น

- ขนาดของโครงการที่ประมาณไว้ อาจจะใช้วิธีคำนวณจากจำนวนบรรทัดของโปรแกรม หรือ ฟังก์ชันพอยต์ ก็ได้
- คุณลักษณะของโครงการ เช่น ภาษาโปรแกรมที่ใช้พัฒนา ประเภทธุรกิจของลูกค้า ลักษณะของซอฟต์แวร์ เป็นต้น
- พารามิเตอร์ของโครงการ เพื่อที่จะระบุว่าเป็นโครงการที่ต้องพัฒนาขึ้นใหม่ หรือโครงการที่สามารถใช้ของเดิมได้ (Reuse) หรือเป็นการบำรุงรักษาโครงการ (Maintenance)
- ข้อมูลเก่าของโครงการต่างๆ ที่องค์กรเคยพัฒนาขึ้น โดยข้อมูลส่วนนี้จะถูกนำไปใช้ในการเปรียบเทียบค่าคงที่ในสมการ

เมื่อบันทึกข้อมูลต่างๆ ของโครงการเข้าสู่แบบจำลองแล้วก็จะประมาณค่าภาระงานของโครงการตามสมการที่ได้กำหนดไว้ โดย COCOMO II ได้สร้างแบบจำลองสำหรับโครงการพัฒนาซอฟต์แวร์ในแต่ละประเภทไว้ 3 รูปแบบ คือ

1. **Application composition model** ใช้สำหรับการประมาณภาระงานในโครงการที่เป็น การปรับแต่งการทำงาน (Customize) หรือ ในการเริ่มต้นประมาณขนาดของโครงการคร่าวๆ เช่น ขั้นตอนการทำต้นแบบ (Prototype) สำหรับออกแบบส่วนต่อประสานกับผู้ใช้งาน (User Interface)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **Early design stage model** ใช้สำหรับการประมาณขนาดของโครงการในขั้นตอนการเก็บรวบรวมความต้องการของระบบหรือเมื่อโครงการเริ่มมีรูปร่างที่ชัดเจนขึ้น

3. **Post architecture stage model** ใช้สำหรับโครงการที่ทำงานต่อเนื่องหลังจากที่ระบบพัฒนาเสร็จสิ้นแล้ว เช่น โครงการที่อยู่ในช่วงของการบำรุงรักษาระบบ หรือการพัฒนาระบบเพิ่มเติม

แบบจำลอง COCOMO II นั้นมีสมการที่ใช้ในการประมาณภาระงานเป็นหน่วยคนต่อเดือน (Person-Month) โดยมีการกำหนดสมการพื้นฐานเพื่อเป็นการประมาณค่าภาระงานเบื้องต้น ซึ่งสมการที่ใช้ในการประมาณค่าภาระงานนี้จะปรับเปลี่ยนไปตามแบบจำลองของแต่ละเฟสของการพัฒนาซอฟต์แวร์ โดยสมการพื้นฐานในการประมาณค่าภาระงาน และสมการสำหรับหาระยะเวลาในการพัฒนาซอฟต์แวร์ มีรูปแบบดังนี้

$$PM = A * (Size)^B * EM \quad (2.1)$$

$$TDEV = [3.0 * PM^{(0.33+0.2*(B-1.01))}] * SCED \quad (2.2)$$

โดยตัวแปรต่างๆมีความหมาย ดังนี้

- PM** คือ ค่าประมาณภาระงานมีหน่วยเป็นคนต่อเดือน (Person-Month)
- A** คือ ค่าคงที่ในการประมาณภาระงาน โดยถูกกำหนดให้มีค่าเท่ากับ 2.5
- Size** คือ ขนาดของโครงการ โดยใช้ค่าจำนวนบรรทัดของโปรแกรม (KSLOC - Kilo Source Line of Code) ที่คำนวณได้จากวิธีการคิดฟังก์ชันพอยต์ (McConnell, 2006) ควบคู่กับ อีอบเจกต์พอยต์ (Karaka, 1998)
- EM** ผลคูณของปัจจัยขับเคลื่อนด้านต้นทุน (Effort Multiplier) ประกอบด้วย 17 ปัจจัย ได้แก่
- ความเชื่อถือได้ของซอฟต์แวร์ (Product Reliability: RELY)
 - ขนาดของฐานข้อมูล (Database Size: DATA)
 - ความซับซ้อนของผลิตภัณฑ์ (Product Complexity: CPLX)
 - ความต้องการนำซอฟต์แวร์มาใช้ใหม่ (Reused required: RUSE)
 - ความต้องการเอกสารในกระบวนการพัฒนาซอฟต์แวร์ (Documentation Match to Life Cycle: DOCU)
 - ข้อจำกัดด้านระยะเวลาในการประมวลผล (Execution Time Constraint: TIME)
 - ข้อจำกัดด้านพื้นที่ที่ใช้ในการเก็บข้อมูล (Main Storage Constraint: STOR)
 - ความถี่ในการเปลี่ยนแพลตฟอร์ม (Platform Volatility: PVOL)
 - ความสามารถของนักวิเคราะห์ระบบ (Analyst Capability: ACAP)
 - ความสามารถของนักพัฒนาระบบ (Programmer Capability: PCAP)

- ประสบการณ์ในการใช้โปรแกรมประยุกต์ (Application Experience: AEXP)
- ประสบการณ์ด้านแพลตฟอร์ม (Platform Experience: PEXP)
- ประสบการณ์ด้านภาษาและเครื่องมือ (Language and Tools Experience: LTEX)
- ความต่อเนื่องในการทำงานของบุคลากร (Personnel Continuity: PCON)
- บัณฑิตด้านการใช้เครื่องมือในการทำงาน (Use of Software Tools: TOOL)
- การพัฒนาโปรแกรมในหลายที่ตั้ง (Multisite Development: SITE)
- เงื่อนไขระยะเวลาในการพัฒนาระบบ (Required Development Schedule: SCED)

TDEV คือ ค่าประมาณระยะเวลาที่ใช้ในการพัฒนาซอฟต์แวร์ในหน่วยเดือน

B คือ มาตรการส่วนของปัจจัยที่มีผลต่อโครงการ (Scale Factor) โดยคำนวณได้จากสมการ

$$B = 1.01 + 0.01 * \sum W \quad (2.3)$$

โดย W คือค่าน้ำหนักของตัวแปร 5 ตัว ได้แก่

- การมีแบบอย่างมาก่อน (Precedentedness)
- ความยืดหยุ่นของซอฟต์แวร์ (Flexibility)
- ความเสี่ยงของโครงการ (Architecture/Risk resolution)
- ปัจจัยการทำงานร่วมกัน (Team Cohesion)
- วุฒิภาวะในการควบคุมกระบวนการพัฒนา (Process Maturity)

SCED คือ อัตราส่วนร้อยละของปัจจัยที่เกี่ยวกับความเร่งรัดในหมายกำหนดการงานของโครงการ (Required Development Schedule)

การหาขนาดของโครงการด้วยวิธีฟังก์ชันพอยต์

ฟังก์ชันพอยต์ คือการหาขนาดของ โครงการวิธีหนึ่งที่ไม่มุ่งเน้นถึงจำนวนบรรทัดของซอฟต์แวร์ และไม่ขึ้นกับภาษาโปรแกรมใดๆ แต่จะใช้วิธีการนับจำนวนของฟังก์ชันที่มีในโครงการ คุณเข้ากับค่าน้ำหนักของระดับความยากในฟังก์ชันนั้นๆ โดยฟังก์ชันพอยต์แบ่งประเภทของฟังก์ชันเป็น 5 รูปแบบ ดังนี้

- ฟังก์ชันสำหรับการนำข้อมูลเข้าสู่ระบบ (External Input)
- ฟังก์ชันสำหรับการส่งข้อมูลการประมวลผลไปยังระบบอื่น (External Output)
- ฟังก์ชันการทำงานที่เกี่ยวข้องกับเพิ่มข้อมูลในระบบ (Internal Logical File)
- ฟังก์ชันการทำงานที่เกี่ยวข้องกับเพิ่มข้อมูลที่อยู่นอกระบบ (External Interface File)
- ฟังก์ชันสำหรับการแสดงผลข้อมูลที่ผ่านการประมวลผลในระบบ (External Inquiry)

ในการหาขนาดของโครงการจะนำจำนวนของฟังก์ชันที่ได้วิเคราะห์คูณเข้ากับค่านำหนักของแต่ละประเภทฟังก์ชัน โดยกำหนดค่านำหนักของแต่ละประเภทฟังก์ชัน ดังนี้

	Simple	Average	Complex
External Input	3	4	6
External Output	4	5	7
Internal Logical File	7	10	15
External Interface File	5	7	10
External Inquiry	3	4	6

รูปที่ 2.3 ค่านำหนักในแต่ละประเภทฟังก์ชันของฟังก์ชันพอยต์

เมื่อนำจำนวนของฟังก์ชันคูณเข้ากับค่านำหนักของแต่ละประเภทฟังก์ชันตามระดับความซับซ้อนของฟังก์ชันแล้ว ก็จะได้เป็นขนาดของซอฟต์แวร์ในหน่วยฟังก์ชันพอยต์ (FP) โดยในการนำขนาดของซอฟต์แวร์ที่คำนวณได้ไปเปรียบเทียบกับค่าเฉลี่ยของจำนวนบรรทัดในการพัฒนาซอฟต์แวร์ด้วยภาษาโปรแกรมที่ใช้ในโครงการ จะทำให้ได้ขนาดของโครงการในหน่วยบรรทัด (Source Line of Code) และนำค่าขนาดของโครงการในหน่วยบรรทัดไปใช้ในแบบจำลอง COCOMO II เพื่อใช้ในการประมาณการงานต่อไป

การหาขนาดของโครงการด้วยวิธีอ็อบเจกต์พอยต์

อ็อบเจกต์พอยต์ คือการหาขนาดของโครงการวิธีหนึ่งที่ไม่มุ่งเน้นถึงจำนวนบรรทัดของซอฟต์แวร์ และไม่ขึ้นกับภาษาโปรแกรมใดๆ แต่จะใช้วิธีการนับจำนวนของอ็อบเจกต์ที่มีในโครงการ คูณกับค่านำหนักของระดับความยากในฟังก์ชันนั้นๆ โดยอ็อบเจกต์แบ่งประเภทของอ็อบเจกต์เป็น 3 รูปแบบ ดังนี้

- หน้าจอการแสดงผล (Screen Display)
- รายงานที่สร้างขึ้นในระบบ (Report)
- ส่วนประกอบในภาษาโปรแกรมรุ่นที่สาม (3GL Component)

ในการหาขนาดของโครงการจะนำจำนวนของอ็อบเจกต์ที่ได้วิเคราะห์คูณเข้ากับค่านำหนักของแต่ละประเภทอ็อบเจกต์ โดยกำหนดค่านำหนักของแต่ละประเภทอ็อบเจกต์ ดังนี้

	Simple	Average	Complex
Screen Display	1	2	3
Report	2	5	8
3GL Component	10	10	10

รูปที่ 2.4 การกำหนดค่านำหนักในแต่ละประเภทอ็อบเจกต์ของอ็อบเจกต์พอยต์

เมื่อนำจำนวนของอ็อบเจกต์คูณเข้ากับค่าน้ำหนักของแต่ละประเภทอ็อบเจกต์ตามระดับความซับซ้อนของอ็อบเจกต์แล้ว ก็จะได้เป็นขนาดของซอฟต์แวร์ในหน่วยอ็อบเจกต์พอยต์ (OP) โดยในการนำขนาดของซอฟต์แวร์ที่คำนวณได้ไปเปรียบเทียบกับค่าเฉลี่ยของจำนวนบรรทัดในการพัฒนาซอฟต์แวร์ด้วยภาษาโปรแกรมที่ใช้ในโครงการ จะทำให้ได้ขนาดของโครงการในหน่วยบรรทัด (Source Line of Code) และนำค่าขนาดของโครงการในหน่วยบรรทัดไปใช้ในแบบจำลอง COCOMO II เพื่อใช้ในการประมาณภาระงานต่อไป

ตัวอย่างในการประเมินภาระงานด้วยแบบจำลอง COCOMO II

- สมมติให้การนับขนาดของโครงการด้วยวิธีฟังก์ชันพอยต์ของโครงการที่นำมาประเมินภาระงานมีค่าเท่ากับ 80 ฟังก์ชันพอยต์
- ผลคูณของจำนวนบรรทัดต่อฟังก์ชันพอยต์มีค่าเท่ากับ 60 บรรทัดต่อฟังก์ชันพอยต์
- ผลรวมค่าน้ำหนักของปัจจัยในโครงการมีค่าเท่ากับ 14.3 จะได้ค่า B คือ 1.153 จากการคำนวณจาก $B = 1.01 + 0.01 * 14.3$
- ดังนั้นขนาดของโครงการที่ประเมินภาระงานจะมีค่าเท่ากับ

$$PM = 2.5 * [(80*60)/1000]^{1.153}$$

$$= 15.254 PM$$

สมการที่ได้กล่าวมานั้นเป็นสมการพื้นฐานของแบบจำลอง COCOMO II ที่ใช้สำหรับการประมาณภาระงานเบื้องต้น ซึ่งในแบบจำลองของ COCOMO II นั้นก็จะมีการประยุกต์สมการดังกล่าวโดยนำปัจจัยอื่นที่มีผลกระทบต่อโครงการไปใช้ในสมการ เพื่อให้ค่าการประมาณภาระงานมีความน่าเชื่อถือมากขึ้นอีกด้วย และเมื่อได้คำนวณค่าประมาณภาระงาน และระยะเวลาในการพัฒนาซอฟต์แวร์แล้ว ผู้บริหารโครงการจะพิจารณาข้อมูลตามผลการประมาณที่ได้ เพื่อนำไปบริหารทรัพยากรที่จะใช้ในโครงการพัฒนาซอฟต์แวร์นั้นๆ ต่อไป

บทที่ 3

การวิเคราะห์และออกแบบระบบ

จากปัญหาในการบริหาร โครงการของบริษัทที่ใช้เป็นกรณีศึกษาในด้านการประมาณการงานหรือการประเมินขนาดของโครงการ ทำให้เกิดแนวคิดในการพัฒนาระบบสารสนเทศที่จะช่วยในการประมาณการงานในการพัฒนาซอฟต์แวร์ของแต่ละโครงการขึ้น หลังจากที่ได้ทราบถึงปัญหาและศึกษาความเป็นไปได้ของระบบงานแล้ว จึงได้มีการวิเคราะห์และออกแบบระบบงานดังกล่าว โดยมีรายละเอียดของการวิเคราะห์และออกแบบระบบ ดังนี้

3.1 การศึกษาระบบงานปัจจุบัน

การประมาณการงานในปัจจุบันของบริษัทซึ่งรับพัฒนาซอฟต์แวร์ดังกล่าว นั้น เป็นการประมาณขนาดของโครงการ โดยอาศัยประสบการณ์ของผู้บริหาร โครงการเป็นเครื่องมือในการประมาณการงาน ซึ่งผู้บริหารโครงการจะได้รับรายละเอียดในเบื้องต้นของโครงการจากฝ่ายขาย เอกสารสัญญาว่าจ้าง ข้อตกลงในการพัฒนาโครงการ (TOR) รวมถึงการเข้าไปพบผู้ประสานงานโครงการของผู้ว่าจ้าง เพื่อรวบรวมข้อมูลแล้วนำมาประเมินขนาดของโครงการ และในการประเมินขนาดของโครงการแต่ละโครงการ ยังอาจจะประเมินโดยผู้บริหารโครงการที่แตกต่างกัน ทำให้ไม่มีมาตรฐานในการประมาณการงาน อีกทั้งการขาดระบบจัดเก็บข้อมูลการประมาณ รวมถึงรายละเอียดอื่นๆ ของโครงการ จึงไม่สามารถนำข้อมูลการประมาณการงานของโครงการเก่ามาปรับปรุงเปรียบเทียบกับโครงการใหม่ที่ต้องการประมาณได้ นอกจากนี้ในบางโครงการยังเกิดปัญหาที่ผู้ว่าจ้างไม่มั่นใจในระยะเวลาที่ผู้บริหารโครงการประเมินไว้ เนื่องจากไม่สามารถแสดงรายละเอียดของวิธีการประมาณการได้ ดังนั้นจากการศึกษาระบบการทำงานในปัจจุบัน สามารถสรุปถึงรูปแบบการทำงาน และปัญหาที่พบ ได้ดังนี้

3.1.1 รูปแบบการทำงานปัจจุบัน

การประมาณการงานหรือขนาดของโครงการในปัจจุบันนั้น สามารถสรุปขั้นตอนการดำเนินงานได้ ดังนี้

1. ฝ่ายขายได้รับข้อตกลงในการพัฒนาโครงการ (TOR) หรือ เอกสารรายละเอียดของโครงการ จากบริษัทที่ต้องการว่าจ้างให้พัฒนาซอฟต์แวร์ หรือหน่วยงานราชการที่บริษัทต้องการเข้าไปประมูลงาน

2. ฝ่ายขายนำเอกสารที่ได้มาส่งต่อให้กับผู้บริหารโครงการ เพื่อให้ผู้บริหารโครงการประเมินระยะเวลาและค่าใช้จ่ายของโครงการ โดยผู้บริหารโครงการจะศึกษารายละเอียดของ

โครงการจากเอกสารที่ได้ หรือได้เข้าพบผู้ประสานงานของโครงการเพื่อรวบรวมความต้องการของระบบในปัจจุบัน

3. ผู้บริหารโครงการประมาณภาระงานของโครงการและค่าใช้จ่ายเพื่อให้ฝ่ายขายจัดทำใบเสนอราคาพร้อมกับเอกสารอื่นๆเพื่อส่งกลับไปให้กับผู้ว่าจ้าง

4. ฝ่ายขายยื่นใบเสนอราคาเพื่อขอรับพัฒนาโครงการ และเมื่อได้รับการเลือกจากผู้ว่าจ้างให้พัฒนาโครงการได้ ผู้บริหารโครงการจึงจะจัดสรรบุคลากรเพื่อดำเนินการพัฒนาโครงการนั้นๆ ตามกระบวนการพัฒนาซอฟต์แวร์จนโครงการเสร็จสิ้น

จากรูปแบบการทำงานในปัจจุบันของบริษัท สามารถวิเคราะห์และสรุปถึงปัญหาที่เกิดขึ้นได้ ดังต่อไปนี้

3.1.2 ปัญหาของระบบการทำงานปัจจุบัน

จากการวิเคราะห์รูปแบบการทำงานของระบบงานเดิมตามที่ได้กล่าวไว้ข้างต้นนั้น สามารถสรุปถึงปัญหาที่เกิดขึ้นจากการทำงานได้ ดังนี้

1. ผู้บริหารโครงการได้รับเวลาน้อยเกินไปในการทำความเข้าใจกับโครงการที่ประเมิน เนื่องจากขั้นตอนการประมาณขนาดของโครงการเพื่อยื่นข้อเสนอพัฒนาโครงการนั้นเกิดขึ้นก่อนจะมีการรวบรวมความต้องการของระบบ

2. บางโครงการ ผู้บริหารโครงการขาดความคำนึงถึงทักษะความสามารถในการทำงานของผู้พัฒนาระบบในโครงการ ว่ามีความเชี่ยวชาญและเหมาะสมกับโครงการที่ได้รับมาหรือไม่

3. บุคลากรในทีมพัฒนาของโครงการขาดความเชี่ยวชาญในเครื่องมือหรือภาษาโปรแกรมที่ใช้ในโครงการ

4. การประมาณภาระงานในแต่ละโครงการขึ้นอยู่กับประสบการณ์ของผู้บริหารโครงการแต่ละคน ทำให้ไม่มีมาตรฐานในการประมาณภาระงานของบริษัท

5. ผู้ว่าจ้างเกิดความไม่เชื่อถือผลการประเมินขนาดของโครงการ โดยเข้าใจว่าเป็นการประเมินขนาดเกินจริง ทำให้ขาดความมั่นใจในโครงการ และเกิดทัศนคติที่ไม่ดีกับซอฟต์แวร์ที่พัฒนาขึ้น

6. ใช้เวลาในการพัฒนาโครงการมากเกินกว่าที่ประมาณการ ทำให้ส่งมอบโครงการไม่ทันตามกำหนด มีค่าใช้จ่ายในส่วน of ค่าปรับเกิดขึ้นในบางโครงการ ส่งผลให้ค่าใช้จ่ายของการพัฒนาโครงการโดยรวมสูงขึ้นตามไปด้วย

7. พนักงานต้องทำงานหนักขึ้น เพื่อเร่งให้สามารถส่งมอบโครงการได้ ส่งผลให้มีการลาออกจากงาน ทำให้การพัฒนาซอฟต์แวร์ในโครงการติดขัด และเมื่อนำพนักงานใหม่เข้ามาทำงานแทน ก็ต้องใช้เวลาในการเรียนรู้การทำงาน of โครงการเพิ่มเติมอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การวิเคราะห์และออกแบบระบบงานใหม่

จากปัญหาที่พบจากการศึกษาการทำงานเดิมของระบบงานในปัจจุบันนั้น นำไปสู่ความต้องการที่จะพัฒนาระบบสารสนเทศเพื่อช่วยในการประมาณภาระงานในการพัฒนาซอฟต์แวร์สำหรับนำมาช่วยในการบริหาร โครงการให้เป็นอย่างดีและมีประสิทธิภาพมากยิ่งขึ้น โดยในการออกแบบระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นได้ใช้ยูเอ็มแอล (UML) เป็นเครื่องมือในการออกแบบระบบ และใช้แผนภาพความสัมพันธ์ระหว่างเอนทิตี (E-R Diagram) ในการออกแบบฐานข้อมูล ซึ่งมีแบบจำลองที่นำเสนอ ดังนี้

- ยูสเคสไดอะแกรม (Usecase Diagram)
- รายละเอียดยูสเคส (Usecase Description)
- คลาสไดอะแกรม (Class Diagram)
- ซีควเอนซ์ไดอะแกรม (Sequence Diagram)
- แผนภาพความสัมพันธ์ระหว่างเอนทิตี (E-R Diagram)

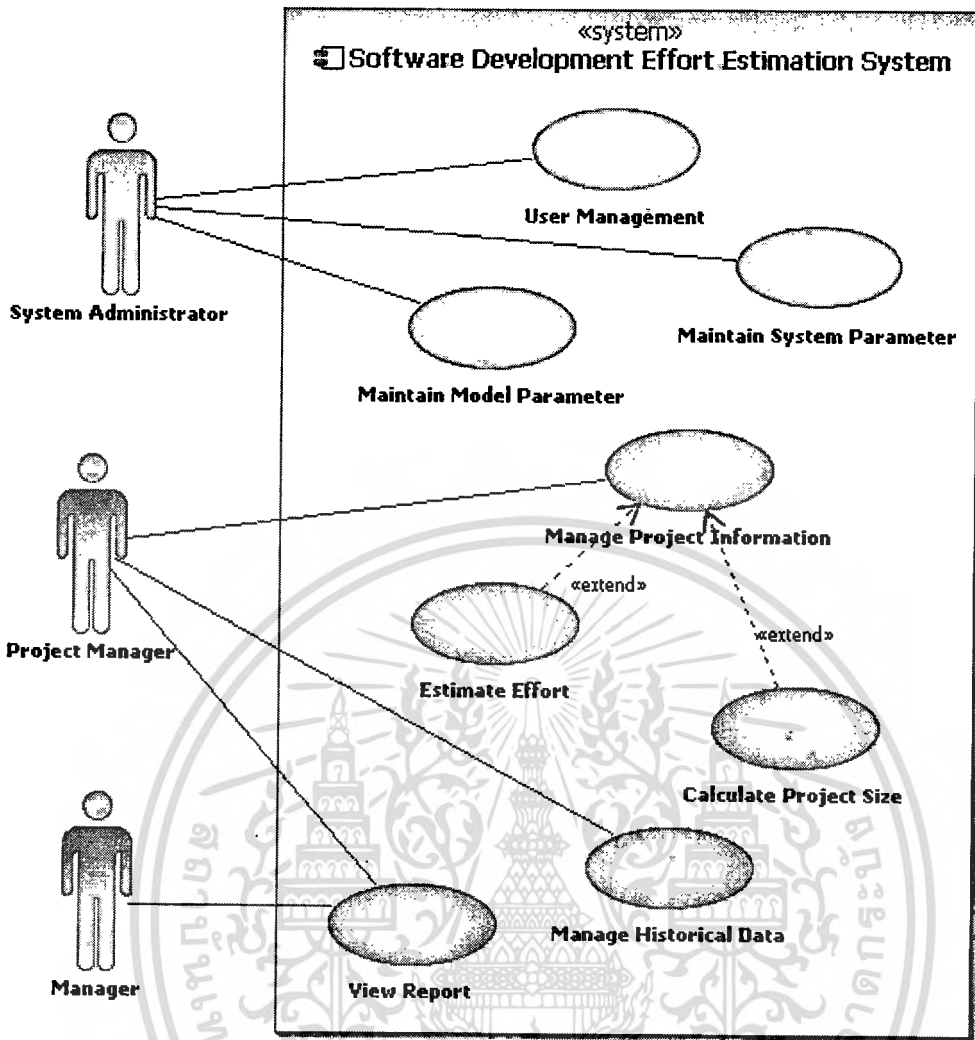
3.2.1 ยูสเคสไดอะแกรม (Usecase Diagram)

ยูสเคสไดอะแกรม เป็นเครื่องมือที่ใช้แสดงขอบเขตการทำงานของระบบ และการปฏิสัมพันธ์ระหว่างแอกเตอร์และฟังก์ชันการทำงานของระบบ โดยยูสเคสไดอะแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้น แสดงได้ดังรูปที่ 3.1

ยูสเคสไดอะแกรมดังที่ปรากฏในรูปที่ 3.1 สามารถอธิบายได้ถึงผู้ใช้ระบบที่มีส่วนร่วมกับระบบได้ทั้งสิ้น 3 แอกเตอร์ ดังนี้

1. **System Administrator** หรือ ผู้ดูแลระบบมีหน้าที่ในการจัดการข้อมูลที่เป็นข้อมูลตั้งต้นในระบบ และรับผิดชอบการปรับปรุงค่าพารามิเตอร์ที่ใช้ในแบบจำลองในการประมาณค่าภาระงานให้เหมาะสมและใกล้เคียงกับความเป็นจริงมากที่สุด
2. **Project Manager** หรือ ผู้บริหารโครงการมีหน้าที่ในการบันทึกข้อมูลของโครงการที่ต้องการประมาณค่าภาระงาน รวมถึงการบันทึกข้อมูลโครงการเก่าขององค์กรที่อยู่นอกระบบเข้ามาเป็นข้อมูลเชิงสถิติในการทำงานของระบบ และสามารถเรียกดูรายงานต่างๆที่สร้างขึ้นจากระบบได้
3. **Manager** หรือ ผู้บริหารของบริษัทที่มีส่วนร่วมกับระบบในการเรียกดูรายงานต่างๆที่ระบบสร้างขึ้น เพื่อวางแผนการดำเนินงานของบริษัทต่อไป

จากยูสเคสไดอะแกรมใน รูปที่ 3.1 ได้แสดงถึงฟังก์ชันการทำงานของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ ได้ทั้งสิ้น 8 ยูสเคส โดยอธิบายรายละเอียดของแต่ละฟังก์ชัน ได้โดยมีรายละเอียดยูสเคส (Usecase Description) ดังนี้



รูปที่ 3.1 ยูสเคสไดอะแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์

ตารางที่ 3.1 รายละเอียดของการจัดการข้อมูลผู้ใช้ระบบ

หมายเลขยูสเคส :	Usecase001
ชื่อยูสเคส :	User Management
รายละเอียดโดยย่อ :	ฟังก์ชันสำหรับจัดการข้อมูลของผู้ใช้งานระบบ เพื่อบันทึกรายละเอียดของผู้ใช้งานระบบ รวมทั้งกำหนดสิทธิในการใช้งาน ซึ่งผู้ใช้งานระบบอาจจะมีสิทธิในการเข้าถึงข้อมูลได้แตกต่างกัน โดยการจัดการข้อมูลผู้ใช้งานระบบในที่นี้ครอบคลุมถึงการเพิ่มผู้ใช้งานระบบ การแก้ไขรายละเอียดของผู้ใช้งานระบบ การระงับสิทธิผู้ใช้งานระบบ และการยกเลิกการระงับสิทธิผู้ใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 (ต่อ)

ทริกเกอร์ :	เมื่อมีความต้องการเพิ่มผู้ใช้งานระบบ การแก้ไขรายละเอียดของผู้ใช้งานระบบ การระงับสิทธิผู้ใช้งานระบบ และการยกเลิกการระงับสิทธิผู้ใช้งานระบบ	
ผู้ใช้งานระบบ :	System Administrator (ผู้ดูแลระบบ)	
เงื่อนไขขั้นต้น :	ผู้ใช้งานระบบที่เข้าใช้งานฟังก์ชันนี้ได้จะต้องเป็นผู้ใช้งานที่ถูกกำหนดสิทธิ์ให้เป็นผู้ดูแลระบบ และได้เข้าสู่ระบบแล้วเท่านั้น	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเข้าสู่ระบบและเลือกเมนู “จัดการผู้ใช้งานระบบ”	
		2. ระบบแสดงหน้าจอการจัดการผู้ใช้งานระบบ
	3. ผู้ใช้ระบบเลือกการทำงานดังนี้ a. เพิ่มผู้ใช้งาน b. แก้ไขข้อมูลผู้ใช้งาน	
		4. ระบบแสดงหน้าจอการทำงานตามที่ผู้ใช้ระบบเลือก
	5. ผู้ใช้ระบบระบุข้อมูลและตั้งบันทึกข้อมูล	
		6. ระบบตรวจสอบความถูกต้องของข้อมูล
		7. บันทึกข้อมูลเข้าสู่ระบบและแสดงข้อความแจ้งผลการบันทึกข้อมูล
เงื่อนไขหลังจบงาน :	ข้อมูลของผู้ใช้งานระบบถูกบันทึกเข้าสู่ระบบ ตามที่ผู้ดูแลระบบระบุข้อมูล	
กรณีผิดปกติ :	<ol style="list-style-type: none"> กรณีที่เพิ่มผู้ใช้ระบบซ้ำกับข้อมูลที่มีอยู่ในระบบ ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ กรณีที่ระบุข้อมูลไม่ครบตามที่ระบบต้องการ ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังขั้นตอนที่ 5 	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 รายละเอียดของการจัดการข้อมูลตั้งต้นของระบบ

หมายเลขยูสเคส :	Usecase002	
ชื่อยูสเคส :	Maintain System Parameter	
รายละเอียดโดยย่อ :	เป็นการทำงานสำหรับผู้ดูแลระบบเข้ามาใช้งานเพื่อจัดการกับข้อมูลตั้งต้นต่างๆ ของระบบ เช่น ข้อมูลประเภทโครงการ ประเภทธุรกิจของลูกค้า หรือ ภาษาโปรแกรมที่ใช้ เป็นต้น โดยการจัดการข้อมูลตั้งต้นของระบบ ในที่นี้ครอบคลุมถึงการเพิ่มข้อมูล การแก้ไขข้อมูล และการลบข้อมูล	
ทริกเกอร์ :	เมื่อมีความต้องการเพิ่ม แก้ไข หรือลบข้อมูลตั้งต้นต่างๆของระบบ	
ผู้ใช้งานระบบ :	System Administrator (ผู้ดูแลระบบ)	
เงื่อนไขขั้นต้น :	ผู้ใช้งานระบบที่เข้าใช้งานฟังก์ชันนี้ได้จะต้องเป็นผู้ใช้งานที่ถูกกำหนดสิทธิ์ให้เป็นผู้ดูแลระบบ และ เข้าสู่ระบบแล้วเท่านั้น	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเข้าสู่ระบบและเลือกเมนูการจัดการข้อมูลกลางที่ต้องการทำงาน	2. ระบบแสดงหน้าจอการจัดการข้อมูลตั้งต้นตามที่ได้เลือก
	3. ผู้ใช้ระบบเลือกการทำงาน a. เพิ่มข้อมูล b. แก้ไขข้อมูล c. ลบข้อมูล	4. ระบบแสดงหน้าจอการทำงานตามที่ผู้ใช้ระบบเลือก
	5. ผู้ใช้ระบบระบุข้อมูลและตั้งบันทึกข้อมูล	6. ระบบตรวจสอบความถูกต้องของข้อมูล
		7. บันทึกข้อมูลเข้าสู่ระบบและแสดงข้อความแจ้งผลการบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 (ต่อ)

เงื่อนไขหลังจบงาน :	ข้อมูลค่าตั้งต้นของระบบถูกบันทึกเข้าสู่ระบบ ตามที่ผู้ดูแลระบบบันทึก สำหรับกรณีลบข้อมูล ระบบจะไม่ลบข้อมูลออกจากฐานข้อมูลจริงแต่จะระบุค่าอ้างอิงไว้ว่าเป็นข้อมูลที่ถูกลบไปแล้ว
กรณีผิดปกติ :	1. กรณีที่ระบุข้อมูลไม่ครบตามที่ระบบต้องการ ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังขั้นตอนที่ 5

ตารางที่ 3.3 รายละเอียดของการปรับปรุงค่าพารามิเตอร์ของแบบจำลอง

หมายเลขยูสเคส :	Usecase003	
ชื่อยูสเคส :	Maintain Model Parameter	
รายละเอียดโดยย่อ :	เป็นการทำงานสำหรับผู้ดูแลระบบเพื่อปรับปรุงค่าพารามิเตอร์ต่างๆที่ใช้ในแบบจำลองการประมาณค่าภาระงานของโครงการ ให้มีความเหมาะสมและใกล้เคียงความเป็นจริงมากขึ้น	
ทริกเกอร์ :	เมื่อมีความต้องการปรับปรุงค่าพารามิเตอร์ของแบบจำลองให้มีความเหมาะสมและสอดคล้องกับความเป็นจริงของบริษัทมากขึ้น	
ผู้ใช้งานระบบ :	System Administrator (ผู้ดูแลระบบ)	
เงื่อนไขขั้นต้น :	ผู้ใช้งานที่เข้าใช้งานฟังก์ชันนี้จะต้องเป็นผู้ใช้งานที่ถูกกำหนดสิทธิ์ให้เป็นผู้ดูแลระบบ และ เข้าสู่ระบบแล้วเท่านั้น	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเข้าสู่ระบบและเลือกเมนูการจัดการพารามิเตอร์	2. ระบบแสดงหน้าจอการจัดการพารามิเตอร์ตาม que เลือก
	3. ผู้ใช้ระบบเลือกค่าพารามิเตอร์ที่ต้องการปรับปรุง	
		4. ระบบแสดงหน้าจอการทำงานตาม que ผู้ใช้ระบบเลือก
	5. ผู้ใช้ระบบระบุข้อมูลและตั้งบันทึกข้อมูล	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 (ต่อ)

		6. ระบบตรวจสอบความถูกต้องของข้อมูล
		7. บันทึกข้อมูลเข้าสู่ระบบและแสดงข้อความแจ้งผลการบันทึกข้อมูล
เงื่อนไขหลังจบงาน :	ข้อมูลพารามิเตอร์ถูกบันทึกเข้าสู่ระบบตามที่ผู้ใช้งานได้บันทึก	
กรณีผิดปกติ :	1. กรณีที่ระบุข้อมูลไม่ครบตามที่ระบบต้องการ ระบบจะแสดงข้อความเตือนต่อผู้ใช้งาน และย้อนกลับไปยังขั้นตอนที่ 5	

ตารางที่ 3.4 รายละเอียดของการจัดการข้อมูลโครงการ

หมายเลขยูสเคส :	Usecase004	
ชื่อยูสเคส :	Manage Project Information	
รายละเอียดโดยย่อ :	ฟังก์ชันสำหรับบริหารข้อมูลของโครงการ โดยผู้บริหาร โครงการ เพื่อใช้สำหรับประเมินภาระงาน หรือบันทึกรายละเอียดอื่นๆของโครงการเพื่อใช้เป็นข้อมูลในการบริหารโครงการ โดยผู้บริหารโครงการสามารถเข้าใช้ฟังก์ชันนี้เพื่อประเมินภาระงาน หรือเพื่อบันทึกข้อมูลของโครงการเพียงอย่างเดียวก็ได้	
ทริกเกอร์ :	เมื่อมีความต้องการประเมินภาระงานของโครงการ หรือต้องการบันทึกข้อมูลรายละเอียดของโครงการ	
ผู้ใช้งานระบบ :	Project Manager (ผู้บริหารโครงการ)	
เงื่อนไขขั้นต้น :	ผู้ใช้งานที่เข้าใช้งานฟังก์ชันนี้จะต้องเป็นผู้ใช้งานที่ถูกกำหนดสิทธิ์ให้เป็นผู้บริหารโครงการ และเข้าสู่ระบบแล้วเท่านั้น รวมถึงได้มีการบันทึกข้อมูลตั้งต้นของระบบ และค่าพารามิเตอร์ของระบบแล้ว	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเข้าสู่ระบบและเลือกเมนูจัดการข้อมูลโครงการ	
		2. ระบบแสดงหน้าจอการจัดการข้อมูลโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 (ต่อ)

	3. ผู้ใช้ระบบเลือกการทำงานที่ต้องการ a. เพิ่มข้อมูลโครงการ b. จัดการข้อมูลรายละเอียดโครงการ c. ประเมินภาระงานของโครงการ	
		4. ระบบแสดงหน้าจอการทำงานตามที่ผู้ใช้ระบบเลือก
	5. ผู้ใช้ระบบระบุข้อมูลและตั้งบันทึกข้อมูล	
		6. ระบบตรวจสอบความถูกต้องของข้อมูล
		7. บันทึกข้อมูลเข้าสู่ระบบและแสดงข้อความแจ้งผลการบันทึกข้อมูล
เงื่อนไขหลังจบงาน :	ข้อมูลโครงการถูกบันทึกเข้าสู่ระบบตามที่ผู้ใช้ระบบได้บันทึก	
กรณีผิดปกติ :	1. กรณีที่ระบุข้อมูลไม่ครบตามที่ระบบต้องการ ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังขั้นตอนที่ 5	

ตารางที่ 3.5 รายละเอียดของการประเมินภาระงาน

หมายเลขยูสเคส :	Usecase005
ชื่อยูสเคส :	Estimate Effort
รายละเอียดโดยย่อ :	เป็นฟังก์ชันการทำงานที่อาจจะถูกเรียกเมื่อผู้บริหารโครงการเข้าใช้งานฟังก์ชันการจัดการข้อมูลโครงการ โดยการประเมินภาระงานนั้นจะนำข้อมูลของโครงการที่ผู้บริหารโครงการได้บันทึก เข้าสู่แบบจำลองเพื่อคำนวณค่าภาระงานที่จะต้องใช้ของโครงการนั้นต่อไป
ทริกเกอร์ :	เมื่อผู้บริหารโครงการมีความต้องการประเมินภาระงานของโครงการ ผ่านทางฟังก์ชันการจัดการข้อมูลโครงการ
ผู้ใช้งานระบบ :	Project Manager (ผู้บริหารโครงการ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 (ต่อ)

เงื่อนไขขั้นต้น :	ผู้ใช้ระบบได้บันทึกข้อมูลโครงการที่จะต้องใช้ในการประเมินภาระงาน รวมถึงได้มีการบันทึกข้อมูลตั้งต้นของระบบ และค่าพารามิเตอร์ของระบบแล้ว	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเรียกดูการประเมินภาระงานผ่านฟังก์ชันการจัดการข้อมูลโครงการ	
		2. ระบบตรวจสอบข้อมูลโครงการว่าครบตามที่จะใช้ในการประเมินภาระงานหรือไม่
		3. ระบบนำข้อมูลโครงการเข้าสู่แบบจำลองเพื่อประเมินภาระงานของโครงการ
		4. ระบบแสดงผลการประเมินภาระงาน
เงื่อนไขหลังจบงาน :	ผลการประเมินภาระงานได้ส่งกลับไปให้ฟังก์ชันการจัดการข้อมูลโครงการและแสดงผลให้ผู้ใช้ระบบ	
กรณีผิดปกติ :	1. กรณีที่ข้อมูลโครงการไม่เพียงพอต่อการประเมินภาระงานตามแบบจำลอง ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังหน้าจอการจัดการข้อมูลโครงการ	

ตารางที่ 3.6 รายละเอียดของการคำนวณขนาดโครงการ

หมายเลขยูสเคส :	Usecase006
ชื่อยูสเคส :	Calculate Project Size
รายละเอียดโดยย่อ :	เป็นฟังก์ชันการทำงานที่อาจจะถูกเรียกเมื่อผู้บริหารโครงการเข้าใช้งานฟังก์ชันการจัดการข้อมูลโครงการ โดยการคำนวณขนาดของโครงการนั้นจะนำข้อมูลของโครงการที่ผู้บริหารโครงการได้บันทึก เข้าสู่แบบจำลองเพื่อคำนวณหาขนาดของโครงการและนำไปใช้สำหรับประเมินภาระงานต่อไป

ตารางที่ 3.6 (ต่อ)

ทริกเกอร์ :	เมื่อผู้บริหารโครงการมีความต้องการประเมินภาระงานของโครงการ หรือ ต้องการเรียกดูขนาดของ โครงการ ผ่านทางฟังก์ชันการจัดการข้อมูลโครงการ	
ผู้ใช้งานระบบ :	Project Manager (ผู้บริหารโครงการ)	
เงื่อนไขขั้นต้น :	ผู้ใช้ระบบได้บันทึกข้อมูลโครงการที่จะต้องใช้ในการประเมินภาระงาน รวมถึง ได้มีการบันทึกข้อมูลตั้งต้นของระบบ และ ค่าพารามิเตอร์ของระบบแล้ว	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเรียกดูขนาดของโครงการ ผ่านฟังก์ชันการจัดการข้อมูลโครงการ	2. ระบบตรวจสอบข้อมูลโครงการ ว่าครบตามที่จะต้องใช้ในการคำนวณขนาดของโครงการหรือไม่
		3. ระบบนำข้อมูลโครงการเข้าสู่แบบจำลองเพื่อคำนวณขนาดของโครงการ
		4. ระบบแสดงผลการคำนวณขนาดของโครงการ หรือส่งต่อไปให้ฟังก์ชันการประเมินภาระงาน
เงื่อนไขหลังจบงาน :	ผลการคำนวณขนาดของโครงการได้ส่งกลับไปให้ฟังก์ชันการจัดการข้อมูลโครงการและแสดงผลให้ผู้ใช้ระบบ หรือส่งไปให้ฟังก์ชันการประเมินภาระงานเพื่อเข้าสู่แบบจำลองต่อไป	
กรณีผิดปกติ :	1. กรณีที่ข้อมูลโครงการไม่เพียงพอต่อการคำนวณขนาดของโครงการตามแบบจำลอง ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังหน้าจอการจัดการข้อมูลโครงการ	

ตารางที่ 3.7 รายละเอียดของการจัดการข้อมูลโครงการเก่าในระบบ

หมายเลขยูสเคส :	Usecase007	
ชื่อยูสเคส :	Manage Historical Data	
รายละเอียดโดยย่อ :	เป็นฟังก์ชันการทำงานของผู้บริหารโครงการในการบันทึกข้อมูลของโครงการที่บริษัทเคยได้พัฒนาไว้ก่อนจะได้ใช้ระบบช่วยประมวลผลภาระงานในการพัฒนาซอฟต์แวร์ เข้าสู่ระบบเพื่อบันทึกเป็นข้อมูลสถิติสำหรับใช้อ้างอิงในการประเมินภาระงานต่อไป ซึ่งการจัดการข้อมูลโครงการที่บริษัทได้เคยพัฒนานั้น คือ การบันทึกข้อมูลโครงการเก่าเข้าสู่ระบบ และการแก้ไขข้อมูลโครงการเก่าในระบบ	
ทริกเกอร์ :	เมื่อผู้บริหารโครงการมีความต้องการจัดการข้อมูลโครงการที่บริษัทได้เคยพัฒนานั้น คือ การบันทึกข้อมูลโครงการเก่าเข้าสู่ระบบ และการแก้ไขข้อมูลโครงการเก่าในระบบ	
ผู้ใช้งานระบบ :	Project Manager (ผู้บริหารโครงการ)	
เงื่อนไขขั้นต้น :	ผู้ใช้งานระบบที่เข้าใช้งานฟังก์ชันนี้จะต้องเป็นผู้ใช้งานที่ถูกกำหนดสิทธิ์ให้เป็นผู้บริหารโครงการ และเข้าสู่ระบบแล้วเท่านั้น	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเข้าสู่ระบบและเลือกเมนูการจัดการข้อมูลโครงการเก่า	
		2. ระบบแสดงหน้าจอการจัดการข้อมูลโครงการเก่าตามที่ใช้ระบบเลือก
	3. ผู้ใช้ระบบเลือกการทำงาน a. บันทึกข้อมูลโครงการเก่า b. แก้ไขข้อมูลโครงการเก่า	
		4. ระบบแสดงหน้าจอการทำงานตามที่ใช้ระบบเลือก
	5. ผู้ใช้ระบบระบุข้อมูลและส่งบันทึกข้อมูล	
	6. ระบบตรวจสอบความถูกต้องของข้อมูล	

ตารางที่ 3.7 (ต่อ)

		7. บันทึกข้อมูลเข้าสู่ระบบและแสดงข้อความแจ้งผลการบันทึก
เงื่อนไขหลังจบงาน :	ข้อมูลโครงการเดิมที่ได้พัฒนาแล้วถูกบันทึกเข้าสู่ระบบตามข้อมูลที่ผู้บริหารโครงการบันทึก	
กรณีผิดปกติ :	1. กรณีที่ระบุข้อมูลไม่ครบตามที่ระบบต้องการ ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังขั้นตอนที่ 5	

ตารางที่ 3.8 รายละเอียดของการเรียกดูรายงาน

หมายเลขยูสเคส :	Usecase008	
ชื่อยูสเคส :	View Report	
รายละเอียดโดยย่อ :	เป็นการเรียกดูรายงานของผู้บริหารโครงการและผู้บริหารของบริษัท โดยจะเป็นรายงานต่างๆ ที่สร้างขึ้นจากข้อมูลในระบบเพื่อแสดงสถิติต่างๆ หรือแสดงรายละเอียดการประมาณค่าภาระงานของโครงการเพื่อใช้กำหนดแนวทางในการทำงานของบริษัทต่อไป	
ทริกเกอร์ :	เมื่อผู้บริหารโครงการหรือผู้บริหารของบริษัทมีความต้องการเรียกดูรายงานจากข้อมูลในระบบ เพื่อใช้สำหรับวางแผนการดำเนินงานของบริษัทและการบริหารโครงการต่อไป	
ผู้ใช้งานระบบ :	Project Manager (ผู้บริหารโครงการ), Manager (ผู้บริหาร)	
เงื่อนไขขั้นต้น :	ผู้ใช้ระบบที่เข้าใช้งานฟังก์ชันนี้ได้จะต้องเป็นผู้ใช้งานที่ถูกกำหนดสิทธิ์ให้เป็นผู้บริหารโครงการ และเข้าสู่ระบบแล้วเท่านั้น	
การทำงานปกติ :	แอกเตอร์	ระบบ
	1. ผู้ใช้ระบบเข้าสู่ระบบและเลือกเมนูดูรายงาน	
		2. ระบบแสดงหน้าจอการเงื่อนไขการแสดงผลรายงาน
	3. ผู้ใช้ระบบระบุเงื่อนไขการเรียกดูรายงานและสั่งแสดงผลรายงาน	
		4. ระบบตรวจสอบความถูกต้องของข้อมูล

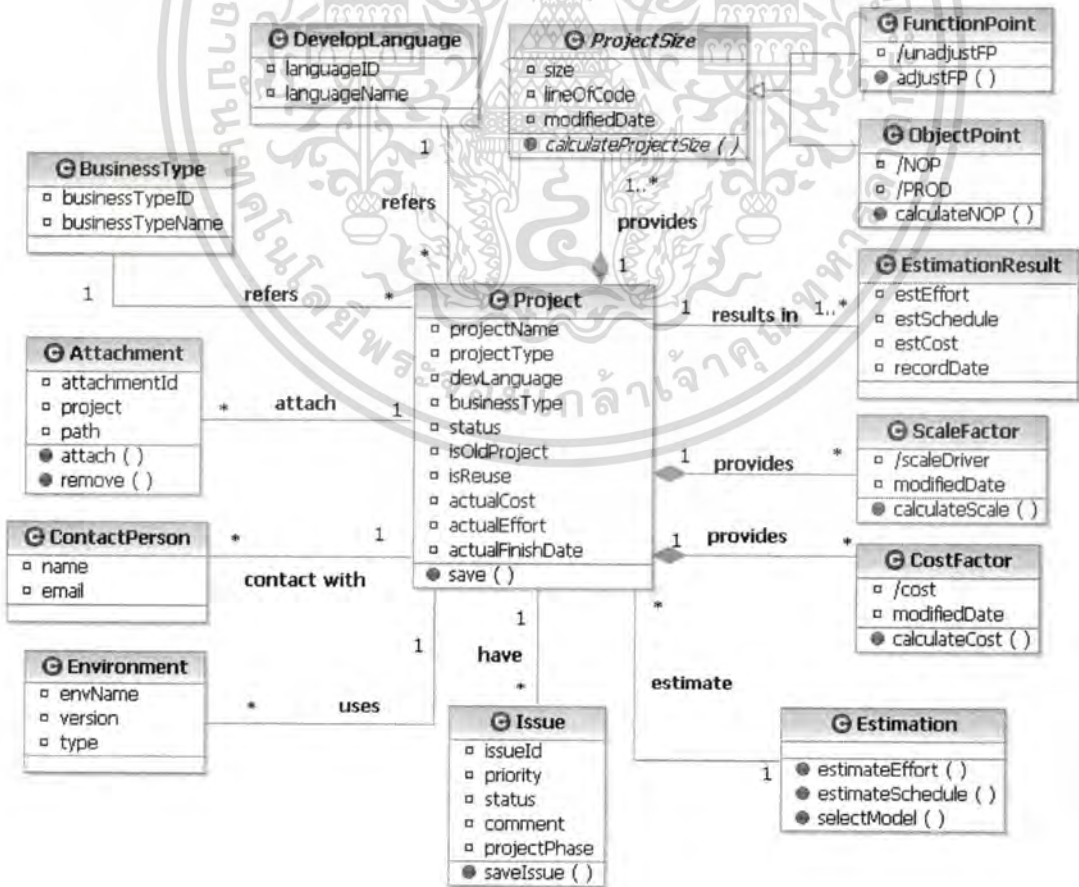
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้เผยแพร่ใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 (ต่อ)

		5. ระบบค้นหาข้อมูลและแสดงรายงานตามเงื่อนไขที่ผู้ใช้ระบบได้ระบุมา
เงื่อนไขหลังจบงาน :	ระบบแสดงผลรายงานตามเงื่อนไขที่ผู้ใช้ระบบได้ระบุ	
กรณีผิดปกติ :	1. กรณีที่ระบุข้อมูลไม่ครบตามที่ระบบต้องการ ระบบจะแสดงข้อความเตือนต่อผู้ใช้ระบบ และย้อนกลับไปยังขั้นตอนที่ 3	

3.2.2 คลาสไดอะแกรม (Class Diagram)

คลาสไดอะแกรม เป็นแผนภาพที่ใช้อธิบายมุมมองเชิงโครงสร้างคงที่ของระบบ โดยรายละเอียดภายในแต่ละคลาสประกอบด้วยคุณลักษณะและพฤติกรรมของคลาส โดยคลาสไดอะแกรมยังแสดงรายละเอียดของความสัมพันธ์ระหว่างคลาส เพื่อเป็นแนวทางในการพัฒนาฐานข้อมูลของระบบ สำหรับคลาสไดอะแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นแสดงได้ ดังนี้



รูปที่ 3.2 คลาสไดอะแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์
 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานภายในเท่านั้น เมื่อผู้ยื่นขอเสนอขอรับใช้ระบบด้านการศึกษา
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสโคแอดแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นประกอบไปด้วยคลาสทั้งหมด 14 คลาส ดังนี้

1. คลาส *Project* เป็นคลาสที่เป็นตัวแทนข้อมูลโครงการในการพัฒนาซอฟต์แวร์ โดยอาจจะมีข้อมูลผู้ติดต่อ (*ContactPerson*) เอกสารแนบ (*Attachment*) สภาพแวดล้อมในการทำงาน (*Environment*) หรือประเด็น (*Issue*) ที่เกิดขึ้นในการดำเนินงานของโครงการ ตลอดจนยังคงมีความสัมพันธ์กับข้อมูลที่เกี่ยวข้องในการประเมินภาระงานอีกด้วย
2. คลาส *DevelopLanguage* เป็นคลาสที่เป็นตัวแทนภาษาโปรแกรมที่ใช้ในโครงการพัฒนาซอฟต์แวร์
3. คลาส *BusinessType* เป็นคลาสที่เป็นตัวแทนประเภทธุรกิจของลูกค้าที่เป็นผู้ว่าจ้างในโครงการพัฒนาซอฟต์แวร์
4. คลาส *Attachment* เป็นคลาสที่แสดงถึงเอกสารแนบที่ได้แนบเพื่อเป็นข้อมูลประกอบของโครงการ ซึ่งในแต่ละโครงการไม่จำเป็นต้องแนบเอกสาร หรือสามารถแนบเอกสารได้มากกว่าหนึ่งเอกสาร
5. คลาส *ContactPerson* เป็นคลาสที่เป็นตัวแทนผู้ที่เกี่ยวข้องในโครงการ ซึ่งแต่ละโครงการไม่จำเป็นต้องบันทึกข้อมูล หรือบันทึกได้มากกว่าหนึ่งครั้ง
6. คลาส *Environment* เป็นคลาสที่เป็นตัวแทนเครื่องมือ (*Tools*) ที่จะต้องใช้ในโครงการ ซึ่งแต่ละโครงการไม่จำเป็นต้องบันทึกข้อมูล หรือบันทึกได้มากกว่าหนึ่งครั้ง
7. คลาส *Issue* เป็นคลาสที่เป็นตัวแทนถึงประเด็นต่างๆที่เกิดขึ้นในการดำเนินงานของโครงการ ซึ่งในแต่ละโครงการไม่จำเป็นต้องบันทึกประเด็นต่างๆ หรือบันทึกได้มากกว่าหนึ่งประเด็น
8. คลาส *ProjectSize* เป็นคลาสนามธรรม (*Abstract*) ที่ใช้ในการกำหนดการจัดการเกี่ยวกับการระบุและการคำนวณหาขนาดของโครงการ โดยมีคลาสที่สืบทอดคุณสมบัติต่อไปอีกสองคลาสคือคลาสฟังก์ชันพอยต์ และ คลาสอ็อบเจกต์พอยต์ โดยในการประเมินภาระงานแต่ละเฟสจะต้องมีข้อมูลขนาดของโครงการเพื่อใช้ในการประเมินทุกครั้ง
9. คลาส *FunctionPoint* เป็นคลาสที่จัดการเกี่ยวกับการคำนวณหาขนาดของโครงการด้วยวิธีของฟังก์ชันพอยต์
10. คลาส *ObjectPoint* เป็นคลาสที่จัดการเกี่ยวกับการคำนวณหาขนาดของโครงการด้วยวิธีของ อ็อบเจกต์พอยต์
11. คลาส *CostFactor* เป็นคลาสที่จัดการเกี่ยวกับค่าปัจจัยที่ส่งผลต่อค่าใช้จ่ายของโครงการ โดยการประเมินภาระงานในบางครั้งอาจจะไม่ได้ใช้ข้อมูลนี้ในการประเมินภาระงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. คลาส *ScaleFactor* เป็นคลาสที่จัดการเกี่ยวกับการปัจจัยที่ส่งผลต่อขนาดของโครงการ

13. คลาส *EstimationResult* เป็นคลาสที่เป็นตัวแทนผลของการประมาณภาระงานของโครงการ

14. คลาส *Estimation* เป็นคลาสที่ใช้ในการประเมินภาระงานของโครงการ ซึ่งทำหน้าที่เลือกแบบจำลองที่เหมาะสมกับประเภทของโครงการนั้น

3.2.3 ซีควেনซ์ไดอะแกรม (Sequence Diagram)

ซีควেনซ์ไดอะแกรมเป็นแผนภาพสำหรับแสดงการปฏิสัมพันธ์ (Interaction) กันระหว่างอ็อบเจกต์ ตามลำดับการทำงานของเหตุการณ์ที่เกิดขึ้น โดยแต่ละอ็อบเจกต์จะถูกกระตุ้นให้ทำงานผ่านทางข้อความ (Message) ซึ่งในระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้น ได้ออกแบบลำดับโครงสร้างของการทำงานออกเป็น 4 รูปแบบ คือ

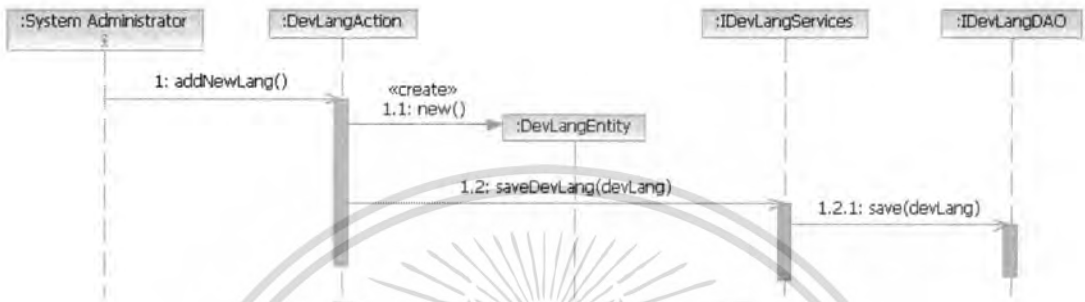
- **คลาสปฏิบัติการ (Action Class)** ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้ระบบและบริการของระบบ (Services) ที่ระบบได้เตรียมไว้ โดย Action Class นี้จะรับข้อความจากผู้ใช้ระบบและจะทำหน้าที่เรียกใช้บริการที่เหมาะสมกับข้อความที่ผู้ใช้ระบบได้ระบุมา คลาสดังกล่าวนี้จึงจะไม่มีการทำงานในส่วนของตรรกะทางธุรกิจ (Business Logic) โดยในระบบจะต่อท้ายชื่อคลาสประเภทนี้ว่า “Action”
- **คลาสให้บริการ (Services Class)** ทำหน้าที่ประมวลผลการทำงานให้เป็นไปตามตรรกะทางธุรกิจ (Business Logic) โดยในระบบจะต่อท้ายชื่อคลาสประเภทนี้ว่า “Services”
- **คลาสเชื่อมต่อฐานข้อมูล (Data Access Object Class)** ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อระหว่างระบบกับฐานข้อมูล และมีการทำงานพื้นฐานของฐานข้อมูล เช่นการเรียกดูข้อมูล การบันทึกข้อมูล รวมถึงการลบข้อมูล โดยในระบบจะต่อท้ายชื่อคลาสประเภทนี้ว่า “DAO”
- **คลาสเอนทิตี (Entity Class)** ทำหน้าที่เสมือนเป็นข้อมูลหนึ่งเรคคอร์ดของตารางในฐานข้อมูล โดยแอทริบิวต์ของคลาสเอนทิตี จะมีลักษณะข้อมูลเช่นเดียวกับแอทริบิวต์ของตารางในฐานข้อมูลที่คลาสเอนทิตีนั้นแสดงถึง โดยเอนทิตีคลาสนั้นในระบบจะต่อท้ายชื่อคลาสว่า “Entity”

สำหรับการแสดงซีควেনซ์ไดอะแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์จะอธิบายถึงการทำงานที่สำคัญ ดังนี้

1. การจัดการข้อมูลภาษาโปรแกรม

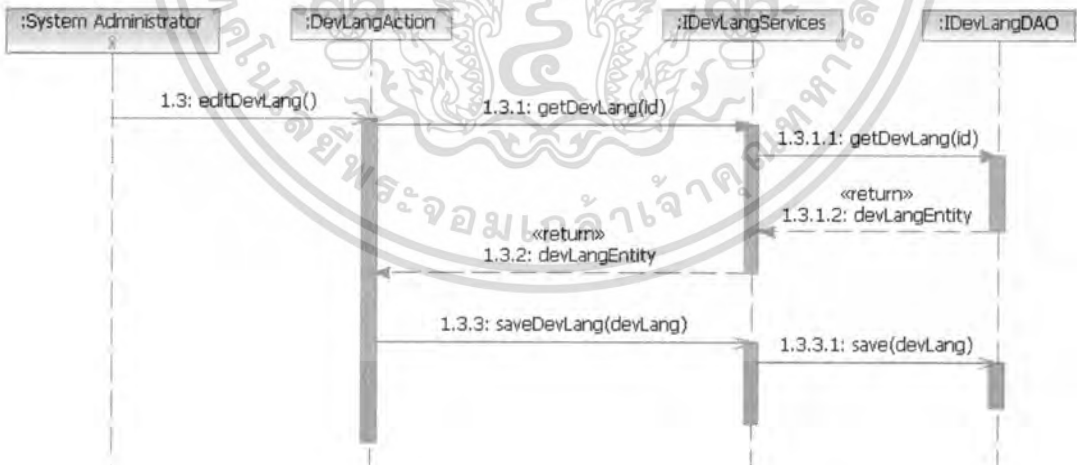
การทำงานในส่วนของการจัดการข้อมูลภาษาโปรแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นแบ่งการทำงานออกเป็น 2 ส่วน คือ การเพิ่มข้อมูลภาษาโปรแกรม และเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขข้อมูลภาษาโปรแกรม ดังแสดงซีเควนซ์ไดอะแกรมได้ในรูปที่ 3.3 และ รูปที่ 3.4 การเพิ่มข้อมูลภาษาโปรแกรมนั้นมีการทำงาน โดยเริ่มต้นจากผู้ดูแลระบบเข้าสู่หน้าจอการเพิ่มข้อมูล ภาษาโปรแกรม จากนั้นระบุข้อมูลที่ต้องการบันทึกแล้วจึงส่งข้อความ addNewLang() ไป เพื่อ บันทึกข้อมูล ระบบจะสร้างเอนทิตีอ็อบเจกต์ที่เป็นตัวแทนของข้อมูลภาษาโปรแกรมขึ้นมาใหม่เพื่อกำหนดค่าและส่งต่อไปเพื่อเรียกบริการบันทึกข้อมูลลงฐานข้อมูลต่อไป



รูปที่ 3.3 ซีเควนซ์ไดอะแกรมของการเพิ่มข้อมูลภาษาโปรแกรม

การทำงานในส่วนของการแก้ไขข้อมูลภาษาโปรแกรมนั้นเริ่มต้นจากการที่ผู้ดูแลระบบส่งข้อความเพื่อเรียกข้อมูลภาษาโปรแกรมขึ้นมาแก้ไข เมื่อระบบเรียกข้อมูลขึ้นมาแล้วนั้นผู้ดูแลระบบจะบันทึกข้อมูลที่ต้องการแก้ไข จากนั้นส่งบันทึกลงฐานข้อมูลต่อไป ดังรูปที่ 3.4



รูปที่ 3.4 ซีเควนซ์ไดอะแกรมของการแก้ไขข้อมูลภาษาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การจัดการข้อมูลประเภทธุรกิจ

การทำงานในส่วนของการจัดการข้อมูลประเภทธุรกิจของระบบช่วยประมาณภาระงาน ในการพัฒนาซอฟต์แวร์นั้นแบ่งการทำงานออกเป็น 2 ส่วน คือ การเพิ่มข้อมูลประเภทธุรกิจ และการแก้ไขข้อมูลประเภทธุรกิจ ดังแสดงซีเควนซ์ไดอะแกรมได้ในรูปที่ 3.5 และ รูปที่ 3.6

การเพิ่มข้อมูลประเภทธุรกิจนั้นมีการทำงาน โดยเริ่มต้นจากผู้ดูแลระบบเข้าสู่หน้าจอการเพิ่มข้อมูล จากนั้นระบุข้อมูลที่ต้องการบันทึกแล้วจึงส่งข้อความ addNew () ไป เพื่อบันทึกข้อมูล ระบบจะสร้างเอนทิตีอ็อบเจกต์ที่เป็นตัวแทนของข้อมูลประเภทธุรกิจขึ้นมาใหม่เพื่อกำหนดค่าและส่งต่อไปเพื่อเรียกบริการบันทึกข้อมูลลงฐานข้อมูลต่อไป



รูปที่ 3.5 ซีเควนซ์ไดอะแกรมของการเพิ่มประเภทธุรกิจ

สำหรับในส่วนของการแก้ไขข้อมูลนั้นมีการทำงานคล้ายกับการแก้ไขข้อมูลภาษาโปรแกรม โดยผู้ใช้ระบบจะเรียกข้อมูลของประเภทธุรกิจที่ต้องการแก้ไขขึ้นมา จากนั้นทำการระบุข้อมูลที่ต้องการแก้ไข แล้วบันทึกข้อมูลการแก้ไขลงฐานข้อมูล ดังรูปที่ 3.6

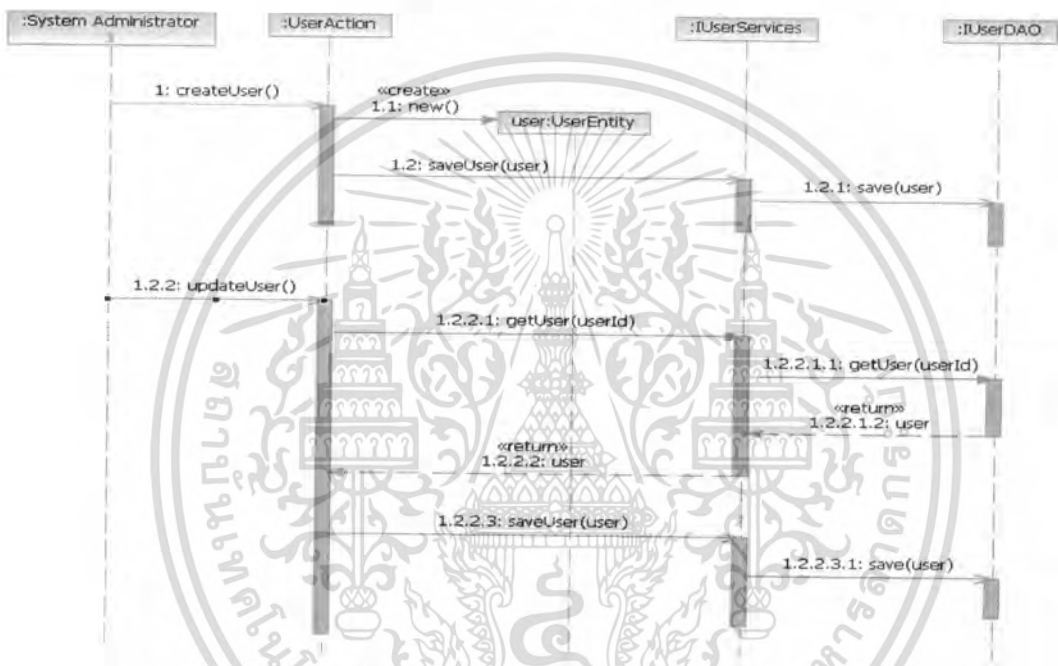


รูปที่ 3.6 ซีเควนซ์ไดอะแกรมของการแก้ไขข้อมูลประเภทธุรกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การจัดการข้อมูลผู้ใช้ระบบ

การจัดการข้อมูลผู้ใช้ระบบมีการทำงาน 2 ส่วน คือการเพิ่มข้อมูลผู้ใช้ระบบ และการแก้ไขข้อมูลของผู้ใช้ระบบ โดยในการเพิ่มข้อมูลผู้ใช้ระบบจะเริ่มต้นการทำงานจากการที่ผู้ดูแลระบบได้เข้าสู่หน้าจอของการเพิ่มข้อมูลผู้ใช้ระบบ จากนั้นระบุข้อมูลส่วนตัวของผู้ใช้ระบบที่ต้องการเพิ่มข้อมูล แล้วจึงส่งบันทึกข้อมูล ระบบจะทำงานตามโครงสร้างลำดับชั้นที่ได้กำหนดไว้เพื่อบันทึกข้อมูลผู้ใช้ระบบต่อไป เช่นเดียวกับการแก้ไขข้อมูลผู้ใช้ระบบที่จะเรียกข้อมูลของผู้ใช้ระบบขึ้นมาก่อน แล้วจึงบันทึกข้อมูลที่แก้ไขแล้วกลับไปยังฐานข้อมูล ดังแสดงได้ในรูปที่ 3.7



รูปที่ 3.7 ซีควเอนซ์ไดอะแกรมของการเพิ่มและแก้ไขข้อมูลผู้ใช้ระบบ

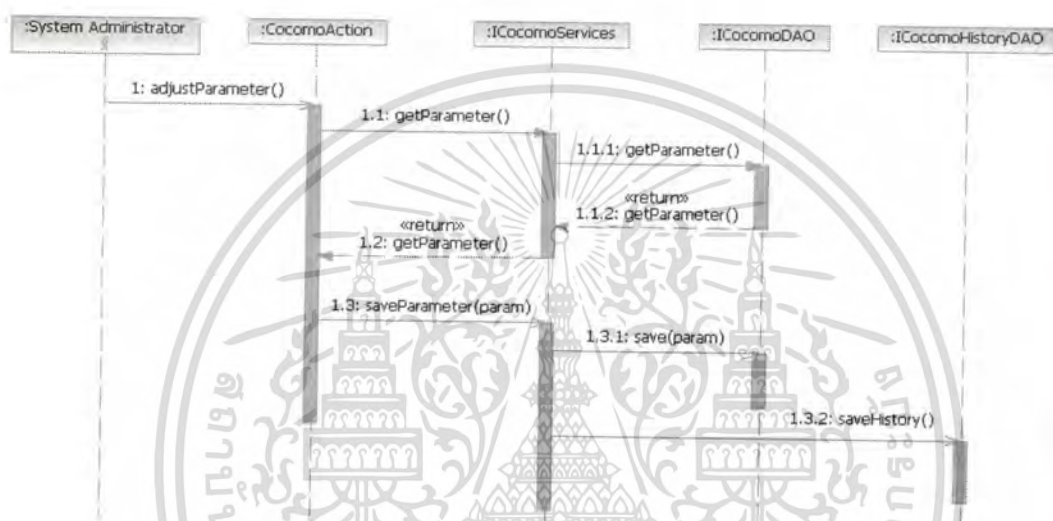
4. การปรับปรุงค่าพารามิเตอร์ของแบบจำลอง COCOMO II

ซีควเอนซ์ไดอะแกรมของการปรับปรุงค่าพารามิเตอร์ของแบบจำลอง COCOMO II นั้นแสดงได้ดังรูปที่ 3.8 โดยแบบจำลอง COCOMO II ที่ใช้ในระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์มีอยู่หลายรูปแบบ ในการปรับปรุงค่าพารามิเตอร์ของแบบจำลอง COCOMO II นั้นได้สรุปการทำงานที่เหมือนกันออกมาเป็นซีควเอนซ์ไดอะแกรมดังรูปด้านล่าง ซึ่งมีการทำงานที่เริ่มต้นจากการที่ผู้ดูแลระบบได้เข้าสู่หน้าจอปรับปรุงค่าพารามิเตอร์ จากนั้นเลือกพารามิเตอร์ที่ต้องการปรับปรุง ระบบจะเรียกข้อมูลของพารามิเตอร์ดังกล่าวขึ้นมาให้กับผู้ดูแลระบบได้ปรับปรุงและบันทึกลงสู่ฐานข้อมูล ซึ่งการปรับปรุงค่าพารามิเตอร์ในทุกๆครั้ง ระบบจะบันทึกประวัติการแก้ไขพารามิเตอร์เอาไว้เพื่อเป็นข้อมูลในการตรวจสอบการทำงานรวมถึงใช้ในการปรับปรุงพารามิเตอร์ครั้งต่อไปให้มีความเหมาะสมมากยิ่งขึ้น

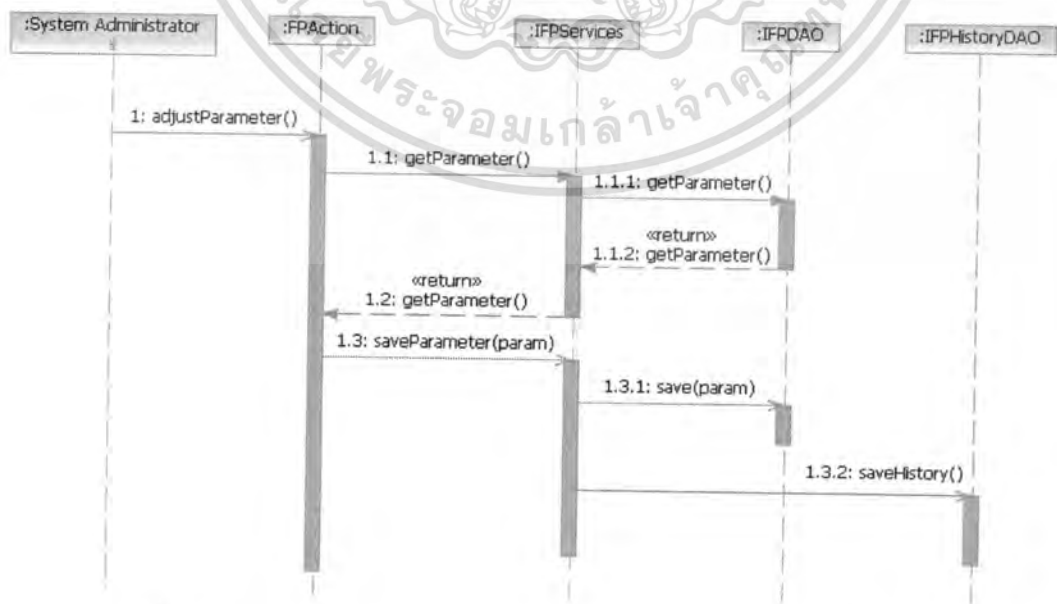
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การปรับปรุงค่าพารามิเตอร์ของแบบจำลองฟังก์ชันพอยต์

การทำงานของ การปรับปรุงค่าพารามิเตอร์ของแบบจำลองฟังก์ชันพอยต์นั้นเริ่มต้นจากการที่ผู้ดูแลระบบเข้าสู่หน้าจอปรับปรุงค่าพารามิเตอร์ จากนั้นเลือกพารามิเตอร์ที่ต้องการปรับปรุง ระบบจะเรียกข้อมูลของพารามิเตอร์ดังกล่าวขึ้นมาให้กับผู้ดูแลระบบได้ปรับปรุงและบันทึกลงสู่ฐานข้อมูล ซึ่งการปรับปรุงค่าพารามิเตอร์ในทุกๆ ครั้ง ระบบจะบันทึกประวัติการแก้ไขพารามิเตอร์เอาไว้เพื่อเป็นข้อมูลในการตรวจสอบการทำงาน โดยซีเควนซ์ไคอะแกรมของการปรับปรุงค่าพารามิเตอร์ของแบบจำลองฟังก์ชันพอยต์ นั้นแสดงได้ ดังรูปที่ 3.9



รูปที่ 3.8 ซีเควนซ์ไคอะแกรมของการปรับปรุงพารามิเตอร์ของ COCOMO II



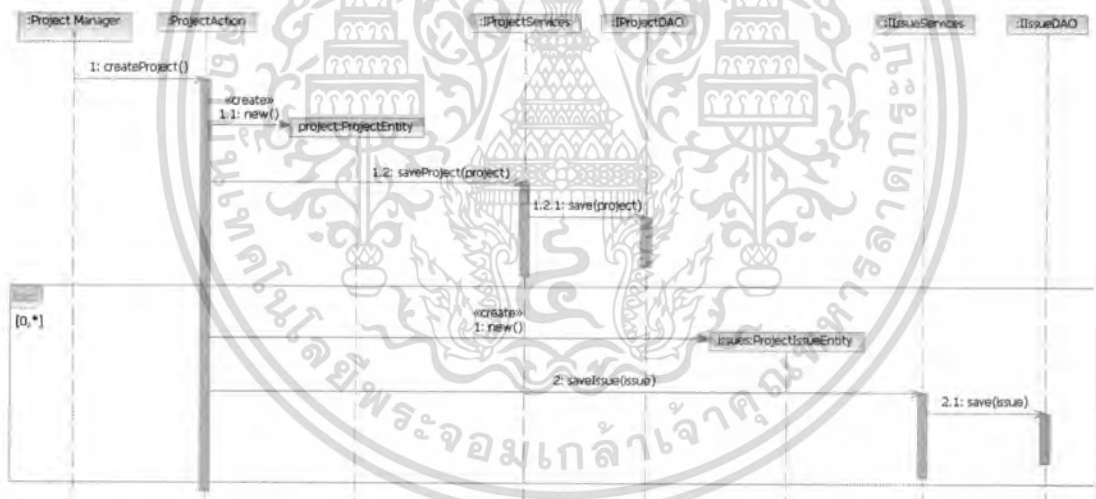
รูปที่ 3.9 ซีเควนซ์ไคอะแกรมของการปรับปรุงพารามิเตอร์ของฟังก์ชันพอยต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การบันทึกข้อมูลโครงการ

การบันทึกข้อมูลโครงการถือเป็นการทำงานหลักของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ โดยในการบันทึกข้อมูลของโครงการจะมีการทำงานส่วนอื่นแฝงอยู่ภายใน เช่น การบันทึกประเด็นต่างๆที่เกี่ยวข้องกับโครงการ การคำนวณหาขนาดของซอฟต์แวร์ด้วยวิธีฟังก์ชันพอยต์ การประมาณภาระงานด้วยแบบจำลอง COCOMO II เป็นต้น สำหรับซีเควนซ์ไดอะแกรมในส่วนของการบันทึกข้อมูลโครงการนั้นจะกล่าวถึงเพียงการบันทึกข้อมูลโครงการและการบันทึกประเด็นของโครงการ ดังแสดงได้ในรูปที่ 3.10

ซีเควนซ์ไดอะแกรมของการบันทึกข้อมูลโครงการนั้นแสดงการทำงานโดยเริ่มต้นจากผู้บริหารโครงการเข้าสู่หน้าจอการบันทึกข้อมูลโครงการ จากนั้นระบุข้อมูลต่างๆที่เป็นรายละเอียดของโครงการแล้วจึงส่งบันทึกโครงการผ่านทางคลาส ProjectAction เพื่อเรียกใช้บริการบันทึกข้อมูลโครงการต่อไป ในการบันทึกข้อมูลโครงการนั้นผู้บริหารโครงการสามารถบันทึกประเด็นที่เกิดขึ้น หรือ คาดว่าจะเกิดขึ้นในโครงการนั้นๆ ได้ ไม่จำกัดจำนวนหรือผู้บริหารโครงการจะบันทึกประเด็นของโครงการภายหลังก็ได้ ดังที่แสดงด้วยสัญลักษณ์การทำซ้ำแบบไม่จำกัดจำนวน



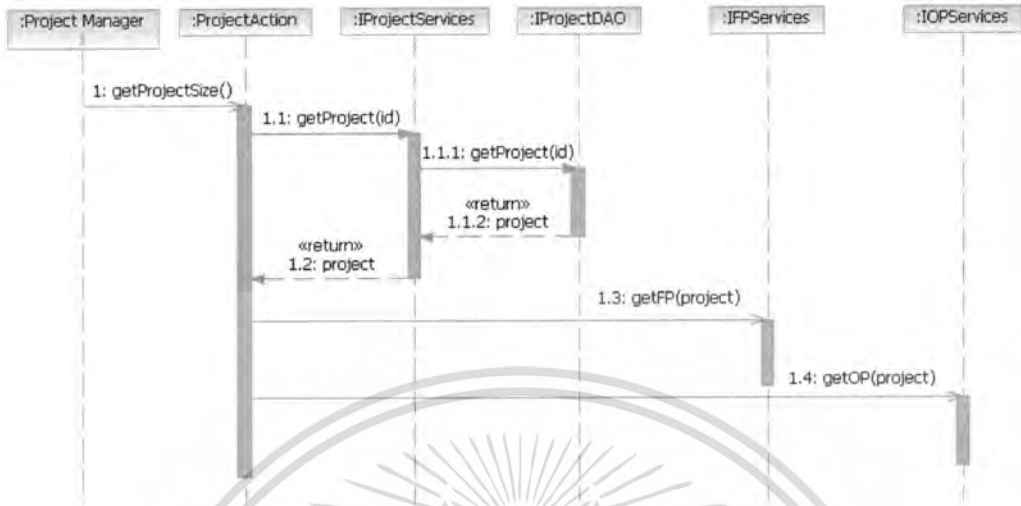
รูปที่ 3.10 ซีเควนซ์ไดอะแกรมของการบันทึกข้อมูลโครงการ

7. การคำนวณขนาดของโครงการ

การคำนวณขนาดของโครงการเป็นการทำงานย่อยภายใต้การบันทึกข้อมูลโครงการ ใช้สำหรับคำนวณขนาดของโครงการเพื่อใช้ในการประมาณภาระงานในแบบจำลอง COCOMO II โดยในระบบนั้นได้ใช้เครื่องมือในการหาขนาดของโครงการอยู่ด้วยกัน 2 เครื่องมือคือ ฟังก์ชันพอยต์ และอีอบเจกต์พอยต์ ซึ่งการทำงานของกรคำนวณขนาดของโครงการเริ่มต้นจากการที่ผู้บริหารโครงการเรียกดูขนาดของโครงการผ่านทางหน้าจอการบันทึกข้อมูลโครงการ ระบบจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

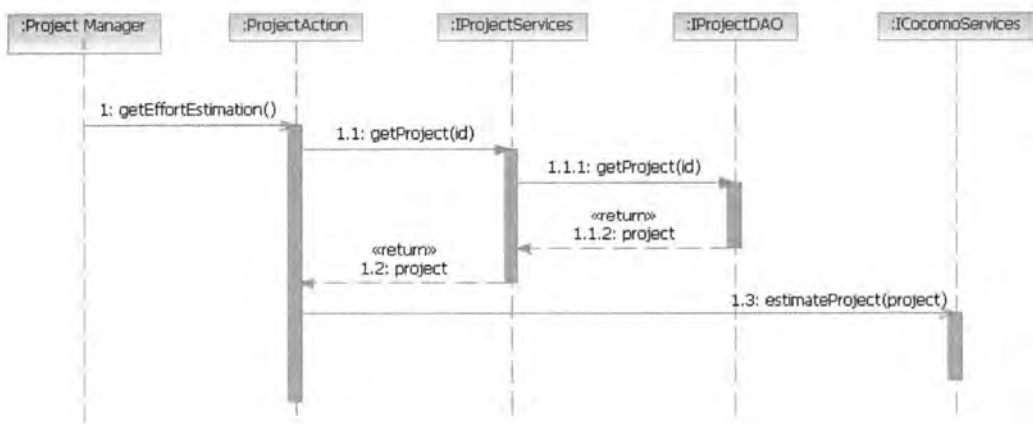
เรียกดูข้อมูลของโครงการที่ต้องการหาขนาด จากนั้นส่งข้อมูลของโครงการดังกล่าวไปให้กับ ฟังก์ชันพอยต์และอ็อบเจกต์พอยต์เพื่อคำนวณหาขนาดของโครงการต่อไป ดังรูปที่ 3.11



รูปที่ 3.11 ซีควেনซ์ไดอะแกรมของการคำนวณขนาดโครงการ

8. การประมาณภาระงานของโครงการ

การประมาณภาระงานของโครงการเป็นการทำงานย่อยภายใต้การบันทึกข้อมูลของโครงการ ใช้สำหรับประมาณภาระงานของโครงการตามแบบจำลอง COCOMO II โดยนำข้อมูลโครงการที่บันทึกไว้มาเข้าสู่แบบจำลองเพื่อประมาณภาระงาน ซึ่งการทำงานของประมาณภาระงานของโครงการเริ่มต้นจากการที่ผู้บริหารโครงการเรียกดูการประมาณภาระงานผ่านทางหน้าจอการบันทึกข้อมูลโครงการ ระบบจะเรียกดูข้อมูลของโครงการที่ต้องการประมาณภาระงาน จากนั้นส่งข้อมูลของโครงการดังกล่าวไปให้กับคลาสที่ให้บริการ เพื่อเรียกดูภาระงานที่ประมาณได้ต่อไป โดยซีควেনซ์ไดอะแกรมของการประมาณภาระงานของโครงการแสดงได้ ดังรูปที่ 3.12



รูปที่ 3.12 ซีควেনซ์ไดอะแกรมของการประมาณภาระงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 แผนภาพความสัมพันธ์ระหว่างเอนทิตี (E-R Diagram)

ในส่วนของการออกแบบโครงสร้างฐานข้อมูลของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้น ได้ออกแบบตามหลักของฐานข้อมูลเชิงสัมพันธ์ โดยความสัมพันธ์ของเอนทิตี แสดงได้ดังแผนภาพความสัมพันธ์ระหว่างเอนทิตี ดังรูปที่ 3.13 โดยมีเอนทิตีในระบบทั้งสิ้น 20 เอนทิตี ดังแผนภาพต่อไปนี้



3.2.4 พจนานุกรมข้อมูล

ตารางที่ 3.9 T_DEVELOPMENT_LANGUAGE ข้อมูลภาษาโปรแกรมที่ใช้ในการพัฒนาระบบ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
LANGUAGE_ID	เลขที่อ้างอิงภาษาโปรแกรมที่ใช้ในการพัฒนา	int	PK	
LANGUAGE_NAME	ภาษาโปรแกรมที่ใช้ในการพัฒนา	varchar(100)		
DESCRIPTION	คำอธิบายภาษาโปรแกรมที่ใช้ในการพัฒนา	varchar(255)		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.10 T_BUSINESS_TYPE_LKUP ข้อมูลประเภทธุรกิจของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
BUSINESS_TYPE_ID	เลขที่อ้างอิงประเภทธุรกิจ	int	PK	
BUSINESS_TYPE_NAME	ประเภทธุรกิจ	varchar(100)		
DESCRIPTION	คำอธิบายประเภทธุรกิจ	varchar(255)		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.11 T_CUSTOMER ข้อมูลลูกค้าที่จ้างให้พัฒนาซอฟต์แวร์

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
CUSTOMER_ID	เลขที่อ้างอิงลูกค้า	int	PK	
CUSTOMER_NAME	ชื่อลูกค้า	varchar(255)		
ADDRESS	ที่อยู่ของลูกค้า	varchar(1000)		
TELEPHONE	หมายเลขโทรศัพท์	varchar(100)		
E_MAIL	อีเมลล์ของลูกค้าที่ติดต่อได้	varchar(100)		
DESCRIPTION	รายละเอียดของลูกค้า	varchar(1000)		

ตารางที่ 3.11 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.12 T_USER ข้อมูลผู้ใช้ระบบ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
USER_ID	เลขที่อ้างอิงผู้ใช้งานระบบ	int	PK	
POS_ID	เลขที่อ้างอิงตำแหน่ง	int	FK	T_POSITION
TEAM_ID	เลขที่อ้างอิงทีมในการทำงาน	int	FK	T_USER
USER_NAME	ชื่อที่ใช้ในการเข้าสู่ระบบ	varchar(15)		
USER_PASSWORD	รหัสผ่านที่ใช้ในการเข้าสู่ระบบ	varchar(100)		
FIRST_NAME	ชื่อจริงของผู้ใช้ระบบ	varchar(100)		
LAST_NAME	นามสกุลของผู้ใช้ระบบ	varchar(100)		
E_MAIL	อีเมลล์ของผู้ใช้ระบบ	varchar(100)		
IS_TEAM_LEADER	ค่าที่ใช้อ้างอิงการเป็นหัวหน้าทีม (0=ไม่ใช่, 1=ใช่)	bit		
ROLE	สิทธิในการใช้งานระบบ	int		
STATUS	สถานะการใช้งานระบบ (0=ปกติ, 1=ระงับการใช้งาน)	bit		

ตารางที่ 3.13 T_POSITION ข้อมูลตำแหน่งงาน

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
POS_ID	เลขที่อ้างอิงตำแหน่ง	int	PK	
POS_NAME	ชื่อตำแหน่ง	varchar(100)		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.14 T_FUNCTION_POINT ข้อมูลค่านำหนักของฟังก์ชันพอยต์ของแบบจำลองในการ
คำนวณหาขนาดของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
FP_ID	เลขอ้างอิงค่านำหนักของฟังก์ชันพอยต์	int	PK	
ILF_LOW	ค่านำหนักของอ้างอิงถึงเพิ่มข้อมูลที่เก็บภายในระบบที่มีระดับความยากต่ำ	int		
ILF_AVG	ค่านำหนักของอ้างอิงถึงเพิ่มข้อมูลที่เก็บภายในระบบที่มีระดับความยากปานกลาง	int		
ILF_HIGH	ค่านำหนักของอ้างอิงถึงเพิ่มข้อมูลที่เก็บภายในระบบที่มีระดับความยากสูง	int		
EIF_LOW	ค่านำหนักของอ้างอิงถึงเพิ่มข้อมูลภายนอกระบบที่มีระดับความยากต่ำ	int		
EIF_AVG	ค่านำหนักของอ้างอิงถึงเพิ่มข้อมูลภายนอกระบบที่มีระดับความยากปานกลาง	int		
EIF_HIGH	ค่านำหนักของอ้างอิงถึงข้อมูลภายนอกระบบที่มีระดับความยากสูง	int		
EO_LOW	ค่านำหนักของการแสดงผลที่มีระดับความยากต่ำ	int		
EO_AVG	ค่านำหนักของการแสดงผลที่มีระดับความยากปานกลาง	int		
EO_HIGH	ค่านำหนักของการแสดงผลที่มีระดับความยากสูง	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.14 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
EI_LOW	ค่านำหนักของการป้อนข้อมูลเข้าสู่ระบบที่มีระดับความยากต่ำ	int		
EI_AVG	ค่านำหนักของการป้อนข้อมูลเข้าสู่ระบบที่มีระดับความยากปานกลาง	int		
EI_HIGH	ค่านำหนักของการป้อนข้อมูลเข้าสู่ระบบที่มีระดับความยากสูง	int		
EQ_LOW	ค่านำหนักของการสืบค้นข้อมูลที่มีระดับความยากต่ำ	int		
EQ_AVG	ค่านำหนักของการสืบค้นข้อมูลที่มีระดับความยากปานกลาง	int		
EQ_HIGH	ค่านำหนักของการสืบค้นข้อมูลที่มีระดับความยากสูง	int		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลล่าสุด	int		
UPDATE_DATE	วันที่บันทึกข้อมูลล่าสุด	date		

ตารางที่ 3.15 T_OBJECT_POINT ข้อมูลค่านำหนักของอ็อบเจกต์พอยต์ของแบบจำลองในการคำนวณหาขนาดของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
OP_ID	เลขอ้างอิงค่านำหนักของอ็อบเจกต์พอยต์	int	PK	
SCN_LOW	ค่านำหนักของหน้าจอย่างง่าย	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาเบใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.15 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
SCN_AVG	ค่านำหนักของหน้าจอปานกลาง	int		
SCN_HIGH	ค่านำหนักของหน้าจอยก	int		
RPT_LOW	ค่านำหนักของรายงานอย่างง่าย	int		
RPT_AVG	ค่านำหนักของรายงานปานกลาง	int		
RPT_HIGH	ค่านำหนักของรายงานอย่างยาก	int		
CPN_LOW	ค่านำหนักของส่วนประกอบ (Component) อย่างง่าย	Int		
CPN_AVG	ค่านำหนักของส่วนประกอบ (Component) ปานกลาง	int		
CPN_HIGH	ค่านำหนักของส่วนประกอบ (Component) อย่างยาก	int		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		

ตารางที่ 3.16 T_SCALE_DRIVER ข้อมูลแรงขับเคลื่อนด้านขนาดของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
SCALE_DRIVER_ID	เลขอ้างอิงค่านำหนักของแรงขับเคลื่อนด้านขนาดของโครงการ	int	PK	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.16 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PMAT_WEIGHT	ค่าน้ำหนักของปัจจัยวุฒิภาวะในการควบคุมกระบวนการพัฒนา (Process Maturity)	decimal(5, 2)		
TEAM_WEIGHT	ค่าน้ำหนักของปัจจัยปัจจัยการทำงานร่วมกัน (Team Cohesion)	decimal(5, 2)		
RESL_WEIGHT	ค่าน้ำหนักของปัจจัยความเสี่ยงของโครงการ (Architecture/Risk resolution)	decimal(5, 2)		
FLEX_WEIGHT	ค่าน้ำหนักของปัจจัยความยืดหยุ่นของซอฟต์แวร์ (Flexibility)	decimal(5, 2)		
PREC_WEIGHT	ค่าน้ำหนักของปัจจัยการมีแบบอย่างมาก่อน (Precedentedness)	decimal(5, 2)		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		

ตารางที่ 3.17 T_COST_DRIVER ปัจจัยขับเคลื่อนด้านต้นทุนที่แปรผันต่อขนาดของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
COST_DRIVER_ID	เลขที่อ้างอิงปัจจัยขับเคลื่อนด้านต้นทุน	int	PK	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.17 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
TIME_WEIGHT	ค่าน้ำหนักของข้อจำกัดด้านเวลาในการทำงาน (Execution Time Constraint)	decimal(5,2)		
DOCU_WEIGHT	ค่าน้ำหนักของความต้องการด้านเอกสาร (Documentation match to lift cycle needs)	decimal(5,2)		
RUSE_WEIGHT	ค่าน้ำหนักของความต้องการนำซอฟต์แวร์มาใช้ได้ใหม่ (Require Reusability)	decimal(5,2)		
CPLX_WEIGHT	ค่าน้ำหนักของความซับซ้อนของโครงการ (Product Complexity)	decimal(5,2)		
STOR_WEIGHT	ค่าน้ำหนักของขนาดของพื้นที่การเก็บข้อมูล (Main Storage Constraint)	decimal(5,2)		
DATA_WEIGHT	ค่าน้ำหนักของขนาดของฐานข้อมูล (Database Size)	decimal(5,2)		
RELY_WEIGHT	ค่าน้ำหนักของด้านความน่าเชื่อถือของซอฟต์แวร์ (Required Software Reliability)	decimal(5,2)		
PVOL_WEIGHT	ค่าน้ำหนักของความจำเป็นที่ต้องปรับเปลี่ยนแพลตฟอร์ม (Platform Volatility)	decimal(5,2)		
ACAP_WEIGHT	ค่าน้ำหนักของปัจจัยความสามารถในการวิเคราะห์ระบบ (Analyst Capability)	decimal(5,2)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.17 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PCAP_WEIGHT	ค่านำหนักของระดับปัจจัย ความสามารถของผู้พัฒนา ระบบ (Programmer Capability)	decimal(5,2)		
PCON_WEIGHT	ค่านำหนักของระดับปัจจัย ระยะเวลาการทำงานของ บุคคลากร (Personnel Continuity)	decimal(5,2)		
AEXP_WEIGHT	ค่านำหนักของระดับปัจจัย ด้านประสบการณ์ที่มีกับแอ ปพลิเคชัน (Application Experience)	decimal(5,2)		
PEXP_WEIGHT	ค่านำหนักของระดับปัจจัย ด้านประสบการณ์ที่มีกับ แพลตฟอร์ม (Platform Experience)	decimal(5,2)		
LTEX_WEIGHT	ค่านำหนักของระดับปัจจัย ด้านประสบการณ์ที่มีกับ ภาษาโปรแกรมและ เครื่องมือ (Language and Tool Experience)	decimal(5,2)		
TOOL_WEIGHT	ค่านำหนักของระดับปัจจัย ด้านเครื่องมือ (Use of Software Tools)	decimal(5,2)		
SITE_WEIGHT	ค่านำหนักของระดับปัจจัย ด้านการทำงานหลายที่ตั้ง (Multisite Development)	decimal(5,2)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.17 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
SCED_WEIGHT	ค่าน้ำหนักของระดับปัจจัยด้านความเร่งรัดในหมายกำหนดการงานของโครงการ (Required Development Schedule)	decimal(5,2)		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		

ตารางที่ 3.18 T_PROJECT ข้อมูลของโครงการพัฒนาซอฟต์แวร์ที่นำมาประเมินภาระงาน

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PROJ_ID	เลขที่อ้างอิงโครงการ	int	PK	
LANGUAGE_ID	เลขที่อ้างอิงภาษาโปรแกรมที่ใช้ในการพัฒนา	int	FK	T_DEVELOPMEN T_LANGUAGE
BUSINESS_TYP E_ID	เลขที่อ้างอิงประเภทธุรกิจ	int	FK	T_BUSINESS_ TYPE_LKUP
USER_ID	เลขที่อ้างอิงผู้ใช้งานระบบที่เป็นผู้บริหารโครงการ	int	FK	T_USER
CUSTOMER_ID	เลขที่อ้างอิงลูกค้า	int	FK	T_CUSTOMER
PROJ_CODE	รหัสโครงการ	varchar(10)		
PROJ_NAME	ชื่อโครงการ	varchar(100)		
DESCRIPTION	คำอธิบายโครงการ	varchar(255)		
IS_REUSED	ค่าที่ใช้ระบุว่าเป็นโครงการที่นำระบบเก่ามาพัฒนาต่อหรือไม่ (0=พัฒนาใหม่, 1=พัฒนาต่อจากระบบเก่า)	char(1)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาเบใช้ประโยชน์ด้านการศึกษา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.18 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
IS_OLD_PROJECT	ค่าที่ใช้ระบุว่า เป็นโครงการเดิมขององค์กรก่อนจะใช้ระบบประเมินหรือไม่ (0=โครงการในระบบ, 1=โครงการนอกระบบ)	char(1)		
START_DATE	วันที่เริ่มต้นโครงการ	date		
DUE_DATE	วันที่กำหนดส่งโครงการ	date		
EST_FINISH_DATE	วันที่ประมาณการว่าโครงการจะปิดโครงการ	date		
ACT_FINISH_DATE	วันสิ้นสุดโครงการ	date		
BUDGET	งบประมาณในการจัดทำโครงการ	decimal(10,2)		
EST_COST	ค่าใช้จ่ายของโครงการที่ประมาณการโดยผู้บริหารโครงการ	decimal(10,2)		
PROJ_ACT_COST	ค่าใช้จ่ายของโครงการที่เกิดขึ้นจริง	decimal(10,2)		
CONTRACT_NO	เลขที่หนังสือสัญญาว่าจ้าง	varchar(100)		
CONTRACT_DATE	วันที่ทำหนังสือสัญญาว่าจ้าง	date		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูลโครงการ (0=โครงการปกติ, 1=โครงการที่ถูกลบ)	char(1)		

ตารางที่ 3.19 T_PROJECT_CONTACT ข้อมูลของผู้ที่มีส่วนร่วมในโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
CONTACT_ID	เลขที่อ้างอิงผู้ติดต่อ	int	PK	
PROJ_ID	เลขที่อ้างอิงโครงการ	int	FK	T_PROJECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.19 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
CONTACT_NAME	ชื่อผู้ติดต่อ	varchar(255)		
POSITION	ตำแหน่งของผู้ติดต่อ	varchar(255)		
DEPARTMENT	หน่วยงานหรือแผนกของผู้ติดต่อ	varchar(255)		
TELEPHONE	หมายเลขโทรศัพท์ของผู้ติดต่อ	varchar(255)		
E_MAIL	อีเมลล์ของผู้ติดต่อ	varchar(255)		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.20 T_PROJECT_ENVIRONMENT ข้อมูลสภาพแวดล้อมที่ใช้ในโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
ENV_ID	เลขที่อ้างอิงข้อมูลสภาพแวดล้อม	int	PK	
PROJ_ID	เลขที่อ้างอิงโครงการ	int	FK	T_PROJECT
ENV_NAME	ชื่อสภาพแวดล้อม	varchar(255)		
ENV_VERSION	รุ่นของสภาพแวดล้อม	varchar(20)		
COMMENT	คำอธิบายเพิ่มเติม	varchar(512)		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.21 T_PROJECT_ATTACHMENT ข้อมูลเอกสารที่แนบเป็นข้อมูลประกอบในโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
ATTACH_ID	เลขที่อ้างอิงเอกสารแนบ	int	PK	
PROJ_ID	เลขที่อ้างอิงโครงการ	int	FK	T_PROJECT
ATTACH_NAME	ชื่อเอกสาร	varchar(255)		
PATH	ที่จัดเก็บเอกสารแนบ	varchar(255)		
DESCRIPTION	รายละเอียดเอกสารแนบ	varchar(255)		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้เผยแพร่ในวงกว้างโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.21 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
ATTACH_BY	ผู้ใช้ระบบที่แนบเอกสาร	int		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.22 T_PROJECT_ISSUE ข้อมูลประเด็นที่เกิดขึ้นในการดำเนินโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
ISSUE_ID	เลขที่อ้างอิงประเด็น	int	PK	
PROJ_ID	เลขที่อ้างอิงโครงการ	int	FK	T_PROJECT
ISSUE_TYPE	ประเภทของประเด็น	int		
TOPIC	หัวเรื่องของประเด็น	varchar(255)		
ISSUE_CODE	รหัสของประเด็น	varchar(50)		
FACTOR	ปัจจัยที่เป็นสาเหตุของ ประเด็น	int		
PRIORITY	ความสำคัญของประเด็น	int		
DUE_DATE	กำหนดการตอบสนอง ประเด็น	date		
FINISH_DATE	วันที่เสร็จสิ้นการทำงาน	date		
DESCRIPTION	คำอธิบายประเด็น	varchar(1000)		
RESOLUTION	รายละเอียดการตอบสนอง ต่อประเด็น	varchar(1000)		
STATUS	สถานะ	int		
ISSUED_BY	ผู้ใช้ระบบที่แจ้งประเด็น	int		
ASSIGNED_USER	ผู้ใช้ระบบที่ถูกมอบหมายให้ ตอบสนองต่อประเด็น	int		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.23 T_PROJECT_ISSUE_COMMENT ข้อมูลคำอธิบายประเด็นที่เกิดขึ้นในโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
COMMENT_ID	เลขที่อ้างอิงคำอธิบาย	int	PK	
ISSUE_ID	เลขที่อ้างอิงประเด็น	int	FK	T_PROJECT_ISSUE
DESCRIPTION	รายละเอียดคำอธิบาย	varchar(1000)		
COMMENT_BY	ผู้ใช้ระบบบันทึกคำอธิบาย	int		
COMMENT_DATE	วันที่ผู้ใช้ระบบบันทึกคำอธิบาย	date		
IS_DELETED	ค่าที่ใช้อ้างอิงการลบข้อมูล (0=ปกติ, 1=ถูกลบ)	bit		

ตารางที่ 3.24 T_PROJECT_FUNCTION_POINT ข้อมูลค่าฟังก์ชันพอยต์ของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PROJ_ID	เลขอ้างอิงโครงการ	int	PK, FK	T_PROJECT
FP_ID	เลขที่อ้างอิงค่าน้ำหนักของฟังก์ชันพอยต์	int	FK	T_FUNCTION_POINT
ILF_LOW_COUNT	จำนวนแฟ้มข้อมูลที่เก็บภายในระบบที่มีระดับความยากต่ำ	int		
ILF_AVG_COUNT	จำนวนแฟ้มข้อมูลที่เก็บภายในระบบที่มีระดับความยากปานกลาง	int		
ILF_HIGH_COUNT	จำนวนแฟ้มข้อมูลที่เก็บภายในระบบที่มีระดับความยากสูง	int		
EIF_LOW_COUNT	จำนวนแฟ้มข้อมูลภายนอกในระบบที่มีระดับความยากต่ำ	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.24 (ต่อ)

ชื่อแอททริบิวต์*	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
EIF_AVG_COUN T	จำนวนเพิ่มข้อมูลภายนอก ระบบที่มีระดับความยาก ปานกลาง	int		
EIF_HIGH_COUN T	จำนวนเพิ่มข้อมูลภายนอก ระบบที่มีระดับความยากสูง	int		
EO_LOW_COUN T	จำนวนหน้าจอกการแสดงผล ที่มีระดับความยากต่ำ	int		
EO_AVG_COUN T	จำนวนหน้าจอกการแสดงผล ที่มีระดับความยากปานกลาง	int		
EO_HIGH_COUN T	จำนวนหน้าจอกการแสดงผล ที่มีระดับความยากสูง	int		
EI_LOW_COUNT	จำนวนหน้าจอกการป้อน ข้อมูลเข้าสู่ระบบที่มีระดับ ความยากต่ำ	int		
EI_AVG_COUNT	จำนวนหน้าจอกการป้อน ข้อมูลเข้าสู่ระบบที่มีระดับ ความยากปานกลาง	int		
EI_HIGH_COUN T	จำนวนหน้าจอกการป้อน ข้อมูลเข้าสู่ระบบที่มีระดับ ความยากสูง	int		
EQ_LOW_COUN T	จำนวนการทำงานที่มีการ สืบค้นข้อมูลที่มีระดับความ ยากต่ำ	int		
EQ_AVG_COUN T	จำนวนการทำงานที่มีการ สืบค้นข้อมูลที่มีระดับความ ยากปานกลาง	int		
EQ_HIGH_COUN T	จำนวนการทำงานที่มีการ สืบค้นข้อมูลที่มีระดับความ ยากสูง	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.24 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
SUM_FP	ผลรวมค่าฟังก์ชันพอยต์	decimal(8,2)		
SLOC	ผลรวมจำนวนบรรทัดของการพัฒนาซอฟต์แวร์ในโครงการ	decimal(10,2)		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		

ตารางที่ 3.25 T_PROJECT_OBJECT_POINT ข้อมูลค่าอ็อบเจกต์พอยต์ของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PROJ_ID	เลขอ้างอิงโครงการ	int	PK, FK	T_PROJECT
OP_ID	เลขที่อ้างอิงค่านำหนักของอ็อบเจกต์พอยต์	int	FK	T_OBJECT_POIN T
SCN_LOW_COUNT	จำนวนหน้าจอลงอย่างง่าย	int		
SCN_AVG_COUNT	จำนวนหน้าจอลงปานกลาง	int		
SCN_HIGH_COUNT	จำนวนหน้าจอลงอย่างยาก	int		
RPT_LOW_COUNT	จำนวนรายงานอย่างง่าย	int		
RPT_AVG_COUNT	จำนวนรายงานปานกลาง	int		
RPT_HIGH_COUNT	จำนวนรายงานอย่างยาก	int		
CPN_LOW_COUNT	จำนวนส่วนประกอบ (Component) อย่างง่าย	int		
CPN_AVG_COUNT	จำนวนส่วนประกอบ (Component) อย่างปานกลาง	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.25 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
CPN_HIGH_COUNT	จำนวนส่วนประกอบ (Component) อย่างยาก	int		
SUM_FP	ผลรวมค่าฟังก์ชันพอยต์	decimal(8,2)		
SLOC	ผลรวมจำนวนบรรทัดของการพัฒนาซอฟต์แวร์ในโครงการ	decimal(10,2)		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		

ตารางที่ 3.26 T_PHASE_SCALE_FACTOR ข้อมูลมาตราส่วนของปัจจัยที่มีผลต่อโครงการ (Scale Factor)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PROJ_ID	เลขอ้างอิงโครงการ	int	PK, FK	T_PROJECT
SCALE_DRIVER_ID	เลขอ้างอิงค่านำหนักของแรงขับเคลื่อนด้านขนาดของโครงการ	int	FK	T_SCALE_DRIVER
PMAT_WEIGHT	ระดับปัจจัยวุฒิภาวะในการควบคุมกระบวนการพัฒนา (Process Maturity)	int		
TEAM_WEIGHT	ระดับปัจจัยปัจจัยการทำงานร่วมกัน (Team Cohesion)	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.26 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
RESL_WEIGHT	ระดับปัจจัยความเสี่ยงของโครงการ (Architecture/Risk resolution)	int		
FLEX_WEIGHT	ระดับปัจจัยความยืดหยุ่นของซอฟต์แวร์ (Flexibility)	int		
PREC_WEIGHT	ระดับปัจจัยการมีแบบอย่างมาก่อน (Precedentedness)	int		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		

ตารางที่ 3.27 T_PROJECT_COST_DRIVER ข้อมูลปัจจัยขับเคลื่อนด้านต้นทุนที่แปรผันต่อขนาดของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
PROJ_ID	เลขอ้างอิงโครงการ	int	PK, FK	T_PROJECT
COST_DRIVER_ID	เลขที่อ้างอิงค่านำหนักของปัจจัยขับเคลื่อนด้านต้นทุน	int	FK	T_COST_DRIVER
TIME_WEIGHT	ระดับปัจจัยข้อจำกัดด้านเวลาในการทำงาน (Execution Time Constraint)	int		
DOCU_WEIGHT	ระดับปัจจัยความต้องการด้านเอกสาร (Documentation match to lift cycle needs)	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.27 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
RUSE_WEIGHT	ระดับปัจจัยความต้องการนำซอฟต์แวร์มาใช้ได้ใหม่ (Require Reusability)	int		
CPLX_WEIGHT	ระดับปัจจัยความซับซ้อนของโครงการ (Product Complexity)	int		
STOR_WEIGHT	ระดับปัจจัยขนาดของพื้นที่การเก็บข้อมูล (Main Storage Constraint)	int		
DATA_WEIGHT	ระดับปัจจัยขนาดของฐานข้อมูล (Database Size)	int		
RELY_WEIGHT	ระดับปัจจัยด้านความน่าเชื่อถือของซอฟต์แวร์ (Required Software Reliability)	int		
PVOL_WEIGHT	ระดับปัจจัยความจำเป็นที่ต้องปรับเปลี่ยนแพลตฟอร์ม (Platform Volatility)	int		
ACAP_WEIGHT	ระดับปัจจัยความสามารถในการวิเคราะห์ระบบ (Analyst Capability)	int		
PCAP_WEIGHT	ระดับปัจจัยความสามารถของผู้พัฒนาระบบ (Programmer Capability)	int		
PCON_WEIGHT	ระดับปัจจัยระยะเวลาการทำงานของบุคลากร (Personnel Continuity)	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.27 (ต่อ)

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
AEXP_WEIGHT	ระดับปัจจัยด้านประสบการณ์ที่มีกับแอปพลิเคชัน (Application Experience)	int		
PEXP_WEIGHT	ระดับปัจจัยด้านประสบการณ์ที่มีกับแพลตฟอร์ม (Platform Experience)	int		
LTEX_WEIGHT	ระดับปัจจัยด้านประสบการณ์ที่มีกับภาษาโปรแกรมและเครื่องมือ (Language and Tool Experience)	int		
TOOL_WEIGHT	ระดับปัจจัยด้านเครื่องมือ (Use of Software Tools)	int		
SITE_WEIGHT	ระดับปัจจัยด้านการทำงานหลายที่ตั้ง (Multisite Development)	int		
SCED_WEIGHT	ระดับปัจจัยด้านความเร่งรัดในหมายกำหนดการงานของโครงการ (Required Development Schedule)	int		
SUM_COST_DRI VER	ผลคูณรวมของปัจจัยขับเคลื่อนด้านต้นทุน	decimal(6,2)		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		
UPDATE_BY	ผู้ที่ยืนยันข้อมูลล่าสุด	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.28 T_ESTIMATION_RESULT ข้อมูลผลการประเมินภาระงานของโครงการ

ชื่อแอททริบิวต์	คำอธิบาย	ชนิดข้อมูล	คีย์	ตารางอ้างอิง
RESULT_ID	เลขที่อ้างอิงผลลัพธ์การประเมิน	int	PK	
PROJ_ID	เลขอ้างอิงโครงการ	int	FK	T_PROJECT
EFFORT	ค่าภาระงานของการพัฒนาโครงการที่ประมาณได้ หน่วยเป็น PM (Person-Month)	decimal(6,2)		
SCHEDULE	ระยะเวลาของการพัฒนาโครงการที่ประมาณได้	decimal(6,2)		
SCALE_FACTOR	ผลการคำนวณค่าปัจจัยที่มีผลต่อการประเมินภาระงาน	decimal(6,2)		
COST_DRIVER	ผลการคำนวณค่าปัจจัยขับเคลื่อนด้านต้นทุนที่มีผลต่อการประเมินภาระงาน	decimal(6,2)		
FP_SIZE	ขนาดของโครงการที่คิดโดยฟังก์ชันพอยต์	decimal(6,2)		
OP_SIZE	ขนาดของโครงการที่คิดโดยอ็อบเจกต์พอยต์	decimal(6,2)		
IS_ACTIVE	ค่าอ้างอิงการใช้งาน (0=ไม่ได้ใช้งาน, 1=ใช้งานปกติ)	bit		
UPDATE_DATE	วันที่ผู้ใช้ระบบบันทึกข้อมูลครั้งล่าสุด	date		
UPDATE_BY	ผู้ใช้ระบบที่บันทึกข้อมูลครั้งล่าสุด	int		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบหน้าจอการทำงาน

4.1 การออกแบบหน้าจอ

ระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ถูกพัฒนาขึ้นในรูปแบบของเว็บแอปพลิเคชัน ซึ่งมีข้อดีอยู่ที่การติดตั้งระบบที่เครื่องแม่ข่ายเพียงที่เดียว และผู้ใช้ระบบสามารถใช้งานระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ผ่านทางเว็บเบราว์เซอร์ได้โดยไม่ต้องติดตั้งซอฟต์แวร์เพิ่มเติม ดังนั้นในส่วนของ การออกแบบหน้าจอของระบบนั้นจึงเป็นการออกแบบหน้าจอเว็บเพจ โดยระบบได้แบ่งการทำงานออกเป็น 4 ส่วน ดังนี้

1. การจัดการข้อมูลของผู้ดูแลระบบ เป็นหน้าจอกำหนดการทำงานสำหรับผู้ดูแลระบบที่ได้รับมอบหมายให้เป็นผู้ดูแลระบบเพื่อจัดการข้อมูลต่างๆ ที่เป็นข้อมูลตั้งต้นของระบบ โดยการจัดการข้อมูลครอบคลุมถึงการเพิ่มข้อมูล แก้ไขข้อมูล และยกเลิกการใช้งาน
2. การปรับปรุงพารามิเตอร์ เป็นหน้าจอกำหนดการทำงานในส่วนการทำงานของผู้ใช้ระบบที่ได้รับมอบหมายให้เป็นผู้ดูแลระบบเพื่อปรับปรุงค่าพารามิเตอร์ของแบบจำลองที่ใช้ในการประเมินภาระงานให้มีความเหมาะสมและใกล้เคียงกับการดำเนินงานของบริษัทมากที่สุด
3. การบันทึกข้อมูลโครงการ เป็นหน้าจอกำหนดการทำงานของผู้บริหารโครงการ ซึ่งเข้าใช้ระบบเพื่อบันทึกข้อมูลของโครงการทั้งโครงการที่ต้องการประเมินภาระงาน หรือโครงการเดิมของบริษัทที่ได้พัฒนาเสร็จสิ้นแล้ว
4. รายงานการพัฒนาโครงการ เป็นหน้าจอกำหนดการทำงานของผู้บริหารโครงการ และผู้บริหารของบริษัท ที่เรียกดูรายงานการดำเนินงานของโครงการแต่ละโครงการ เพื่อใช้ในการควบคุม ติดตาม และบริหารการทำงานของโครงการ

หน้าจอกำหนดการทำงานของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้น จะเริ่มต้นจากหน้าจอล็อกอิน เพื่อตรวจสอบสิทธิ์ในการเข้าใช้ระบบดังรูปที่ 4.1 โดยหลังจากที่ผู้ใช้ระบบผ่านการตรวจสอบสิทธิ์ในการเข้าใช้ระบบแล้ว ระบบจะแสดงหน้าจอกำหนดการทำงานหลักที่มีแถบเมนูการทำงานให้เลือกอยู่ฝั่งซ้ายมือของจอภาพ ซึ่งการแสดงผลเมนูนั้นจะขึ้นอยู่กับสิทธิ์ของผู้ใช้ระบบที่ล็อกอินเข้ามาดังรูปที่ 4.2

Motif
Technology

VISION OF PRACTICAL TECHNOLOGY

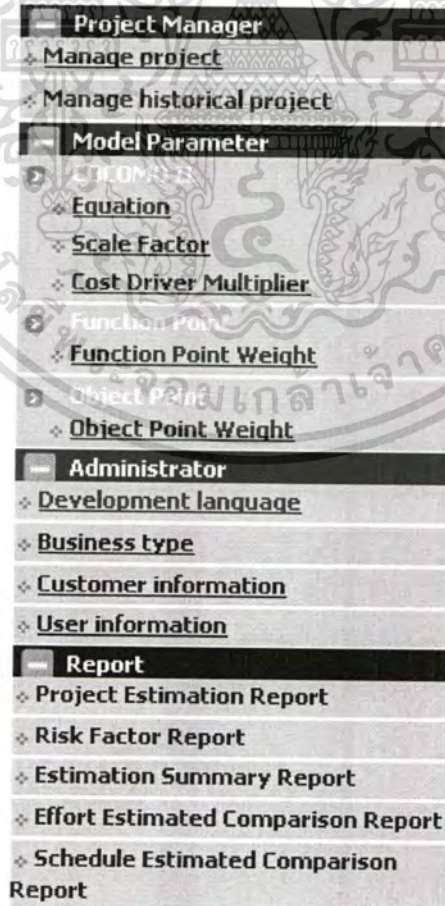
User name*

Password*

Login Reset

Copyright ©2008 All right reserved.
Designed by Thitichai Wongpiya (1523.2)

รูปที่ 4.1 หน้าจอล็อกอินเข้าสู่ระบบ



รูปที่ 4.2 เมนูการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวข้างต้นว่าระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นประกอบไปด้วย 4 ส่วนการทำงานหลัก สามารถอธิบายการทำงานของหน้าจอในแต่ละส่วนงานได้ ดังนี้

4.2 การออกแบบหน้าจอการจัดการข้อมูลของผู้ดูแลระบบ

การทำงานในส่วนนี้เป็นส่วนการทำงานของผู้ดูแลระบบ ซึ่งสามารถแบ่งการทำงานออกเป็น 4 การทำงานย่อย ดังนี้

4.2.1 ข้อมูลภาษาโปรแกรม

การทำงานในส่วนนี้เป็นการทำงานเกี่ยวกับการจัดการข้อมูลภาษาโปรแกรมที่จะต้องใช้ในการพัฒนาซอฟต์แวร์ของโครงการ และใช้สำหรับปรับปรุงค่าเปรียบเทียบจำนวนบรรทัดของซอฟต์แวร์ในแต่ละภาษาโปรแกรมกับค่าฟังก์ชันพอยต์และอ็อบเจกต์พอยต์ โดยการทำงานในส่วนนี้จะเริ่มจากหน้าแสดงรายการภาษาโปรแกรมที่บันทึกไว้ในระบบ ซึ่งในหน้าจอนี้จะมีปุ่มสำหรับบันทึกภาษาโปรแกรมใหม่เข้าสู่ระบบ รวมถึงยังมีไอคอนสำหรับแก้ไขข้อมูล และลบข้อมูลภาษาโปรแกรมอีกด้วย ซึ่งหน้าจอแสดงรายการภาษาโปรแกรมของระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์มีรูปแบบ ดังรูปที่ 4.3

List development language

Language	SLOC/FPs		
JAVA	150		
C	100		
C++	120		
C#.net	140		
ASP.net	125		
ADA	100		

รูปที่ 4.3 รายการข้อมูลภาษาโปรแกรมในระบบ

จากรูปที่ 4.3 เมื่อผู้ใช้ระบบกดปุ่มเพื่อเพิ่มข้อมูลภาษาโปรแกรม ระบบจะเปลี่ยนหน้าจอไปยังหน้าจอบันทึกข้อมูลภาษาโปรแกรม ดังรูปที่ 4.4 และเมื่อผู้ใช้ระบบคลิกที่ไอคอนแก้ไขข้อมูลระบบจะเปลี่ยนหน้าจอไปยังหน้าจอแก้ไขข้อมูลดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Add Development Language

รูปที่ 4.4 บันทึกข้อมูลภาษาโปรแกรม

Edit Development Language

รูปที่ 4.5 แก้ไขข้อมูลภาษาโปรแกรม

4.2.2 ข้อมูลประเภทธุรกิจ

การทำงานในส่วนนี้เป็นการทำงานเกี่ยวกับการจัดการข้อมูลประเภทธุรกิจ เพื่อใช้กำหนดหมวดหมู่ของซอฟต์แวร์ที่พัฒนาขึ้นว่าเป็นการพัฒนาระบบเกี่ยวกับธุรกิจประเภทใด และใช้เป็นข้อมูลประกอบในรายงาน โดยการทำงานในส่วนนี้จะเริ่มจากหน้าแสดงรายการประเภทธุรกิจที่บันทึกไว้ในระบบ ในหน้าจอนี้จะมีปุ่มสำหรับบันทึกประเภทธุรกิจใหม่เข้าสู่ระบบ รวมถึงยังมีไอคอนสำหรับแก้ไขข้อมูล และลบข้อมูลอีกด้วย

หน้าจอแสดงรายการประเภทธุรกิจของระบบช่วยประมาณภาระงานในการพัฒนา

ซอฟต์แวร์มีรูปแบบ ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List business type

Add		
4 items found, displaying all items.		
Business Type	Description	
Banking	ธนาคาร, Core banking	
Telecommunication	กลุ่มธุรกิจการสื่อสาร	
Finance	การเงิน, ลิขสิทธิ์, กฎหมาย, iLegals	
Government	โครงการของราชการ	

รูปที่ 4.6 รายการประเภทธุรกิจ

จากรูปที่ 4.6 เมื่อผู้ใช้ระบบคลิกปุ่มเพื่อเพิ่มข้อมูลประเภทธุรกิจ ระบบจะเปลี่ยนหน้าจอไปยังหน้าจอบันทึกข้อมูลประเภทธุรกิจ ดังรูปที่ 4.7 และเมื่อผู้ใช้ระบบคลิกที่ไอคอนแก้ไขข้อมูล ระบบจะเปลี่ยนหน้าจอไปยังหน้าจอแก้ไขข้อมูลดังรูปที่ 4.8

Add business type

Business Type *	<input type="text"/>
Description	<input type="text"/>
Back	Save Reset

รูปที่ 4.7 เพิ่มข้อมูลประเภทธุรกิจ

Edit business type

Business Type *	<input type="text" value="Banking"/>
Description	<input type="text" value="ธนาคาร, Core banking"/>
Back	Save Reset

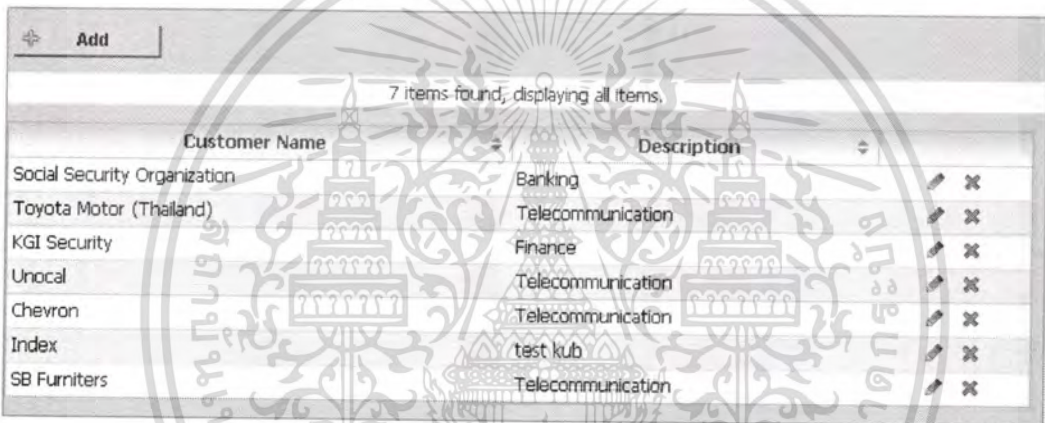
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.8 แก้ไขข้อมูลประเภทธุรกิจ ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ข้อมูลลูกค้า

การจัดการข้อมูลลูกค้าใช้สำหรับบันทึกข้อมูลของลูกค้า โดยการทำงานในส่วนนี้จะเป็นส่วนการทำงานสำหรับผู้ดูแลระบบเข้ามาใช้งานเพื่อเพิ่มข้อมูลลูกค้า รวมถึงแก้ไขข้อมูลลูกค้า โดยข้อมูลลูกค้าจะถูกบันทึกเป็นข้อมูลหนึ่งของโครงการ และใช้สำหรับเป็นเงื่อนไขในการสร้างรายงานอีกด้วย โดยการทำงานในส่วนนี้จะเริ่มจากหน้าแสดงรายการข้อมูลลูกค้าที่บันทึกไว้ในระบบ ซึ่งในหน้าจอนี้จะมีปุ่มสำหรับบันทึกข้อมูลลูกค้าใหม่เข้าสู่ระบบ รวมถึงยังมีไอคอนสำหรับแก้ไขข้อมูล และลบข้อมูลอีกด้วย

หน้าจอแสดงรายการลูกค้าของระบบช่วยประมาณการงานในการพัฒนาซอฟต์แวร์มีรูปแบบ ดังรูปที่ 4.9 และหน้าจอการบันทึกข้อมูลลูกค้าแสดงได้ ดังรูปที่ 4.10

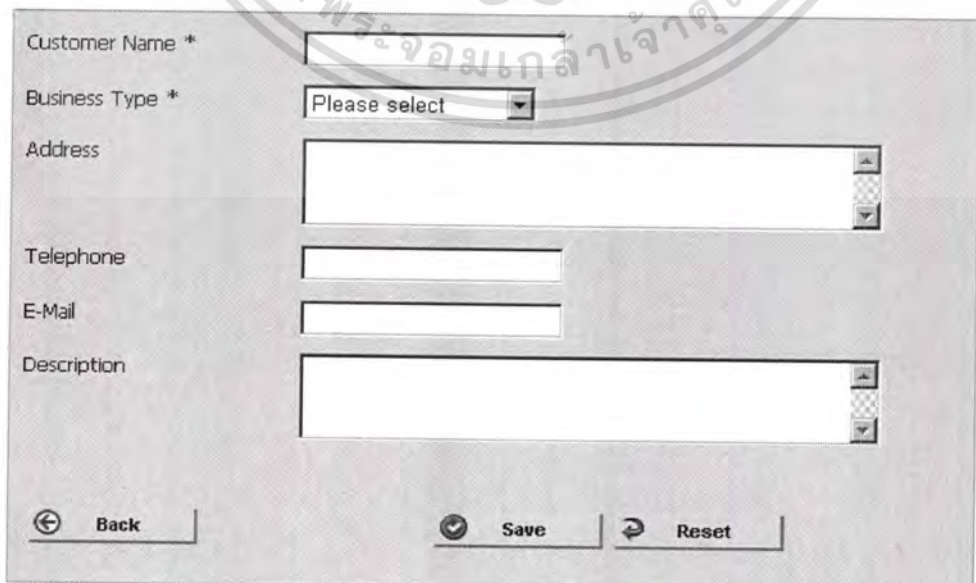
List customer



Customer Name	Description		
Social Security Organization	Banking		
Toyota Motor (Thailand)	Telecommunication		
KGI Security	Finance		
Unocal	Telecommunication		
Chevron	Telecommunication		
Index	test kub		
SB Furniters	Telecommunication		

รูปที่ 4.9 รายการข้อมูลลูกค้า

Add customer



Customer Name *

Business Type * Please select

Address

Telephone

E-Mail

Description

Back Save Reset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรูปรูปที่ 4.10 เพิ่มข้อมูลลูกค้า นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ข้อมูลผู้ใช้งานระบบ

การจัดการข้อมูลผู้ใช้งานระบบเป็นส่วนการทำงานของผู้ดูแลระบบสำหรับเข้ามาใช้งานเพื่อจัดการสิทธิ์ของผู้ใช้ระบบ ทั้งการบันทึกผู้ใช้งานใหม่เข้าสู่ระบบ แก้ไขข้อมูลส่วนตัวของผู้ใช้ระบบ ตลอดจนแก้ไขสิทธิ์ในการเข้าถึงข้อมูลของผู้ใช้แต่ละคน รวมถึงยกเลิกการระงับสิทธิ์การใช้งานของผู้ที่ถูกระงับไปได้ โดยการทำงานในส่วนนี้จะเริ่มจากหน้าแสดงรายการข้อมูลผู้ใช้งานระบบที่บันทึกไว้ในระบบดังรูปที่ 4.11 ซึ่งในหน้าจอนี้จะมีปุ่มสำหรับบันทึกข้อมูลลูกค้าใหม่เข้าสู่ระบบ รวมถึงยังมีไอคอนสำหรับแก้ไขข้อมูล และลบข้อมูลอีกด้วย

เมื่อผู้ใช้งานกดปุ่มเพื่อเพิ่มข้อมูลผู้ใช้งานระบบแล้ว ระบบจะเปลี่ยนหน้าจอไปยังหน้าจอบันทึกข้อมูลผู้ใช้งานระบบ ดังรูปที่ 4.12 และเมื่อผู้ใช้งานคลิกที่ไอคอนแก้ไขข้อมูลผู้ใช้งานระบบ ระบบจะเปลี่ยนหน้าจอไปยังหน้าจอแก้ไขข้อมูลดังรูปที่ 4.13

List user

First Name	Last Name	User Login	Department	Position	Team		
Thitichai	Wongpiya	thitichai.w	Production	System Analyst	Suchada		
Suchada	Wachirapraditporn	suchada.w	Production	Project Manager	Suchada		
Supanan	Suksupakit	supanan.s	Production	Software Engineer	Suchada		
Vorawit	Nuakoawai	vorawit.n	Production	Software Engineer	Suchada		
Wittaya	Yaowaratch	wittaya.y	Production	Project Leader	Wittaya		
Pasunan	Klina	pasunan.k	Production	Software Engineer	Wittaya		
Pisan	Sorasit	pisan.s	Production	System Analyst	Suchada		

รูปที่ 4.11 รายการข้อมูลผู้ใช้งานระบบ

Add user

Form fields and options:

- First Name * [text input]
- Last Name * [text input]
- Position * [Please select dropdown]
- Department * [Please select dropdown]
- User Name * [text input]
- Password * [password input]
- E-Mail [text input]
- Team [Please select dropdown]
- User status: Disabled all permission
- Team Leader: Team Leader
- Role:
 - System administrator
 - Project Manager
 - Viewer (Project Information)
 - Viewer (Report)

Buttons: Back, Save, Reset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.12 เพิ่มข้อมูลผู้ใช้งานระบบ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit user

First Name *	<input type="text" value="Thitichai"/>	Last Name *	<input type="text" value="Wongpiya"/>
Position *	<input type="text" value="System Analyst"/>	Department *	<input type="text" value="Production"/>
User Name *	<input type="text" value="thitichai.w"/>	Password *	<input type="password" value="....."/> <input type="button" value="Reset"/>
E-Mail	<input type="text" value="thitichai.w@motiftech.com"/>	Team	<input type="text" value="Suchada"/>
User status	<input type="checkbox"/> Disabled all permission	Team Leader	<input checked="" type="checkbox"/> Team Leader
Role	<input checked="" type="checkbox"/> System administrator <input checked="" type="checkbox"/> Project Manager <input checked="" type="checkbox"/> Viewer (Project Information) <input type="checkbox"/> Viewer (Report)		
<input type="button" value="Back"/>		<input type="button" value="Save"/> <input type="button" value="Reset"/>	

รูปที่ 4.13 แก้ไขข้อมูลผู้ใช้ระบบ

4.3 การออกแบบหน้าจอการปรับปรุงพารามิเตอร์

หน้าจอในส่วนของการปรับปรุงพารามิเตอร์นั้นเป็นอีกส่วนการทำงานของผูดูแลระบบ ในการปรับปรุ่ค่าของพารามิเตอร์ที่ใช้ในแบบจำลองต่างๆ ในการประเมินภาระงาน เพื่อให้ผลการประมาณภาระงานใกล้เคียงกับการทำงานของบริษัทมากที่สุด และในส่วนของการปรับปรุงพารามิเตอร์เองได้แบ่งส่วนของการปรับปรุงพารามิเตอร์ตามแบบจำลองที่ใช้ในการประมาณภาระงานของระบบย่อยลงไปได้อีก 3 แบบจำลองคือ แบบจำลอง COCOMO II แบบจำลองฟังก์ชันพอยต์ และแบบจำลองอ็อบเจกต์พอยต์

4.3.1 การปรับปรุ่ค่าของแบบจำลอง COCOMO II

ในระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์นั้นได้เลือกใช้แบบจำลอง COCOMO II ในการประมาณภาระงาน โดยแบบจำลอง COCOMO II เป็นแบบจำลองทางคณิตศาสตร์ (Algorithmic/Parametric Model) ซึ่งนำข้อมูลที่ผู้บริหารโครงการบันทึกเข้าสู่สมการที่ใช้ในแบบจำลองเพื่อประเมินขนาดของโครงการ ดังนั้นเพื่อที่จะทำให้การประเมินภาระงานของระบบมีความใกล้เคียงกับการทำงานจริง จะต้องมีกรปรับปรุ่ค่าพารามิเตอร์ต่างๆ ให้มีความเหมาะสม โดยในส่วนของแบบจำลอง COCOMO II นั้น ได้มีการออกแบบหน้าจอสำหรับปรับปรุ่ค่าพารามิเตอร์ต่างๆ ดังนี้

4.3.1.1 การปรับปรุ่ค่าในสมการ (Equation)

หน้าจอการปรับปรุ่ค่าในสมการของ COCOMO II เป็นหน้าจอปรับปรุ่ค่าพารามิเตอร์ที่จะใช้ในสมการของ COCOMO II ทั้งในการประมาณภาระงาน (Effort) ระยะเวลาการคำนวณส่วนเบี่ยงเบนค่าที่ส่งคืนไว้สำหรับการเชิงเส้นเพื่อแก้ไขค่าที่มีอยู่และในการคำนวณค่าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการดำเนินโครงการ (TDEV, Schedule) และค่าอัตราส่วนของปัจจัยที่มีผลต่อขนาดโครงการ (B) โดยในทุกครั้งที่มีการปรับปรุค่าพารามิเตอร์ ระบบจะบันทึกประวัติการปรับปรุงเอาไว้ เพื่อเป็นข้อมูลในการอ้างอิงและตรวจสอบการแก้ไขข้อมูลโดยมีรูปแบบหน้าจอ ดังรูปที่ 4.14

COCOMO II Equation

Effort = a * (size) ^B		
a :	<input type="text" value="2.5"/>	
B = x + y * Σ(W _i)		
x :	<input type="text" value="1.01"/>	y : <input type="text" value="0.01"/>
TDEV = (w * PM (**y*(B-z))) * SCED		
w :	<input type="text" value="3.0"/>	x : <input type="text" value="0.33"/>
y :	<input type="text" value="0.2"/>	z : <input type="text" value="1.01"/>

Buttons: Back, Save, Reset

History	
Updated date	Updated user
7 Oct 2008 16:45	ฉัตรชัย วงศ์ปิยะ
19 Nov 2008 10:13	ฉัตรชัย วงศ์ปิยะ

รูปที่ 4.14 หน้าจอปรับปรุค่าพารามิเตอร์ที่ใช้ในแบบจำลอง COCOMO II

4.3.1.2 การปรับปรุค่า Scale Factor

ปัจจัยมาตราส่วน (Scale Factor) เป็นพารามิเตอร์ที่แสดงถึงปัจจัยที่ส่งผลต่อการขยายตัวของโครงการที่ประมาณการงาน โดยปัจจัยมาตราส่วนจะประกอบไปด้วย 5 ปัจจัย ได้แก่

1. ปัจจัยการมีแบบอย่างของการพัฒนามาก่อนหน้านี้ (Precedentedness: PREC) ในโครงการที่มีการพัฒนาซอฟต์แวร์ในรูปแบบที่คล้ายคลึงกับของเดิม ปัจจัยนี้จะมีค่าสูง
2. ปัจจัยความยืดหยุ่นในการพัฒนา (Development Flexibility: FLEX) ปัจจัยด้านความยืดหยุ่นจะมีค่าสูง ในกรณีที่ไม่จำเป็นต้องทำงานภายใต้ข้อกำหนด (Specification) ภายนอก
3. ปัจจัยด้านสถาปัตยกรรมและการแก้ไขความเสี่ยง (Architecture/Risk Resolution: RESL) ปัจจัยข้อนี้จะมีค่าสูงเมื่อมีการพัฒนาซอฟต์แวร์ภายใต้สถาปัตยกรรมที่มีอยู่แล้ว
4. ปัจจัยด้านการทำงานร่วมกัน (Team Cohesion: TEAM) ปัจจัยที่พิจารณาถึงระดับความยากง่ายในการประสานงานกันของบุคลากรในโครงการ
5. ปัจจัยด้านวุฒิภาวะการควบคุมกระบวนการพัฒนา (Process Maturity: PMAT) ปัจจัยที่พิจารณาขีดความสามารถด้านการควบคุมกระบวนการจากมาตรฐาน CMM หรือ CMMI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบหน้าจอของการปรับปรุงพารามิเตอร์ปัจจัยมาตราส่วน (Scale Factor) สามารถแสดงได้ดังรูปที่ 4.15 โดยในหน้าจอก็จะมีการระบุค่าปัจจัยต่างๆ ตามระดับของปัจจัยนั้นๆ จากน้อยที่สุด (Very Low) ไปจนถึงสูงที่สุด (Extra High) และมีช่องทางสำหรับระบุค่าปริยาย (Default) สำหรับใช้ในกรณีที่ไม่ได้ระบุปัจจัยดังกล่าวมาในขั้นตอนบันทึกข้อมูลโครงการในการประมาณภาระงาน และในทุกครั้งที่มีการปรับค่าพารามิเตอร์ ระบบจะบันทึกประวัติการปรับปรุงเอาไว้ เพื่อเป็นข้อมูลในการอ้างอิงและตรวจสอบการแก้ไขข้อมูล

Scale Factor

	Very Low	Low	Nominal	High	Very High	Extra High	Default
Precedentedness	4.05	3.24	2.43	1.62	0.81	0.00	----
Development Flexibility	6.07	4.86	3.64	2.43	1.21	0.00	----
Architecture / Risk solution	4.22	3.38	2.53	1.69	0.84	0	----
Team Cohesion	4.94	3.95	2.97	1.98	0.99	0.00	----
Process Maturity	4.54	3.64	2.73	1.82	0.91	0.00	----

Back Save Reset

History

Updated date	Updated user
7 Oct 2008 16:45	ธิดิษฐ์ วงศ์ปิยะ
19 Nov 2008 10:13	ธิดิษฐ์ วงศ์ปิยะ

รูปที่ 4.15 หน้าจอการปรับค่า Scale Factor

4.3.1.3 การปรับค่า Cost Driver

การออกแบบหน้าจอของการปรับปรุงพารามิเตอร์ตัวขับเคลื่อนด้านต้นทุน (Cost Driver) ในหน้าจอก็จะเป็นการระบุค่าปัจจัยต่างๆ ตามระดับของปัจจัยนั้นๆ จากน้อยที่สุด (Very Low) ไปจนถึงสูงที่สุด (Extra High) และมีช่องทางสำหรับระบุค่าปริยาย (Default) สำหรับใช้ในกรณีที่ไม่ได้ระบุปัจจัยดังกล่าวมาในขั้นตอนบันทึกข้อมูลโครงการในการประมาณภาระงาน และในทุกครั้งที่มีการปรับค่าพารามิเตอร์ ระบบจะบันทึกประวัติการปรับปรุงเอาไว้ เพื่อเป็นข้อมูลในการอ้างอิงและตรวจสอบการแก้ไขข้อมูล

สำหรับการออกแบบหน้าจอของการปรับค่าตัวขับเคลื่อนด้านต้นทุนสามารถแสดงได้ดังรูปที่ 4.16

Cost Driver

Effort multiplier for : **ACAP - Analyst Capability Cost Driver**

	Intermediate	Maintenance	Product Design	Detail Design	Code & Unit test	Integration test	Default
Extra Low	4.05	3.24	2.43	1.62	0.81	0.00	—
Very Low	6.07	4.86	3.64	2.43	1.21	0.00	—
Low	4.22	3.38	2.53	1.69	0.84	0	—
Nominal	4.94	3.95	2.97	1.98	0.99	0.00	—
High	4.54	3.64	2.73	1.82	0.91	0.00	—
Very High	4.54	3.64	2.73	1.82	0.91	0.00	—
Extra High	4.54	3.64	2.73	1.82	0.91	0.00	—

Back Save Reset

History

Updated date	Updated user
7 Oct 2008 16:45	ชิตชัย วงศ์ปิยะ
19 Nov 2008 10:13	ชิตชัย วงศ์ปิยะ

รูปที่ 4.16 หน้าจอแสดงรายการ Cost Driver

4.3.2 การปรับปรุ่ค่าของแบบจำลองฟังก์ชันพอยต์

4.3.2.1 การปรับปรุ่ค่าน้ำหนัก (Weight)

การออกแบบหน้าจอของการปรับปรุ่พารามิเตอร์ค่าน้ำหนักของฟังก์ชันพอยต์สามารถแสดงได้ดังรูปที่ 4.17 โดยในหน้าจอจะมีการระบุค่าน้ำหนักต่างๆ ตามระดับของปัจจัยนั้นๆ

Function Point Weight

	Simple	Average	Complex
External Input	3	4	6
External Output	4	5	7
Logical Internal File	7	10	15
External Interface File	5	7	10
External Inquiry	3	4	6

Back Save Reset

History

Updated date	Updated user
7 Oct 2008 16:45	ชิตชัย วงศ์ปิยะ
19 Nov 2008 10:13	ชิตชัย วงศ์ปิยะ

รูปที่ 4.17 หน้าจอปรับปรุ่ค่าน้ำหนักของฟังก์ชันพอยต์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การปรับปรุงค่าของแบบจำลองอ็อบเจกต์พอยต์

4.3.3.1 การปรับปรุงค่าน้ำหนัก (Weight)

การออกแบบหน้าจอของการปรับปรุงพารามิเตอร์ค่าน้ำหนักของอ็อบเจกต์พอยต์สามารถแสดงได้ดังรูปที่ 4.18 โดยในหน้าจอนี้จะมีการระบุค่าน้ำหนักต่างๆ ตามระดับของปัจจัยนั้นๆ และในทุกครั้งที่มีการปรับปรุงค่าพารามิเตอร์ ระบบจะบันทึกประวัติการปรับปรุงเอาไว้ เพื่อเป็นข้อมูลในการอ้างอิงและตรวจสอบการแก้ไขข้อมูล

Object Point Weight

	Simple	Average	Complex
Screen Display	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="3"/>
Report	<input type="text" value="2"/>	<input type="text" value="5"/>	<input type="text" value="8"/>
3GL Component	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="10"/>

History

Updated date	Updated user
7 Oct 2008 16:45	อติชัย วงศ์ปิยะ
19 Nov 2008 10:13	อติชัย วงศ์ปิยะ

รูปที่ 4.18 หน้าจอปรับปรุงค่าน้ำหนักของอ็อบเจกต์พอยต์

4.4 การออกแบบหน้าจอการบันทึกข้อมูลโครงการ

การทำงานในส่วนนี้เป็นการทำงานของผู้บริหารโครงการในการบันทึกข้อมูลของโครงการ และข้อมูลประกอบอื่นๆ เช่น เอกสารแนบของโครงการ ข้อมูลผู้ติดต่อ และประเด็นต่างๆ ที่เกิดขึ้นระหว่างการดำเนินงานของโครงการ รวมถึงประเมินภาระงานที่จะใช้ในการพัฒนาซอฟต์แวร์ โดยมีหน้าจอการทำงานหลัก ดังนี้

4.4.1 การบันทึกข้อมูลโครงการ

หน้าจอการบันทึกข้อมูลโครงการ เป็นหน้าจอสำหรับผู้บริหารโครงการใช้ในการบันทึกข้อมูลทั่วไปของโครงการที่ต้องการประเมินภาระงาน ซึ่งออกแบบหน้าจอ ได้ดังรูปที่ 4.19

Add project

Project Name *	<input type="text"/>	Project Code *	<input type="text"/>
Customer *	<input type="text" value="Please select"/>	Project Line *	<input type="text" value="Please select"/>
Project Type *	<input type="text" value="Please select"/>	Development Language *	<input type="text" value="Please select"/>
Contract No. *	<input type="text"/>	Contract Start date *	<input type="text"/>
Contract Due Date *	<input type="text"/>	Kick-off Date *	<input type="text"/>
Budget *	<input type="text" value="0.0"/>		
Description	<input type="text"/>		
<input type="button" value="Back"/>		<input type="button" value="Save"/> <input type="button" value="Reset"/>	

รูปที่ 4.19 หน้าจอบันทึกข้อมูล โครงการ

4.4.2 รายละเอียดโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับผู้บริหาร โครงการ ใช้ในการบันทึกข้อมูลการดำเนินงานของโครงการ เช่น สถานะของโครงการ หรือค่าใช้จ่ายที่เกิดขึ้นจริง เป็นต้น โดยในการทำงานของหน้าจอนี้ ยังสามารถเชื่อมโยงไปยังหน้าจอการประเมินภาระงาน หน้าจอแสดงรายละเอียดของสภาพแวดล้อมที่ใช้ในโครงการ หน้าจอผู้มีส่วนร่วมในโครงการ หน้าจอประเด็นต่างๆ ที่เกิดขึ้นในโครงการ และหน้าจอเอกสารแนบ โดยแสดงตัวอย่างหน้าจอ ได้ดังรูปที่ 4.20

Manage project

Project Information	Estimation	Environment	Contact	Issues	Attachment
Project Name	Imported Part Inventory Simulation System		Project Code	NS-TMT-IPIS	
Customer	Toyota Motor (Thailand)		Project Line	Logistics	
Project Type	New System		Development Language	JAVA	
Contract No.	MT-CT-001		Contract Start date	12/01/2009	
Contract Due Date	13/07/2009		Kick-off Date	31/01/2009	
Budget	4,525,000.00				
Description	For mass production, component part planning is very important, especially imported parts. However, the current imported part inventory control and monitor for TMT relies on semi-manual process using Excel as a main tool to prevent part shortage in production.				
Project Status *	<input type="text" value="Please select"/>		Actual Cost	<input type="text"/>	
Estimated Cost	3,500,000.00		Estimated Finish Date	30/06/2009	
Actual Start Date	<input type="text"/>		Actual Finish Date	<input type="text"/>	
<input type="button" value="Back"/>		<input type="button" value="Save"/> <input type="button" value="Reset"/>			

รูปที่ 4.20 หน้าจอแสดงรายละเอียดโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 การประเมินภาระงาน

หน้าจอนี้เป็นหน้าจอสำหรับผู้บริหารโครงการใช้ในการประเมินภาระงานของโครงการ โดยในการประเมินภาระงานนั้น จะต้องบันทึกข้อมูลที่ใช้ในการประเมินภาระงานของแบบจำลอง COCOMO II ก่อน ได้แก่ ขนาดของโครงการ (ข้อมูลฟังก์ชันพอยต์ หรืออ็อบเจกต์พอยต์) ข้อมูลปัจจัยที่ส่งผลต่อขนาดของโครงการ (Scale Factor) และข้อมูลตัวขับเคลื่อนด้านต้นทุน (Cost Driver) เมื่อได้บันทึกข้อมูลที่ใช้ในการประเมินภาระงานแล้ว ระบบจะประเมินภาระงานของโครงการออกมาในรูปของขนาดโครงการในหน่วย Person-month และระยะเวลาที่ใช้ในการดำเนินงาน (Schedule) โดยแสดงหน้าจอการประเมินภาระงาน ได้ดังรูปที่ 4.21

Project Estimation

Effort	Schedule	Cost	Updated date	Updated user
24.3	5.2	1,649,393	7 Oct 2008 16:45	ธิดิษฐ์ วงศ์ปิยะ
25.7	4.1	1,259,393	19 Nov 2008 10:13	ธิดิษฐ์ วงศ์ปิยะ

รูปที่ 4.21 หน้าจอการประเมินภาระงาน

4.4.4 การคำนวณขนาดของโครงการโดยวิธีฟังก์ชันพอยต์

หน้าจอนี้เป็นหน้าจอสำหรับคำนวณหาขนาดของโครงการ โดยวิธีฟังก์ชันพอยต์ ผู้บริหารโครงการจะบันทึกจำนวนของงานต่างๆ แบ่งตามระดับความยากตั้งแต่ ง่าย ปานกลาง ไปจนถึงการทำงานที่ซับซ้อน จากนั้นระบบจะคำนวณค่าฟังก์ชันพอยต์และคูณเข้ากับจำนวนบรรทัดของภาษาที่ใช้ในโครงการ เพื่อให้ได้ขนาดของโครงการในรูปแบบของจำนวนบรรทัดในการเขียนโปรแกรม (Source Line of Code: SLOC) โดยแสดงหน้าจอการทำงาน ได้ดังรูปที่ 4.22

4.4.5 การคำนวณขนาดของโครงการโดยวิธีอ็อบเจกต์พอยต์

หน้าจอนี้เป็นหน้าจอสำหรับคำนวณหาขนาดของโครงการ โดยวิธีอ็อบเจกต์พอยต์ ผู้บริหารโครงการจะบันทึกจำนวนของงานต่างๆ แบ่งตามระดับความยากตั้งแต่ ง่าย ปานกลาง ไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จนถึงการทำงานที่ซับซ้อน จากนั้นระบบจะคำนวณค่าอ็อบเจกต์พอยต์และคูณเข้ากับจำนวนบรรทัดของภาษาที่ใช้ในโครงการ เพื่อให้ได้ขนาดของโครงการในรูปแบบของจำนวนบรรทัดในการเขียนโปรแกรม (Source Line of Code: SLOC) โดยแสดงหน้าจอกการทำงาน ได้ดังรูปที่ 4.23

Project Function Point

	Simple	Average	Complex
External Input	<input type="text" value="3"/>	<input type="text" value="4"/>	<input type="text" value="6"/>
External Output	<input type="text" value="4"/>	<input type="text" value="5"/>	<input type="text" value="7"/>
Logical Internal File	<input type="text" value="7"/>	<input type="text" value="10"/>	<input type="text" value="15"/>
External Interface File	<input type="text" value="5"/>	<input type="text" value="7"/>	<input type="text" value="10"/>
External Inquiry	<input type="text" value="3"/>	<input type="text" value="4"/>	<input type="text" value="6"/>

Development Language: JAVA
 LOC/FP: 150
 Function Point Score:
 SLOC:

History

	Function Point	SLOC	Updated date	Updated user
1	1	1	7 Oct 2008 16:45	ธิดิษฐ์ วงศ์ปิยะ
2	1	1	19 Nov 2008 10:13	ธิดิษฐ์ วงศ์ปิยะ

รูปที่ 4.22 หน้าจอกำหนดขนาดของโครงการด้วยวิธีฟังก์ชันพอยต์

Project Object Point

	Simple	Average	Complex
Screen Display	<input type="text" value="3"/>	<input type="text" value="4"/>	<input type="text" value="6"/>
Report	<input type="text" value="4"/>	<input type="text" value="5"/>	<input type="text" value="7"/>
3GL Component	<input type="text" value="7"/>	<input type="text" value="10"/>	<input type="text" value="15"/>

Development Language: JAVA
 LOC/FP: 150
 Function Point Score:
 SLOC:

History

	Object Point	SLOC	Updated date	Updated user
1	1	1	7 Oct 2008 16:45	ธิดิษฐ์ วงศ์ปิยะ
2	2	2	19 Nov 2008 10:13	ธิดิษฐ์ วงศ์ปิยะ

รูปที่ 4.23 หน้าจอกำหนดขนาดของโครงการด้วยวิธีอ็อบเจกต์พอยต์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับการใช้งานเพื่อการศึกษาเท่านั้น มิได้อนุญาตให้เผยแพร่ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.6 การคำนวณค่าปัจจัยที่ส่งผลต่อขนาดของโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับบันทึกค่าปัจจัยที่ส่งผลต่อขนาดของโครงการ ผู้บริหารโครงการจะบันทึกค่าปัจจัยต่างๆ ที่เกิดขึ้นในโครงการนั้น เพื่อใช้เป็นข้อมูลในการประเมินภาระงานตามแบบจำลอง COCOMO II โดยแสดงหน้าจอการทำงาน ได้ดังรูป 4.24

Project Scale Factor

PREC	FLEX	RESL	TEAM	PMAT	Total	Updated date	Updated user
Norminal	Norminal	Norminal	High	Low	16	7 Oct 2008 16:45	ฉัตรชัย วงศ์ปิยะ
Norminal	Norminal	High	Very High	Low	20	19 Nov 2008 10:13	ฉัตรชัย วงศ์ปิยะ

รูปที่ 4.24 หน้าจอการคำนวณค่าปัจจัยที่ส่งผลต่อขนาดของโครงการ

4.4.7 การคำนวณค่าปัจจัยที่ส่งผลต่อต้นทุนของโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับบันทึกค่าปัจจัยที่ส่งผลต่อต้นทุนของโครงการ ผู้บริหารโครงการจะบันทึกค่าปัจจัยต่างๆ ที่เกิดขึ้นในโครงการนั้น เพื่อใช้เป็นข้อมูลในการประเมินภาระงานตามแบบจำลอง COCOMO II โดยแสดงหน้าจอการทำงาน ได้ดังรูป 4.25

4.4.8 การบันทึกสภาพแวดล้อมของโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับบันทึกสภาพแวดล้อมต่างๆ ที่ใช้ในโครงการ เช่น ฐานข้อมูลเว็บเซิร์ฟเวอร์ หรือเครื่องมืออื่นๆ โดยแสดงหน้าจอการทำงาน ได้ดังรูป 4.26

4.4.9 การบันทึกข้อมูลผู้มีส่วนร่วมในโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับบันทึกข้อมูลของผู้มีส่วนร่วมในโครงการ เช่น ลูกค้า หรือที่ปรึกษาโครงการ โดยแสดงหน้าจอการทำงาน ได้ดังรูป 4.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Cost Driver

Analyst capability (ACAP) How capable are the analysts for this project?	Norminal
Application experience (APEX) How much experience does your team have with this type of application?	Norminal
Programmer capability (PCAP) How capable are the programmers for this project?	Norminal
Language and tools experience (LTEX) How much experience does your team have with the language and tools?	Norminal
Development Flexibility (FLEX) Are your requirement flexible?	Norminal
Personel continuity (PCON) What is the annual turnover rate for your organization?	Norminal
Use of Software Tools (TOOL) What tools will your team use?	Norminal
Multisite Development (SITE) Is the team split among several sites?	Norminal
Development schedule (SCED) Is the schedule compressed from the nominal schedule?	Norminal
Execution time constraint (TIME) How much CPU time will your software use?	Norminal
Main storage constraint (STOR) How much of main memory will your software use?	Norminal
Platform volatility (PVOL) How often will the platform (OS, DBMS, etc.) change?	Norminal
Required reliability (RELY) What is the consequence of a software failure?	Norminal
Database size (DATA) How much data is required to test the software?	Norminal
Product complexity (CPLX) How complex will the software be?	Norminal
Required reusability (RUSE) Are you developing your software components to be reused?	Norminal
Documentation match to life-cycle needs (Docu) How much documentation are you creating?	Norminal
Total Multiplier	15.0

History

Total	Updated date	Updated user
16	7 Oct 2008 16:45	ธิดิษฐ์ วงศ์ปิยะ
20	19 Nov 2008 10:13	ธิดิษฐ์ วงศ์ปิยะ

รูปที่ 4.25 หน้าจอการคำนวณค่าปัจจัยที่ส่งผลต่อต้นทุนของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Add Project's Environment

Name *

Version

Type Please Select

Description

Back Save Reset

รูปที่ 4.26 หน้าจอบันทึกสภาพแวดล้อมของโครงการ

Add Project's Contact

Name *

Position

Department

Telephone

E-Mail

Back Save Reset

รูปที่ 4.27 หน้าจอบันทึกข้อมูลผู้มีส่วนร่วมในโครงการ

4.4.10 การบันทึกประเด็นที่เกิดขึ้นในโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับบันทึกข้อมูลประเด็นต่างๆ ที่เกิดขึ้นในการดำเนินงานของโครงการ เช่น แจ้งผลการทำงานผิดพลาดของระบบ (Bug) งานที่จะต้องทำ (Task) หรือข้อมูลทั่วไปของโครงการ (Info) โดยแสดงหน้าจอการทำงาน ได้ดังรูป 4.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Add project's issue

รูปที่4.28 หน้าจอบันทึกประเด็นที่เกิดขึ้นในโครงการ

4.4.11 การบันทึกเอกสารแนบของโครงการ

หน้าจอนี้เป็นหน้าจอสำหรับบันทึกเอกสารแนบของโครงการ เพื่อใช้เป็นข้อมูลอ้างอิงในการดำเนินงานของโครงการ โดยแสดงหน้าจการทำงาน ได้ดังรูป 4.29

Add project's attachment

รูปที่4.29 หน้าจอบันทึกเอกสารแนบของโครงการ

4.5 การออกแบบหน้าจอรายงาน

การทำงานในส่วนนี้เป็นการทำงานของผู้บริหารของบริษัท และผู้บริหารโครงการในการเรียกดูรายงานสรุปผลการดำเนินงาน เพื่อใช้เป็นข้อมูลในการกำหนดแนวทางการทำงานของบริษัท และโครงการ โดยมีตัวอย่างรายงาน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.1 รายงานผลการดำเนินโครงการ

รายงานนี้เป็นรายงานสำหรับแสดงผลการดำเนินโครงการ โดยจะแสดงข้อมูลของโครงการตามเงื่อนไขที่ระบุ และสำหรับโครงการที่สิ้นสุดการดำเนินงานแล้ว จะแสดงข้อมูลการเปรียบเทียบระหว่างการดำเนินงานจริงกับผลการประมาณการะงานของโครงการ ว่ามีความคลาดเคลื่อนมากน้อยอย่างไร โดยแสดงตัวอย่างรายงาน ได้ดังรูป 4.30 และผลการประเมินภาระงานในรายงานนี้ จะขึ้นอยู่กับค่าพารามิเตอร์ที่ใช้ในแบบจำลอง เมื่อมีการปรับปรุงค่าพารามิเตอร์ ผลของการประเมินภาระงานในแต่ละโครงการก็จะแตกต่างกัน

Project Code	Customer	Type	Start Date	Finish Date	Duration	Estimated Schedule	Diff(%)	Effort	Estimated Effort	Diff(%)
NS-PJ1	Customer1	New System	01/07/2008	06/02/2009	7.25	6.13	-18.27%	60.11	42.21	-42.41%
MA-PJ2	Customer2	New System	01/10/2008	-	0	8.27	-	-	60.17	-
NS-PJ3	Customer3	Customize Product	01/10/2008	-	0	3.54	-	-	15.8	-
PD-PD1	Customer4	Maintainance	29/11/2008	28/12/2008	1	1.17	14.53%	0.6	0.69	13.04%

รูปที่4.30 ตัวอย่างรายงานผลการดำเนินโครงการ

4.5.2 รายงานสรุปปัจจัยที่มีผลในการดำเนินงาน

รายงานนี้เป็นรายงานสำหรับสรุปปัจจัยที่ส่งผลต่อการดำเนินโครงการ โดยนำข้อมูลปัจจัยต่างๆ มาจากการบันทึกประเด็นในการดำเนินโครงการ เพื่อนำข้อมูลนี้ไปใช้ในการปรับปรุงการทำงานของบริษัทต่อไป โดยแสดงตัวอย่างรายงาน ได้ดังรูป 4.31

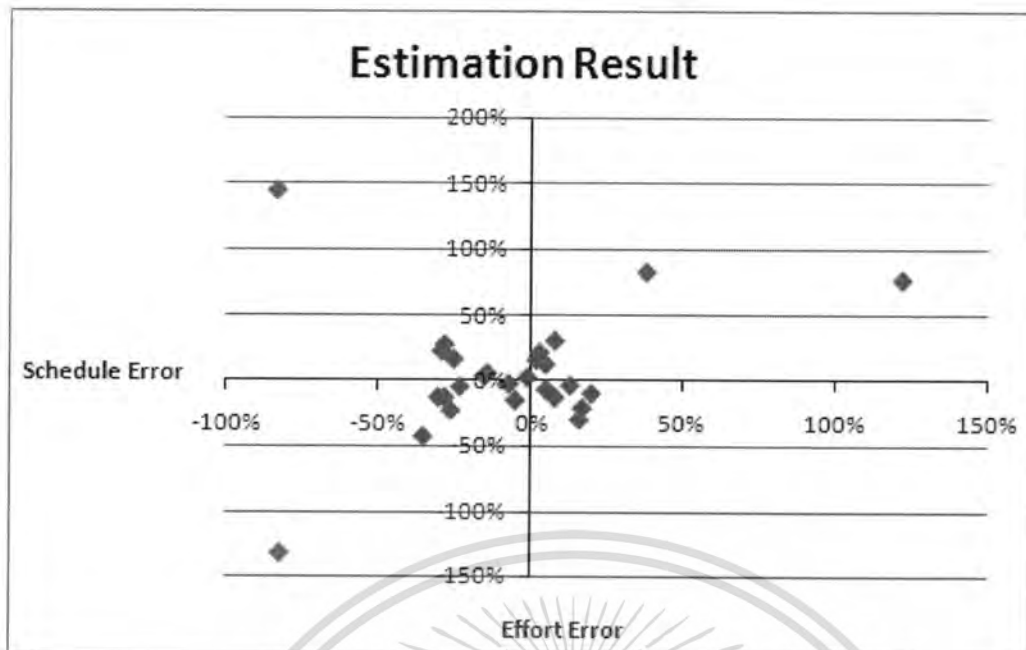
Type	Code	Project Code	Topic	Priority	Status	Factor	Time spent
Info	INF-001	NS-PJ1	Warranty period	Low	Closed	-	-
Bug	BUG-002	MA-PJ2	ปัญหาเกี่ยวกับ Error OutOfMemory ของ IES Server	High	Closed	TIME	1
Task	TSK-003	NS-PJ3	Hutch ขอวิธีการทำให้ Provisioning ทำการ Feed ข้อมูลจากBackup File	Normal	Open	-	-
Change	CHG-004	PD-PD1	Idoc ไซร้นัดนำเดี่ยว ขอวิธีการแก้ไขให้สามารถแยกหน้าได้	High	Closed	ACAP	3

รูปที่4.31 ตัวอย่างรายงานสรุปปัจจัยที่มีผลในการดำเนินงาน

4.5.3 รายงานสรุปผลการประเมินภาระงาน

รายงานนี้เป็นรายงานสำหรับสรุปผลการประเมินภาระงาน โดยนำค่าความคลาดเคลื่อนในการประเมินภาระงาน และระยะเวลาในการดำเนินโครงการ มาเปรียบเทียบในรูปของแผนภูมิเพื่อเป็นข้อมูลให้ผู้บริหารโครงการสามารถประเมินค่าและปรับปรุงค่าพารามิเตอร์ของโครงการให้มีความเหมาะสมยิ่งขึ้น โดยแสดงตัวอย่างรายงาน ได้ดังรูป 4.32 และผลการประเมินภาระงานในรายงานนี้ จะขึ้นอยู่กับค่าพารามิเตอร์ที่ใช้ในแบบจำลอง เมื่อมีการปรับปรุงค่าพารามิเตอร์ ผลของการประเมินภาระงานในแต่ละโครงการก็จะแตกต่างกัน

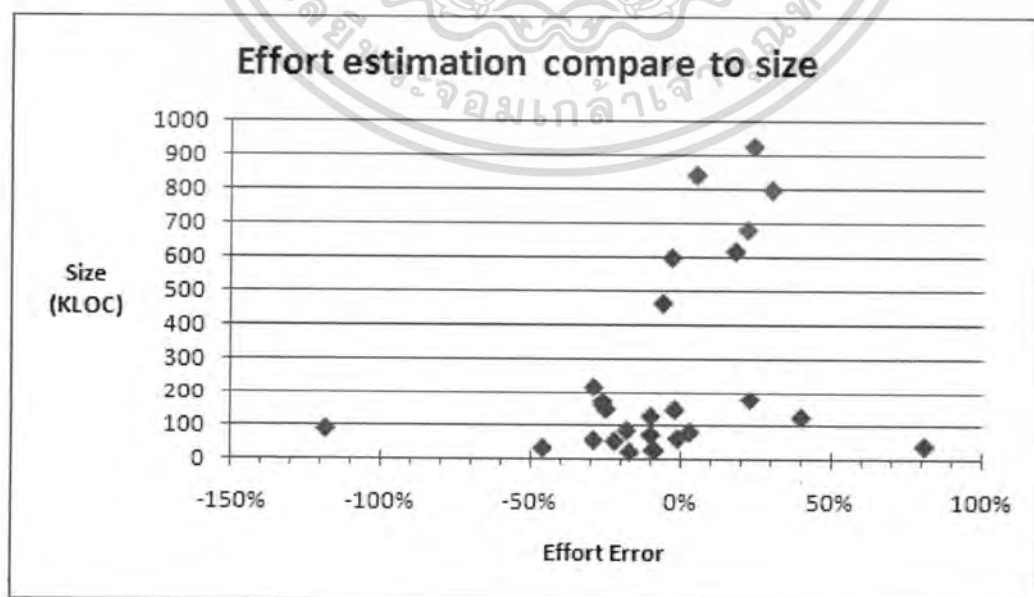
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 ตัวอย่างรายงานสรุปผลการประเมินภาระงาน

4.5.4 รายงานเปรียบเทียบการประเมินภาระงานต่อขนาดของโครงการ

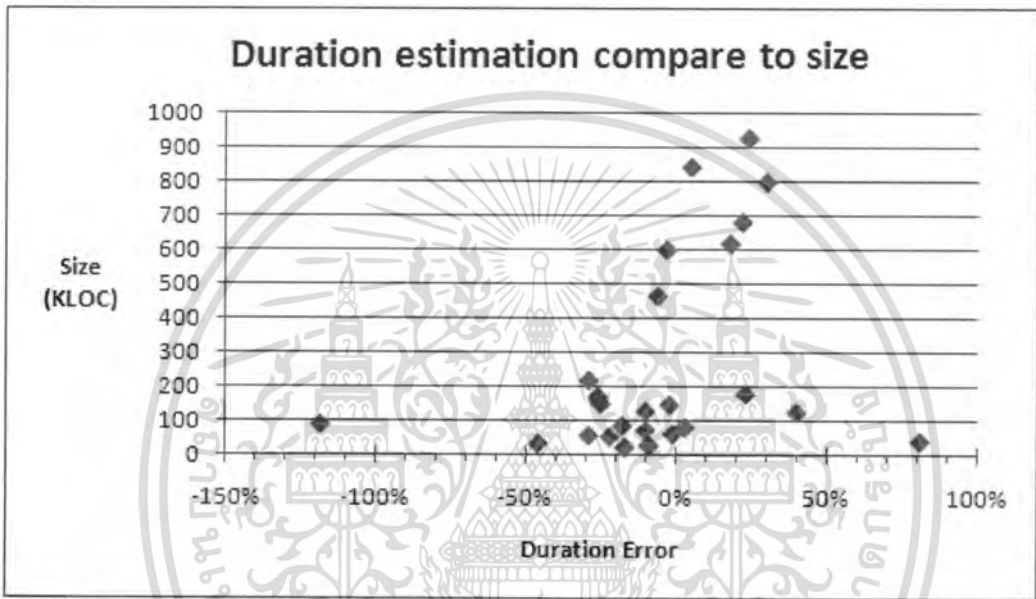
รายงานนี้เป็นรายงานเปรียบเทียบผลการประเมินภาระงานต่อขนาดของโครงการ โดยนำค่าความคลาดเคลื่อนในการประเมินภาระงาน และขนาดของโครงการในรูปแบบของจำนวนบรรทัด (Source Line of Code: SLOC) มาเปรียบเทียบในรูปของแผนภูมิ เพื่อเป็นข้อมูลให้ผู้บริหารโครงการสามารถประเมินค่าและปรับปรุงค่าพารามิเตอร์ของโครงการ ให้มีความเหมาะสมยิ่งขึ้น โดยแสดงตัวอย่างรายงาน ได้ดังรูป 4.33



เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศและการสื่อสาร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.5 รายงานเปรียบเทียบการประเมินระยะเวลาต่อขนาดของโครงการ

รายงานนี้เป็นรายงานเปรียบเทียบผลการประเมินระยะเวลาของโครงการต่อขนาดของโครงการ โดยนำค่าความคลาดเคลื่อนในการประเมินระยะเวลา และขนาดของโครงการในรูปแบบของจำนวนบรรทัด (Source Line of Code: SLOC) มาเปรียบเทียบในรูปของแผนภูมิ เพื่อเป็นข้อมูลให้ผู้บริหาร โครงการสามารถประเมินค่าและปรับปรุงค่าพารามิเตอร์ของ โครงการให้มีความเหมาะสมยิ่งขึ้น โดยแสดงตัวอย่างรายงาน ได้ดังรูป 4.34



รูปที่ 4.34 ตัวอย่างรายงานเปรียบเทียบการประเมินระยะเวลาต่อขนาดของโครงการ

บทที่ 5

บทสรุป

5.1 สรุปโครงการพัฒนาระบบงาน

ในปัจจุบันการดำเนินธุรกิจในยุคอุตสาหกรรมต่างๆ ล้วนมีการนำระบบสารสนเทศเข้ามาช่วยสนับสนุนการทำงาน เพื่อเพิ่มข้อได้เปรียบในการแข่งขันในอุตสาหกรรม และลดค่าใช้จ่ายในการดำเนินกิจการ จึงส่งผลดีต่อการดำเนินธุรกิจพัฒนาซอฟต์แวร์ แต่อย่างไรก็ดีในการดำเนินธุรกิจพัฒนาซอฟต์แวร์นั้น ยังพบว่ามีปัญหาหนึ่งซึ่งเกิดขึ้นบ่อย คือการส่งมอบงานล่าช้าเกินกำหนดสัญญาของโครงการ ซึ่งนอกจากจะบั่นทอนความน่าเชื่อถือในสายตาของลูกค้าแล้ว ยังส่งผลให้ค่าใช้จ่ายของโครงการเพิ่มสูงขึ้นอีกด้วย สาเหตุของปัญหาดังกล่าวอาจเกิดได้จากหลายปัจจัย เช่น ความเข้าใจความต้องการของระบบคลาดเคลื่อนจากผู้ให้ข้อมูล ความซับซ้อนของธุรกิจ แม้กระทั่งเทคโนโลยีที่เปลี่ยนแปลงไปอย่างรวดเร็ว การที่จะลดปัญหาดังกล่าว ผู้ทำหน้าที่ในการวางแผนโครงการจึงควรที่จะสามารถประเมินขนาดของโครงการให้สอดคล้องและใกล้เคียงกับความเป็นจริงมากที่สุดเพื่อให้ได้ระยะเวลาดำเนินโครงการที่เหมาะสม

ระบบช่วยประมาณการงานในการพัฒนาซอฟต์แวร์จึงถูกพัฒนาขึ้นเพื่อช่วยการบริหารโครงการในด้านการประมาณการงานหรือการประเมินขนาดของโครงการ ซึ่งการทำงานของระบบนั้นประกอบไปด้วยการบันทึกข้อมูลโครงการ การประมาณการงาน การจัดการข้อมูลตั้งต้นของระบบ การปรับปรุงค่าพารามิเตอร์ต่างๆ ที่ใช้ในแบบจำลองการประมาณการงาน รวมถึงสร้างรายงานเพื่อใช้บริหารโครงการ ซึ่งระบบช่วยประมาณการงานในการพัฒนาซอฟต์แวร์นี้จะช่วยให้การประมาณการงานและการบริหารโครงการมีประสิทธิภาพเพิ่มมากขึ้น สร้างมาตรฐานการประมาณการงานของบริษัท ตลอดจนช่วยลดปัญหาการส่งมอบงานล่าช้าเกินกำหนดได้

5.2 ผลการดำเนินการพัฒนาระบบ

จากการศึกษาและพัฒนาระบบช่วยประมาณการงานในการพัฒนาซอฟต์แวร์ สามารถสรุปผลของการดำเนินโครงการ ได้ดังนี้

1. ผู้บริหารโครงการสามารถใช้งานระบบเพื่อช่วยในการประเมินภาระงานของโครงการ และสามารถใช้งานบันทึกข้อมูลต่างๆ ที่ใช้ประกอบการบริหารโครงการได้
2. การประเมินภาระงานของบริษัทเริ่มเป็นไปในทิศทางเดียวกันมากยิ่งขึ้น
3. ผู้บริหารโครงการสามารถนำรายงานที่สร้างจากระบบใช้เป็นข้อมูลในการปรับปรุงค่าพารามิเตอร์ของแบบจำลอง เพื่อให้การประเมินภาระงานใกล้เคียงกับการทำงานจริงมากยิ่งขึ้นได้

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของวิศวกรรมศาสตราจารย์ ดร.สุวิทย์ วัฒนศิริกุล การนำเอกสารนี้ไปใช้โดยไม่ผ่านการอนุญาตถือว่าผิดกฎหมาย

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อจำกัดและข้อเสนอแนะ

จากการศึกษาและพัฒนาระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ พบว่ายังคงมีข้อจำกัด แต่สามารถเพิ่มคุณสมบัติบางประการ เพื่อลดข้อจำกัดและพัฒนาให้ระบบมีประสิทธิภาพเพิ่มสูงขึ้นได้ โดยสามารถสรุปประเด็นต่างๆ ได้ดังนี้

1. ระบบช่วยประมาณภาระงานในการพัฒนาซอฟต์แวร์ถูกพัฒนาขึ้นด้วยการประยุกต์ใช้งานแบบจำลอง COCOMO II ซึ่งเป็นการประเมินภาระงานโดยใช้แบบจำลองทางคณิตศาสตร์ที่มีพื้นฐานมาจากการเก็บสถิติการดำเนินโครงการพัฒนาซอฟต์แวร์ในต่างประเทศ ซึ่งวัฒนธรรม และรูปแบบการดำเนินชีวิตที่แตกต่างจากคนไทย อาจทำให้ผลการประเมินภาระงานของระบบในช่วงแรกนั้นยังไม่ถูกต้องเท่าไรนัก ดังนั้นผู้บริหาร โครงการควรจะปรับปรุงพารามิเตอร์ของแบบจำลองในระบบให้มีความเหมาะสมมากยิ่งขึ้น จึงจะทำให้การประเมินภาระงานของระบบมีประสิทธิภาพสูงขึ้น
2. การประเมินภาระงานโดยใช้แบบจำลองทางคณิตศาสตร์นี้ ต้องใช้ข้อมูลของโครงการที่ประเมินภาระงานจากการบินที่กของผู้บริหาร โครงการ ดังนั้นหากผู้บริหาร โครงการบันทึกข้อมูลที่ไม่สมเหตุผล การประเมินภาระงานอาจจะไม่สามารถเชื่อถือได้
3. เชื่อมต่อการทำงานเข้ากับระบบงานอื่นของบริษัท เช่น ฐานข้อมูลพนักงาน (LDAP) ระบบช่วยจัดการทรัพยากรบุคคล เพื่อเชื่อมโยงการทำงาน และลดความซ้ำซ้อนของการจัดเก็บข้อมูล
4. ในอนาคตหากมีข้อมูลการประเมินภาระงานของโครงการมากเพียงพอ ควรจะพัฒนาระบบให้มีการประเมินภาระงานในรูปแบบของการเปรียบเทียบข้อมูลการประเมินโครงการเก่า (Analogy Model) เพิ่มอีกวิธีหนึ่ง เพื่อทำให้การประเมินภาระงานมีความน่าเชื่อถือมากยิ่งขึ้น

บรรณานุกรม

วีระศักดิ์ ชิงฉาวร. 2548. **Java Programming Volume III**. กรุงเทพฯ. ซีเอ็ดยูเคชั่น.

Boehm, Barry and others. 1997. **COCOMO II Model Definition Manual**. [Online]. Available:

<http://sunset.usc.edu/research/COCOMOII/Docs/modelman.pdf>

CCSE Website. 1997. **COCOMO II**. [Online]. Available:

http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html

Galorath, Dan. 2008. **Estimating Software Maintenance**. [Online]. Available:

<http://www.galorath.com/wp/estimating-software-maintenance.php>

Kirk, Jim. 2000. **Project Management for Software Development: Schedule Estimation**.

[Online]. Available: http://www.qsm.com/soft-est_overview.htm

Karaka, Mit and Sultanolu, Sencer. 1998. **Object Points**. [Online]. Available:

<http://yunus.hacettepe.edu.tr/~sencer/objectp.html>

Object Management Group. 2007. **OMG Unified Modeling Language (OMG UML),**

Infrastructure, V2.1.2. [Online]. Available: <http://www.omg.org/docs/formal/07-11-04.pdf>

Red Hat Middleware. 2008. **Java Persistence with Hibernate**. [Online]. Available:

<http://hibernate.org/397.html>

McConnell, Steve. 2006. **Software Estimation: Demystifying the Black Art**. Washington:

Microsoft Press A Division of Microsoft Corporation.

ประวัติผู้เขียน

ชื่อผู้จัดทำโครงการ	นายธิดิษฐ์ วงศ์ปิยะ
วันเดือนปีเกิด	7 ตุลาคม 2523
สถานที่เกิด	กรุงเทพมหานคร
ประวัติการศึกษา	
มัธยมศึกษา	โรงเรียนราชวินิตบางแก้ว ในพระบรมราชูปถัมภ์
อุดมศึกษา	ทล.บ. เทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ
ประวัติการทำงาน	
พ.ศ. 2546 – ปัจจุบัน	ตำแหน่ง System Analyst บริษัท โมทีฟ เทคโนโลยี จำกัด (มหาชน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้