การพัฒนามาตรวัดซอฟต์แวร์สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล

นายจักรนรินทร์ คงเจริญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2548

ISBN 974-533-559-2

DEVELOPMENT OF A MEASUREMENT TOOL FOR PHP LANGUAGE WITH MYSQL DATABASE

Chaknarin Kongcharoen

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2005

ISBN 974-533-559-2

การพัฒนามาตรวัดซอฟต์แวร์สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(ผศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ

(ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(รศ. ดร.นิตยา เกิดประสพ)
กรรมการ

(อ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

(รศ. ดร.เสาวณีย์ รัตนพานี)
รองอธิการบดีฝ่ายวิชาการ

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

จักรนรินทร์ คงเจริญ : การพัฒนามาตรวัดซอฟต์แวร์สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล (DEVELOPMENT OF A MEASUREMENT TOOL FOR PHP LANGUAGE WITH MYSQL DATABASE) อาจารย์ที่ปรึกษา : ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์, 114 หน้า. ISBN 974-533-559-2

การวางแผนและวิเคราะห์ความต้องการ เป็นขั้นตอนแรกของการพัฒนาโครงการทางวิศวกรรมซอฟต์แวร์ ดังนั้นจึงจำเป็นต้องทำประมาณขนาดของโครงการเพื่อเป็นข้อมูลในการจัดทำตารางเวลาและจัดสรรจำนวนนักพัฒนา โดยการประมาณขนาดของโครงการหรือการประมาณขนาดของซอฟต์แวร์นั้นสามารถทำด้วยการใช้มาตรวัดซอฟต์แวร์ และมาตรวัดซอฟต์แวร์ที่มีการใช้งานและเป็นที่ยอมรับของนักพัฒนาคือมาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ เป็นมาตรวัดซอฟต์แวร์ที่ทำการวัดขนาดของซอฟต์แวร์ออกมาเป็นความสามารถทางฟังก์ชัน

งานวิจัยครั้งนี้จะเป็นการพัฒนามาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ สำหรับโปรแกรมที่พัฒนาบนเว็บด้วยโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล โดยการเพิ่มส่วนประกอบเว็บที่เป็นมัลติมีเดียไฟล์เข้าไปในการนับของฟังก์ชันพอยต์

จากผลการวิจัยพบว่าเมื่อเพิ่มส่วนประกอบเว็บที่เป็นมัลติมีเดียไฟล์เข้าไปในการนับฟังก์ชันพอยต์จะทำให้การใช้มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ จะมีค่าประสิทธิภาพทางสถิติที่ดีกว่าใช้มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ปกติ

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2548 ลายมือชื่ออาจารย์ที่ปรึกษา_____

CHAKNARIN KONGCHAROEN : DEVELOPMENT OF A MEASUREMENT TOOL FOR PHP LANGUAGE WITH MYSQL DATABASE. THESIS ADVISOR : ASST. PROF. PICHAYOTAI MAHATTHANAPIWAT, Ph.D., 114 PP. ISBN 974-533-559-2

FUNCTION POINT COUNTING/MULTIMEDIA FILE/MEASUREMENT TOOL

Planning and requirement analysis is the first phase for project development in software engineering. Therefore, it is necessary that the size of project be considered to estimate the schedule and effort used. A measurement tool will be used to estimate the size of the software for the project. One of the widely used measurement technique called a function point which can measure the size of the software by its functional capability.

This research developed a measurement tool for PHP language with MySQL database using function point technique. Multimedia files that are web components will be included as part of the function points.

The result of the rearch showed that the inclusion of the multimedia files to the function point counting improved the performance statistics when compared with the traditional function point.

School of Computer Engineering

Academic Year 2005

Student's Signature _____

Advisor's Signature _____

กิตติกรรมประกาศ

ความสำเร็จจากวิทยานิพนธ์นี้ เริ่มด้วยข้าพเจ้าได้รับความอนุเคราะห์ให้เข้าศึกษาต่อ ณ มหาวิทยาลัยเทคโนโลยีสุรนารี ข้าพเจ้าใคร่ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. พิชโยทัย มหัทธนาภิวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ ที่ได้ให้คำแนะนำ ให้ความรู้ และคำชี้แนะแนวทางอันเป็นประโยชน์ต่อการทำวิทยานิพนธ์ อีกทั้งขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. คะชา ชาญศิลป์ ที่ถ่ายทอดความรู้ประกอบการทำวิทยานิพนธ์นี้จนเสร็จสมบูรณ์

ขอบพระคุณมหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร ที่ให้โอกาสข้าพเจ้าได้ศึกษาต่อและสนับสนุนทุนการศึกษาตลอดหลักสูตรนี้

ขอขอบคุณเพื่อนร่วมชั้นเรียนและน้องร่วมสาขาทุกคน ที่มีส่วนร่วมในการส่งเสริมกิจกรรมระหว่างการศึกษาของข้าพเจ้าให้สนุกสนาน ให้กำลังใจที่ดีระหว่างการศึกษาและให้คำแนะนำที่ข้าพเจ้ามองไม่เห็นระหว่างทำวิทยานิพนธ์เล่มนี้

ท้ายสุดนี้ ข้าพเจ้าจะลืมไม่ได้ ขอขอบพระคุณอย่างยิ่งต่อคุณบิดามารดา พี่สาว และน้องชายที่ส่งเสริมและช่วยเหลือข้าพเจ้าระหว่างศึกษา ประสบการณ์จากการศึกษาในสถาบันแห่งนี้เป็นประสบการณ์ที่ดีในช่วงเวลาหนึ่งที่ข้าพเจ้าจะจดจำตลอดไป การศึกษาเป็นสิ่งที่ข้าพเจ้าเห็นว่ามีคุณค่ามาก เราเรียนรู้ เราได้ความรู้ ความรู้คือสมบัติติดตัวที่ไม่มีใครเอาไปได้ ประสบการณ์และความรู้ที่ได้รับ ข้าพเจ้าจะนำไปใช้ให้เกิดประโยชน์แก่ส่วนรวมและตัวข้าพเจ้าให้มากที่สุด

นายจักรนรินทร์ คงเจริญ

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ช
สารบัญรูป ..... ญ

บทที่

1 บทนำ 1
1.1 ความเป็นมาและที่มาของปัญหาวิจัย 1
1.2 วัตถุประสงค์การวิจัย 4
1.3 ขอบเขตของการวิจัย 4
1.4 ประโยชน์ที่คาดว่าจะได้รับ 4
2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 5
2.1 มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ 5
2.1.1 ขั้นตอนการนับฟังก์ชันพอยต์ (Function Point Counting Procedure) 5
2.1.2 ส่วนประกอบฟังก์ชันพื้นฐาน (Base Functional Components [BFC]) 8
2.1.3 แฟคเตอร์ที่ปรับแต่งค่า (Value Adjustment Factor[VAF]) 19
2.2 การวิเคราะห์ฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ 22
2.3 การประมาณขนาดของซอฟต์แวร์ด้วยแบบจำลองโคโคโม2 24
2.3.1 แบบจำลองแอพพลิเคชันคอมโพซิชัน 24
2.3.2 แบบจำลองเออรีดิไซด์ 26
2.3.3 แบบจำลองโพสอาร์คิเท็กเจอร์ 29
3 วิธีดำเนินการวิจัย 36
3.1 ระเบียบวิธีวิจัย 36

สารบัญ (ต่อ)

หน้า

3.2 มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ 36
3.2.1 การนับค่าฟังก์ชันพอยต์ร่วมกับส่วนประกอบของเว็บ 38
3.3.2 การวิเคราะห์ความถดถอยและสหสัมพันธ์อย่างง่าย (Simple Regression Analysis and Correlation) 40
3.3 การเก็บรวบรวมและการวิเคราะห์โปรแกรมต้นแบบ 48
3.3.1 การเก็บรวบรวมโปรแกรมเว็บและนับฟังก์ชันพอยต์ 48
3.3.2 การสร้างสมการประมาณขนาดด้วยวิธีการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย 53
3.4 โปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ 55
3.4.1 ฐานข้อมูล 55
3.4.2 รูปแบบและการทำงานของโปแกรม 56
3.5 โปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 57
4 ผลการวิเคราะห์ข้อมูลและอภิปรายผล 59
4.1 การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บด้วยวิธี Cross-Validation 59
4.2 ประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บกับมาตรวัดแบบฟังก์ชันพอยต์ 60
4.3 การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บด้วยการทดสอบด้วยโปรแกรมที่พัฒนาขึ้นจริง 62
5 สรุปผลการวิจัยและข้อเสนอแนะ 66
5.1 สรุปผลการวิจัย 66
5.2 การประยุกต์ผลการวิจัย 67
5.3 ข้อเสนอแนะในการวิจัยต่อไป 67
รายการอ้างอิง 68

สารบัญ (ต่อ)

หน้า

ภาคผนวก

ภาคผนวก ก บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการ KMITL International Conference on Science and Applied Science 2006 70

ภาคผนวก ข การกำหนดค่าตัวคูณความพยายาม (Effort Multipliers) สำหรับแบบจำลองโพสอาร์คิเท็กเจอร์ (Post Architecture Model) 75

ภาคผนวก ค คู่มือการใช้งานโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 94

ภาคผนวก ง เอกสารความต้องการของระบบแจ้งการเดินการออนไลน์ 105

ภาคผนวก จ ตารางฟังก์ชันพอยต์ของระบบแจ้งการเดินการออนไลน์ 112

ประวัติผู้เขียน 114

สารบัญตาราง

ตารางที่ **หน้า**

1.1 การคำนวณมาตรวัดแบบฟังชันพอยต์ 2
2.1 การหาขนาดของซอฟต์แวร์ในขั้นตอนต่างๆในวงจรการพัฒนา 6
2.2 ตารางการนับค่าฟังก์ชันพอยต์ 9
2.3 ขบวนการตรรกสำหรับส่วนประกอบทรานเซคชัน 11
2.4 ระดับความซับซ้อนของอินพุทภายนอก 14
2.5 ระดับความซับซ้อนของเอาท์พุทภายนอก 16
2.6 ระดับความซับซ้อนของส่วนการสอบถามภายนอก 17
2.7 ระดับความซับซ้อนของแฟ้มข้อมูลภายใน 18
2.8 ระดับความซับซ้อนของอินเตอร์เฟสแฟ้มข้อมูล 19
2.9 ตารางการปรับค่าตามความซับซ้อนในการประมวลผล 21
2.10 การคำนวณวัตถุเว็บ (Web Object Calulation) 22
2.11 การคำนวณวัตถุเว็บ (Web Object Calulation) 23
2.12 การแบ่งกลุ่มของอ็อบเจ็กต์พอยต์ 25
2.13 การกำหนดน้ำหนักของอ็อบเจ็กต์พอยต์ 25
2.14 ค่าอัตราการผลิต 26
2.15 สัญลักษณ์และคำอธิบายของสมการจากเออรีดิไซด์ 29
2.16 สัญลักษณ์และคำอธิบายของสมการจากโพสอาร์คิเท็กเจอร์ 34
2.17 ค่าตัวคูณความพยายามในแต่ละระดับ 35
3.1 การนับค่าฟังก์ชันพอยต์ร่วมกับส่วนประกอบของเว็บ 38
3.2 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมตู้เพลง 48
3.3 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านหนังสือออนไลน์ 1 49
3.4 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมจัดการภาพ 49
3.5 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 1 50
3.6 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 2 50

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

3.7 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 3 51
3.8 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านหนังสือออนไลน์ 2 51
3.9 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมระบบจองห้องพัก..... 52
3.10 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมระบบตำแหน่งงานออนไลน์..... 52
3.11 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 4 53
3.12 ตารางการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย 54
4.1 ตารางความคลาดเคลื่อนสัมบูรณ์และความคลาดเคลื่อนกำลังสอง 59
4.2 ตารางความคลาดเคลื่อนสัมบูรณ์และความคลาดเคลื่อนกำลังสอง 60
4.3 ตารางเปรียบเทียบค่าประสิทธิภาพทางสถิติ 61
4.4 ขั้นตอนพัฒนาโปรแกรมแบบน้ำตก (Waterfall Phase) 63
4.5 ค่าตัวคูณความพยายามสำหรับโปรแกรมแจ้งการเดินทาง 65
ข.1 ความแตกต่างของแต่ละระดับของ RELY 76
ข.2 ความแตกต่างของแต่ละระดับของ DATA..... 77
ข.3 ความแตกต่างของแต่ละระดับของ Control Operation..... 78
ข.4 ความแตกต่างของแต่ละระดับของ Computation Operation..... 79
ข.5 ความแตกต่างของแต่ละระดับของ Device-dependent Operation..... 80
ข.6 ความแตกต่างของแต่ละระดับของ Data Management Operation 81
ข.7 ความแตกต่างของแต่ละระดับของ Data Management Operation 82
ข.8 ความแตกต่างของแต่ละระดับของ DOCU 83
ข.9 ความแตกต่างของแต่ละระดับของ RUSE..... 84
ข.10 ความแตกต่างของแต่ละระดับของ TIME 85
ข.11 ความแตกต่างของแต่ละระดับของ STOR..... 85
ข.12 ความแตกต่างของแต่ละระดับของ PVOL 86
ข.13 ความแตกต่างของแต่ละระดับของ ACAP 87
ข.14 ความแตกต่างของแต่ละระดับของ PCAP..... 88
ข.15 ความแตกต่างของแต่ละระดับของ APEX 88

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

ข.16 ความแตกต่างของแต่ละระดับของ PLEX 89
ข.17 ความแตกต่างของแต่ละระดับของ LTEX 89
ข.18 ความแตกต่างของแต่ละระดับของ PCON 90
ข.19 ความแตกต่างของแต่ละระดับของ TOOL 91
ข.20 ความแตกต่างของแต่ละระดับของ SITE เกี่ยวกับสถานที่ 92
ข.21 ความแตกต่างของแต่ละระดับของ SITE เกี่ยวกับอุปกรณ์การสื่อสาร 93
ข.22 ความแตกต่างของแต่ละระดับของ SCED 93
จ.1 ตารางจำนวนฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมแจ้งการเดินทาง 112

สารบัญรูป

รูปที่ **หน้า**

2.1 ขั้นตอนการนับฟังก์ชันพอยต์ 7
2.2 ส่วนประกอบของฟังก์ชันพอยต์ 9
3.1 ตัวอย่างแผนภาพการกระจายซึ่งแสดงความสัมพันธ์ระหว่าง X และ Y 42
3.2 ความสัมพันธ์ระหว่างราคาขายบ้าน (Y) กับขนาดของพื้นที่บ้าน (X) ก และ ข 42
3.3 สมการเชิงเส้น 43
3.4 การแสดงค่าของ r ที่มีค่า -1 < r < 1 47
3.5 โปรแกรม Count Code Pro 55
3.6 โปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ 56
3.7 ฐานข้อมูลโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ 57
3.8 โปรแกรมโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 58
3.9 ขั้นตอนการพัฒนาโปรแกรมแบบน้ำตก (waterfall) 58
4.1 การนับฟังก์ชันพอยต์จากโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ 63
4.2 ผลการคำนวณของโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 64
ค.1 หน้าจอภาพโปรแกรมยูเอสซี โคโคโม2 95
ค.2 กล่องรับขนาดของโมดูลที่เป็นจำนวนบรรทัดคำสั่ง 96
ค.3 กล่องรับขนาดของโมดูลที่เป็นฟังก์ชันพอยต์ 97
ค.4 กล่องรับขนาดของโมดูลที่เป็นจำนวนบรรทัดคำสั่งที่ปรับปรุงและการนำกลับมาใช้ 97
ค.5 กล่องรับค่าตัวคูณความพยายามสำหรับแบบจำลองเออรีดิไซด์ 98
ค.6 กล่องรับค่าตัวคูณความพยายามสำหรับแบบจำลองโพสอาร์คิเท็กเจอร์ 98
ค.7 กล่องรับข้อมูลของตารางเวลา 99
ค.8 กล่องค่าสเกลของปัจจัย 100
ค.9 กล่องแสดงความเสี่ยง 100
ค.10 การสร้างโมดูลและการใส่ค่าจำนวนบรรทัดคำสังและค่าใช้จ่าย 101
ค.11 พื้นที่ผลลัพธ์การคำนวณ 101

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ค.12 เมนูแสดงขั้นตอนพัฒนาโปรแกรมแบบน้ำตก 102
ค.13 แสดงขั้นตอนพัฒนาโปรแกรมแบบน้ำตกในทุกช่วง 102
ค.14 แสดงขั้นตอนพัฒนาในช่วงการวางแผนและการกำหนดความต้องการ 103
ค.15 แสดงขั้นตอนพัฒนาในช่วงการออกแบบผลิตภัณฑ์ 103
ค.16 แสดงขั้นตอนพัฒนาในช่วงการเขียนโปรแกรม 104
ค.17 แสดงขั้นตอนพัฒนาในช่วงการรวมโปรแกรมและการทดสอบ 104
ง.1 แผนผังโครงสร้างระบบแจ้งการเดินทางออนไลน์ (State Diagram) 107
ง.2 แผนผังฐานข้อมูลโปรแกรมแจ้งการเดินทาง 110

บทที่ 1
บทนำ

1.1 ความเป็นมาและที่มาของปัญหาวิจัย

การดำเนินโครงการทางวิศวกรรมซอฟต์แวร์ จะมีการแบ่งขั้นตอนการทำงานออกเป็นขั้นตอนหลักได้แก่ การวางแผนและการวิเคราะห์ความต้องการ การออกแบบ การเขียนโปรแกรม การรวมโปรแกรมและการทดสอบ และ การบำรุงรักษา โดยการดำเนินโครงการมีความสำคัญทุกขั้นตอน แต่การวางแผนและการวิเคราะห์ความต้องการเป็นขั้นตอนแรกที่จะต้องดำเนินการ และจะมีผลต่อการดำเนินโครงการในขั้นต่อไป ทำให้ในขั้นตอนนี้จะต้องวางแผนโครงการที่ได้จากการวิเคราะห์ความต้องการของซอฟต์แวร์ (Software Requirement Specification) ให้ครบถ้วน ดังนั้นการวางแผนและวิเคราะห์ความต้องการของซอฟต์แวร์จึงเป็นขั้นตอนที่จะทำให้การดำเนินโครงการเป็นไปอย่างราบรื่นและเสร็จทันกำหนดส่งงาน

ดังนั้นการวางแผนที่ดีต้องการนำเครื่องมือวัด หรือ มาตรวัดซอฟต์แวร์ (Software Measurement Tool) มาช่วยในการประมาณขนาดของซอฟต์แวร์ คำนวณความพยายาม (Effort) และระยะเวลาพัฒนา (Duration) เพื่อเป็นข้อมูลในการวางแผนการดำเนินโครงการ

ในอดีตมาตรวัดซอฟต์แวร์ใช้จำนวนบรรทัดคำสั่ง (Source Line of Code) เป็นขนาดของซอฟต์แวร์ โดยจะทำการจดบันทึกค่าจำนวนบรรทัดคำสั่งต่อเวลาต่อคนเก็บไว้แล้วทำการคำนวณค่าเฉลี่ยเป็นค่ามาตรฐาน เช่น 10.0 พันบรรทัดต่อนักพัฒนาต่อเดือน (10.0 KSLOC/Person months) แต่มาตรวัดแบบนี้ไม่เป็นที่ยอมรับโดยทั่วไป เนื่องจากมีการใช้จำนวนบรรทัดคำสั่งเป็นหลักในการวัด ซึ่งจะขึ้นอยู่กับโปรแกรมภาษาที่ใช้ในการพัฒนาและความชำนาญในการเขียนโปรแกรมของนักพัฒนาที่เขียนงานเดียวกันอาจเขียนเป็นจำนวนบรรทัดคำสั่งแตกต่างกัน ทำให้การประมาณความพยายามและระยะเวลาพัฒนาขึ้นอยู่กับความชำนาญของนักพัฒนาในแต่ละหน่วยงานซึ่งเป็นข้อมูลที่วัดออกมาเป็นจำนวนได้ยาก

ต่อมาในประมาณปี ค.ศ. 1970 Allan J. Albrecht จากบริษัทไอบีเอ็มได้คิดค้นมาตรวัดเชิงฟังก์ชันขึ้นมาเป็นการวัดขนาดซอฟต์แวร์ทางอ้อมโดยใช้ลักษณะทางฟังก์ชันของซอฟต์แวร์ ที่เรียกว่าฟังก์ชันพอยต์ (Function Point[FP]) และในปี ค.ศ. 1984 บริษัทไอบีเอ็มโดย Allan Albrecht ได้ออกคู่มือมาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ชื่อ CIS & A Guideline 313, AD/M

Productivity Measurement and Estimate Validation (International Function Point Users Group [IFPUG], 2005) โดยฟังก์ชันพอยต์นั้นได้ทำการแยกส่วนประกอบ (Component) ของซอฟต์แวร์ออกเป็น 5 ส่วนประกอบและมีการประเมินความซับซ้อนของซอฟต์แวร์ออกเป็น 3 ระดับ โดยการคำนวณแสดงตามตารางที่ 1.1 จะมีการพิจารณาคุณลักษณะของส่วนประกอบของซอฟต์แวร์ เพื่อใช้เป็นพารามิเตอร์ (Parameter)ในการนับ การนับจำนวนพารามิเตอร์เหล่านี้ และบันทึกลงในตาราง โดยส่วนประกอบที่ใช้เป็นพารามิเตอร์ในการนับประกอบด้วย แฟ้มข้อมูลภายใน อินเตอร์เฟสแฟ้มข้อมูล อินพุทภายนอก เอาท์พุทภายนอก และการสอบถามภายนอก เมื่อมีการพิจารณาพารามิเตอร์ต่างๆ ของซอฟต์แวร์ดังกล่าวข้างต้นแล้ว ค่าที่นับได้จะมีความสัมพันธ์กับความซับซ้อน ซึ่งในโครงสร้างของฟังก์ชันพอยต์ได้กำหนดเกณฑ์การเลือกระดับความซับซ้อนว่าจะเป็นระดับต่ำ ระดับปานกลาง หรือระดับสูง และทำการนับรวมค่าฟังก์ชันพอยต์จากแต่ละพารามิเตอร์เป็นค่าฟังก์ชันพอยต์ของซอฟต์แวร์นั้น

ตารางที่ 1.1 การคำนวณมาตรวัดแบบฟังชันพอยต์

ส่วนประกอบ	นับได้	ความซับซ้อน			รวม FP
		ต่ำ	ปานกลาง	สูง	
แฟ้มข้อมูลภายใน		x7	x10	x15	
อินเตอร์เฟสแฟ้มข้อมูล		x5	x7	x10	
อินพุทภายนอก		x3	x4	x6	
เอาท์พุทภายนอก		x4	x5	x7	
การสอบถามภายนอก		x3	x4	x6	
จำนวน FP ทั้งหมด					

จากนั้นมาตรวัดแบบฟังก์ชันพอยต์ก็ได้รับการพัฒนาปรับปรุงใช้งานกันอย่างแพร่หลาย และนำมาใช้กับซอฟต์แวร์ที่พัฒนาจากโปรแกรมภาษา (Programming Language) ในภาษาต่างๆ มากขึ้น จนในปัจจุบันได้มีการจัดตั้งเป็นองค์กรชื่อ International Function Point Users Group หรือ IFPUG ที่ทำการศึกษา วิจัย อบรม และเผยแพร่ความรู้ของมาตรวัดแบบฟังก์ชันพอยต์ (Garmus and Herron, 2001) โดยมาตรวัดแบบฟังก์ชันพอยต์สามารถใช้กับโปรแกรมภาษาในยุคใหม่เช่น C++, LISP, Visual Basic, Oracle, PERL, HTML, XML และ MATHCAD (Reifer, 2002) จะเห็นได้ว่ามาตรวัดแบบฟังก์ชันพอยต์สามารถปรับใช้กับโปรแกรมภาษาทั้งใหม่และเก่าได้

งานวิจัยนี้จะนำมาตรวัดแบบฟังก์พอยต์นำมาปรับใช้กับ โปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล (PHP Language with Mysql Database) ซึ่งเป็นโปรแกรมภาษาที่ทำงานบนเว็บ หรือ โปรแกรมเว็บ (Web Base Application) เป็นโปรแกรมภาษาแบบเปิดเผยซอร์สโค้ด (Open Sources) และ โปรแกรมภาษาดังกล่าวมีการใช้งานกันอย่างแพร่หลายสามารถพัฒนาโปรแกรมที่ใช้งานได้หลายประเภท ทั้งโปรแกรมขนาดเล็กและใหญ่ มีความน่าเชื่อถือ มีองค์กรที่ดูแล มีการวิจัยและพัฒนาอย่างต่อเนื่องทั้งเป็นโปรแกรมภาษาแบบเปิดเผยซอร์สโค้ดจึงทำให้ไม่มีค่าลิขสิทธิ์ในการใช้โปรแกรมภาษานี้ในการพัฒนา

มาตรวัดแบบฟังก์ชันพอยต์สำหรับโปรแกรมเว็บ มีผู้ศึกษาและวิจัยอย่างต่อเนื่องโดยส่วนใหญ่จะทำการนับจำนวนส่วนประกอบเว็บ (Web Component) เป็นหลัก (Reifer, 2000) เพราะในอดีตยังคงเป็นโปรแกรมเว็บแบบสถิต (Static Web) แต่เมื่อมีการมีการนำระบบฐานข้อมูลเข้ามาใช้กับโปรแกรมเว็บ จึงทำให้โครงสร้างของโปรแกรมภาษามีความเหมือนกับโครงสร้างภาษาของโปรแกรมภาษาอื่นๆ ทำให้มีการนำเอามาตรวัดแบบฟังก์ชันพอยต์เข้ามาใช้กับโปรแกรมเว็บ (Garmus and Herron, 2001, pp.200-204) แต่ในงานวิจัยนี้จะทำงานเพิ่มส่วนประกอบเว็บที่เป็นมัลติมีเดียไฟล์ (Multi-Media Files) เข้ามาเป็นส่วนประกอบหนึ่งของมาตรวัด มัลติมีเดียไฟล์เป็นส่วนประกอบที่มีผลเพิ่มขนาดของซอฟต์แวร์ และสามารถวิเคราะห์ออกมาได้จากความต้องการของซอฟต์แวร์ตั้งแต่แรกในการพัฒนาโปรแกรม จากนั้นนำจำนวนบรรทัดคำสั่งและจำนวนฟังก์ชันพอยต์มาสร้างสมการประมาณขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่ง ด้วยวิธีการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย

ส่วนจำนวนความพยายาม (Effort) และระยะเวลาพัฒนา (Duration)นั้น จะได้จากขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่งนำเข้าไปคำนวณโดยการใช้แบบจำลองโคโคโม2 (COCOMO II Model) ซึ่งเป็นแบบจำลองที่มีการพัฒนาจนเป็นที่ยอมรับและใช้งานกันอย่างแพร่หลาย (Boehm, et al, 2000) และโปรแกรมภาษาพีเอชพีนั้นมีลักษณะเป็นโปรแกรมภาษาแบบโครงสร้างเป็นหลักจึงทำให้เหมาะกับแบบจำลองโคโคโม2 อีกทั้งนำจำนวนบรรทัดคำสั่งที่จะนำไปใช้ในโคโคโม2 นั้นจะไม่มีการปรับค่าด้วยค่าแฟคเตอร์ที่ปรับแต่งค่า (Value Adjustment Factor[VAF]) จะช่วยลดขั้นตอนในการประมาณขนาดของซอฟต์แวร์อีกด้วย

งานวิจัยครั้งนี้จะเป็นการสร้างเครื่องมือเพื่อช่วยวัดขนาดของ โปรแกรมภาษาพีเอสพีร่วมกับระบบฐานข้อมูลมายเอสคิวแอล โดยมีพื้นฐานจากฟังก์ชันพอยต์และมีการเพิ่มเติมส่วนประกอบของเว็บเข้าไปเพื่อการวัดจะได้ข้อมูลที่ครบถ้วนมากขึ้น และเมื่อทราบขนาดของซอฟต์แวร์แล้วนั้นจะได้นำขนาดของซอฟต์แวร์นั้นนำไปคำนวณหาจำนวนความพยายามและระยะเวลาพัฒนาจากแบบจำลองโคโคโม2

1.2 วัตถุประสงค์การวิจัย

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อการออกแบบและพัฒนามาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บสำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล

1.3 ขอบเขตของการวิจัย

วิทยานิพนธ์มีขอบเขตของงานวิจัยดังนี้

1.3.1 พัฒนามาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล

1.3.2 สร้างสมการประมาณของของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่ง

1.3.3 คำนวณหาจำนวนความพยายามและระยะเวลาพัฒนาจากแบบจำลองของโคโคโม2

1.3.4 โปรแกรมต้นแบบเป็นโปรแกรมที่พัฒนาบนเว็บโดยโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล ในประเภทเว็บพาณิชย์อิเลกทรอนิคส์ (Web-based electronic commerce) และเว็บข้อมูลสารสนเทศ (Web-based information utilities)

1.3.5 ใช้โปรแกรมต้นแบบจากแหล่งที่เปิดเผยซอร์สโค้ด เป็นข้อมูลฝึก (Train Data) และเป็นข้อมูลทดสอบ (Test Data)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากวิทยานิพนธ์และเครื่องมือวัดซอฟต์แวร์มีดังนี้

1.4.1 สามารถใช้มาตรวัดซอฟต์แวร์ที่พัฒนาขึ้นเป็นมาตรวัดซอฟต์แวร์สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล เมื่อนักพัฒนาโปรแกรมกำหนดความต้องการของซอฟต์แวร์ที่จะพัฒนา จากนั้นสามารถใช้มาตรวัดซอฟต์แวร์นี้คำนวณขนาดโปรแกรม จำนวนความพยายาม และระยะเวลาพัฒนา ของซอฟต์แวร์ ได้ตั้งแต่ในขั้นตอนการวิเคราะห์ความต้องการ ทำให้สามารถวางแผนในการพัฒนาซอฟต์แวร์เป็นแนวทางได้

1.4.2 จำนวนความพยายาม และระยะเวลาพัฒนาขนาดของซอฟต์แวร์ที่ได้จากการคำนวณของมาตรวัดซอฟต์แวร์สามารถเป็นตัววัดความก้าวหน้าของการพัฒนาซอฟต์แวร์ได้ โดยการจัดทำตารางเวลา (Schedule) และตารางเวลานี้จะเป็นมาตรวัดความก้าวหน้าของซอฟต์แวร์ที่กำลังพัฒนาอยู่

1.4.3 เป็นแนวทางในการพัฒนามาตรวัดซอฟต์แวร์สำหรับโปรแกรมเว็บภาษาอื่น

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

การทำวิจัยครั้งนี้ทำการศึกษามาตรวัดซอฟต์แวร์สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล ซึ่งเป็นโปรแกรมเว็บและยังทำงานร่วมกับฐานข้อมูลทำให้จะต้องทำงานศึกษามาตรวัดแบบฟังก์ชันพอยต์ที่ใช้กับโปรแกรมฐานข้อมูลและโปรแกรมเว็บ ซึ่งจะกล่าวในหัวข้อ 2.1 และ 2.2 ตามลำดับ และเมื่อได้ขนาดของโปรแกรมที่เป็นจำนวนบรรทัดคำสั่งแล้วนั้นก็จะนำไปคำนวณในแบบจำลองโคโคโม2 ในหัวข้อ 2.3

2.1 มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์

ในปลายทศวรรษที่ 1970 Allan J. Albrecht ขณะทำงานวิจัยอยู่ที่บริษัท IBM ได้เสนอมาตรวัดซอฟต์แวร์ที่สามารถใช้ได้กับทุกโปรแกรมภาษา และนักพัฒนาซอฟต์แวร์สามารถมองเห็น รับรู้และทำความเข้าใจได้ไม่ยาก สิ่งที่ Allan J. Albrecht ต้องการวัดคือความสามารถต่างๆที่มีของซอฟต์แวร์ (Functional of Software) Allan J. Albrecht สังเกตว่าซอฟต์แวร์ต่างมีองค์ประกอบพื้นฐานที่คล้ายคลึงกัน ซึ่งมีด้วยกัน 5 องค์ประกอบ คือ อินพุทภายนอก (External Inputs), เอาท์พุทภายนอก (External Outputs), การสอบถามภายนอก (External Inquiries), แฟ้มข้อมูลภายใน (Internal Logical Files) และ โปรแกรมอินเตอร์เฟส (External Interface Files) และยังมีแฟคเตอร์ที่ปรับแต่งค่า (Value Adjustment Factor) ที่เป็นคุณลักษณะในแต่ละโปรแกรมอีกที่มีผลต่อขนาดของซอฟต์แวร์นั้นๆ โดยขั้นตอนการนับฟังก์ชันพอยต์ ส่วนประกอบฟังก์ชันพื้นฐาน (Base Functional Components) และแฟคเตอร์ที่ปรับแต่งค่า (Value Adjustment Factor) จะกล่าวในหัวข้อ 2.1.1 2.1.2 และ 2.1.3 ตามลำดับ

2.1.1 ขั้นตอนการนับฟังก์ชันพอยต์ (Function Point Counting Procedure)

การนับฟังก์ชันพอยต์สามารถทำการนับในหลายขั้นตอนการพัฒนาโปรแกรม อันได้แก่การเสนอโครงร่างโครงการ (Proposal) การกำหนดความต้องการซอฟต์แวร์ (Requirements) การออกแบบ (Design) การเขียนโปรแกรม (Construction) การส่งมอบ (Delivery) และ การบำรุงรักษา (Corrective Maintenance) ตามตารางที่ 2.1 (IFPUG, 2005) โดยการนับในขั้นตอนถัดๆไปนั้นยิ่งจะได้ค่าฟังก์ชันพอยต์ที่ถูกต้องมากขึ้น แต่การจะทำการนับในขั้นตอนใดบ้างนั้นจะขึ้นอยู่กับนักพัฒนาเองว่าต้องการที่จะประมาณหรือวัดโครงการในขั้นตอนไหน

ตารางที่ 2.1 การหาขนาดของซอฟต์แวร์ในขั้นตอนต่างๆในวงจรการพัฒนา

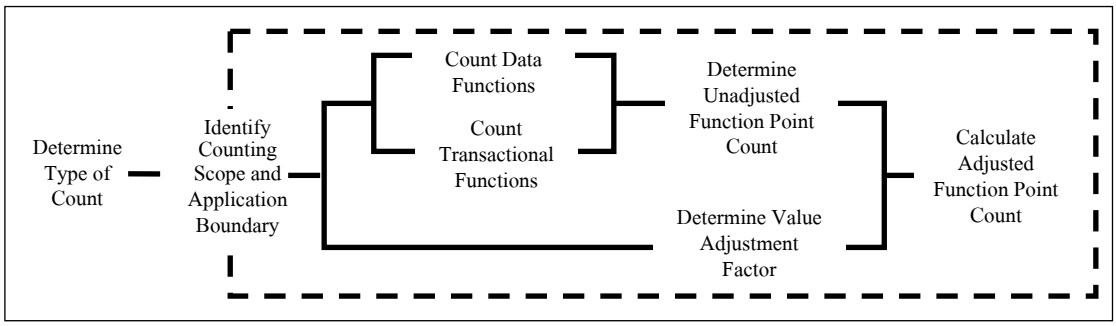
Life Cycle Phase	Size can be approximated	Size can be measured
Proposal: users express needs and intentions	yes	no
Requirements: developers and users review and agree upon expression of user needs and intentions	yes	yes
Design: developers may include elements for implementation that are not used for function point analysis	yes	yes
Construction	yes	yes
Delivery	yes	yes
Corrective Maintenance	yes	yes

โดยการนับฟังก์ชันพอยต์จะมีขั้นตอนการนับตามรูปที่ 2.1 และจะประกอบด้วยขั้นตอนต่อไปนี้

1) กำหนดประเภทของการนับ (Determine the type of function point count) เป็นขั้นตอนแรกในขบวนการนับของฟังก์ชันพอยต์ ประเภทของการนับมีด้วยกัน 3 แบบ คือ
   (1) โครงการที่กำลังพัฒนา (Development Project) เป็นการนับจำนวนฟังก์ชันการทำงานของซอฟต์แวร์ที่ได้รับการติดตั้งใช้งานเป็นครั้งแรก ภายหลังจากการพัฒนาเสร็จสิ้น กล่าวคือเป็นซอฟต์แวร์เวอร์ชันแรกสุดที่ให้กับผู้ใช้ จำนวนฟังก์ชันพอยต์ของโครงการที่กำลังพัฒนา เป็นฟังก์ชันพอยต์เริ่มต้นของการนับแบบแอพพลิเคชัน
   (2) โครงการที่พัฒนาเพิ่ม (Enhancement Project) เป็นการนับจำนวนฟังก์ชันการทำงานของซอฟต์แวร์ที่พัฒนาต่อจากเวอร์ชันก่อนหน้า โดยการเปลี่ยนแปลงของซอฟต์แวร์อาจเป็นการเพิ่มฟังก์ชันการทำงาน การแก้ไขปรับปรุงฟังก์ชันที่มีอยู่เดิม หรือยกเลิกฟังก์ชันที่ไม่จำเป็นหรือผู้ใช้ไม่ต้องการ เป็นต้น จำนวนฟังก์ชันพอยต์ของโครงการที่พัฒนาเพิ่ม จะถูกนำไปปรับปรุงจำนวนฟังก์ชันพอยต์ของแอพพลิเคชันเพื่อสะท้อนให้เห็นถึงการเปลี่ยนแปลงของซอฟต์แวร์

(3) แอพพลิเคชัน (Application) เป็นจำนวนฟังก์ชันพอยต์สุทธิของซอฟต์แวร์ จำนวนฟังก์ชันพอยต์ของโครงการที่กำลังพัฒนา เป็นฟังก์ชันเริ่มต้นของแอพพลิเคชัน และจะถูกปรับปรุงทุกครั้งเมื่อจำนวนฟังก์ชันพอยต์ของโครงการที่พัฒนาเพิ่มได้เกิดขึ้น

โดยการนับฟังก์ชันพอยต์จะเหมือนกันทั้งหมดไม่ว่าจะเป็นประเภทการนับแบบใดก็ตาม แต่จะมีรูปสมการคำนวณฟังก์ชันพอยต์ที่แต่กต่างกันออกไป โดยจำนวนฟังก์ชันหลักๆจะอยู่ในการนับประเภทโครงการที่กำลังพัฒนา



รูปที่ 2.1 ขั้นตอนการนับฟังก์ชันพอยต์ (IFPUG, 2005)

2) กำหนดขอบเขตแอพพลิเคชัน (Identify the counting scope and application boundary) เป็นขั้นตอนการกำหนดขอบเขตแอพพลิเคชันว่ามีคุณสมบัติและความสามารถของซอฟต์แวร์ใดบ้างที่เข้าข่ายในเงื่อนไขการนับ นอกจากนี้ยังต้องกำหนดขอบเขตของแอพพลิเคชันซึ่งแสดงให้เห็นถึงเส้นแบ่งระหว่างซอฟต์แวร์ที่ถูกวัดและผู้ใช้งาน

3) การนับส่วนประกอบข้อมูล (Count the data functions) เป็นการนับจำนวนแฟ้มข้อมูลภายใน (ILF) และอินเตอร์เฟสแฟ้มข้อมูล (EIF) เพิ่มเติมในหัวข้อ 2.1.2

4) การนับส่วนประกอบทรานเซคชัน (Count the transactional functions) เป็นการนับจำนวน อินพุทภายนอก (EI) เอาท์พุทภายนอก (EO) และ การสอบถามภายนอก (EQ) เพิ่มเติมในหัวข้อ 2.1.2

5) การกำหนดปัจจัยความซับซ้อนของส่วนประกอบ (Determine unadjusted function point count) เป็นการกำหนดความซับซ้อบของส่วนประกอบทั้งส่วนประกอบข้อมูล และส่วนประกอบทรานเซคชัน เพิ่มเติมในหัวข้อ 2.1.2

6) การพิจารณาแฟคเตอร์ที่ปรับแต่งค่า (Determine value adjusted factor) เป็นการพิจารณาแฟคเตอร์ที่ปรับแต่งค่าให้กับซอฟต์แวร์ ซึ่งค่าดังกล่าวถูกแบ่งตามลักษณะและประเภทของซอฟต์แวร์ซึ่งมีอยู่ด้วนกัน 14 ประเภท แต่ละประเภทจะมีค่าตั้งแต่ 0 ถึง 5 และค่านี้จะใช้การคำนวณฟังก์ชันพอยต์สุดท้ายต่อไป ดังแสดงในหัวข้อ 2.1.3

7) การคำนวณฟังก์ชันพอยต์ (Calculate the adjusted function point count) เป็นการคำนวณฟังก์ชันพอยต์ซึ่งหลังจากที่มีการพิจรณาจำแนก และ นับส่วนประกอบต่างๆ ของซอฟต์แวร์ และแฟคเตอร์ที่ปรับแต่งค่าดังกล่าวข้างต้นแล้ว ขนาดของซอฟต์แวร์ที่แท้จริงในรูปของหน่วยฟังก์ชัน คำนวณได้จากสูตรสำหรับการนับประเภทโครงการที่กำลังพัฒนา (Development Project) ดังนี้ คือ

$$FP = (UFP + CFP) \times [0.65 + 0.01 \times SUM(C_i)] \quad (1\text{-}1)$$

โดยที่

FP เป็นค่าฟังก์ชันพอยต์

UFP เป็นผลรวมของจำนวน FP ที่นับได้ในส่วนประกอบของซอฟต์แวร์ที่ใช้ในการวัด

CFP เป็นจำนวน FP ที่มาจากการปรับเปลี่ยนข้อมูลเพื่อติดโปรแกรม

C_i (i=1 ถึง 14) เป็นค่าระดับความสำคัญของแฟคเตอร์ที่ปรับแต่งค่าทั้ง 14 ค่า ซึ่งขึ้นอยู่กับการพิจารณา เลือกระดับความสำคัญของแต่ละแฟคเตอร์ที่ปรับแต่งค่า

2.1.2 ส่วนประกอบฟังก์ชันพื้นฐาน (Base Functional Components [BFC])

เป็นการพิจารณาจำนวนส่วนประกอบของซอฟต์แวร์ที่มองเห็นได้ ให้เป็นค่าหน่วยฟังก์ชันพอยต์ ซึ่งส่วนประกอบของซอฟต์แวร์ตามรูปที่ 2.2 จำแนกออกเป็น 5 ส่วนประกอบคือ

1) อินพุทภายนอก (External Inputs)

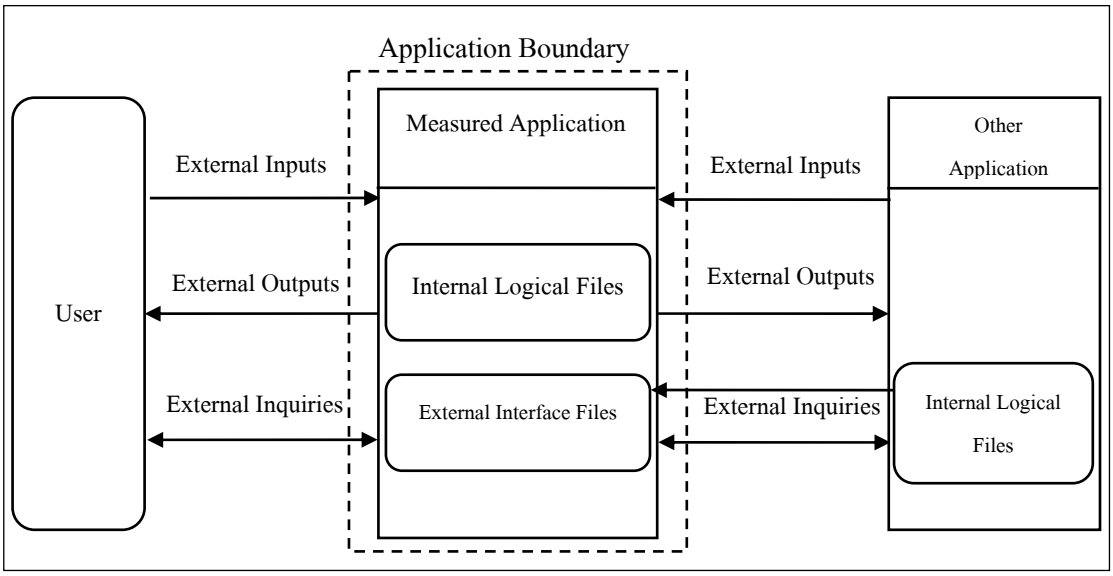
2) เอาท์พุทภายนอก (External Outputs)

3) การสอบถามภายนอก (External Inquiries)

4) แฟ้มข้อมูลภายใน (Internal Logical Files)

5) อินเตอร์เฟสแฟ้มข้อมูล (External Interface Files)

ในแต่ละส่วนประกอบจะมีการจำแนกระดับความซับซ้อนออกเป็น 3 ระดับ คือ ระดับความซับซ้อนต่ำ ระดับความซับซ้อนปานกลาง หรือ ระดับความซับซ้อนสูง



รูปที่ 2.2 ส่วนประกอบของฟังก์ชันพอยต์

โดยการนับฟังก์ชันพอต์และการกำหนดความซับซ้อนของส่วนประกอบนั้น ขึ้นอยู่กับปริมาณหน่วยข้อมูล และปัจจัยอื่นๆ ในแต่ละส่วนประกอบ ตามตารางที่ 2.2 (IFPUG, 2004)

ตารางที่ 2.2 ตารางการนับค่าฟังก์ชันพอยต์

ส่วนประกอบ	ความหมาย	ความซับซ้อน			รวม
		ต่ำ	ปานกลาง	สูง	
ILF	แฟ้มข้อมูลภายใน	☐ x 7	☐ x 10	☐ x 15	
EIF	อินเตอร์เฟสแฟ้มข้อมูล	☐ x 5	☐ x 7	☐ x 10	
EI	อินพุทภายนอก	☐ x 3	☐ x 4	☐ x 6	
EO	เอาท์พุทภายนอก	☐ x 4	☐ x 5	☐ x 7	
EQ	การสอบถามภายนอก	☐ x 3	☐ x 4	☐ x 6	
	รวมค่าฟังก์ชันพอยต์ที่ยังไม่ปรับค่า(UFP)				

ในการนับฟังก์ชันพอยต์จะต้องทำความเข้าใจกับคำศัพท์ต่อไปนี้ก่อน

ขบวนการมูลฐาน (Elementary Process) คือ หน่วยของกิจกรรมขั้นต้นที่ได้มาจากความต้องการของผู้ใช้กำหนด และ หน่วยของกิจกรรมต่างๆนี้เมื่อรวมเข้าด้วยกันก็จะเป็นการทำงานทั้งหมดของแอพพลิเคชัน ขบวนการมูลฐานจะต้องทำกิจกรรมต่างๆให้ครบถ้วนเช่น การเพิ่มข้อมูล

ลูกจ้างจะต้องทำการเพิ่มเงินเดือนและหน่วนงานสังกัด ฉะนั้นเมื่อจะกำหนดขบวนการมูลฐานจะต้องกำหนดให้ครบตามความต้องการจากผู้ใช้

ข้อมูลควมคุม (Control Information) คือข้อมูลที่เกี่ยวข้องกับขบวนการมูลฐานของแอพพิเคชันที่กำลังนับ โดยข้อมูลควมคุมต้องกำหนดว่ามีข้อมูลอะไรบ้าง ประมวลผลเมื่อไหร่ และมีขนวบการทำงานอย่างไร และข้อมูลควมคุมจะไม่เป็นข้อมูลที่เข้ามาบำรุงแฟ้มข้อมูลภายในแต่จะเปลี่ยนพฤติกรรมของระบบ ตัวอย่างเช่น ฝ่ายบัญชีกำหนดวงจรในการจ่ายค่าจ้าง ในวงจรการจ่ายค่าจ้างจะต้องเก็บเวลาจากแอพพลิเคชันในขณะที่ขบวนการมูลฐานของการจ่ายเงินค่าจ้างทำงาน

ผู้ใช้ (User) คือ เป็นบุคคลที่กำหนดฟังก์ชันของความต้องการ (Functional User Requirements) และรวมไปถึงบุคคลหรือสิ่งของที่จะใช้งานหรือสื่อสารกับซอฟต์แวร์นั้นๆ

ข้อกำหนดโดยผู้ใช้ (User Identifiable) คือการกำหนดความต้องการ ขนวบการ (Process) หรือกลุ่มข้อมูล (groups of data) ที่เป็นข้อตกลงรับรู้ทั้งผู้ใช้และนักพัฒนา

การเข้ามาบำรุง (Maintained) คือการเข้าบำรุงแฟ้มข้อมูลในระหว่างขบวนการมูลฐาน ประกอบด้วย การเขียน ลบ และแก้ไขแฟ้มข้อมูลภายใน หรือ อินเตอร์เฟสแฟ้มข้อมูล

ขบวนการตรรก (Processing logic) เป็นขนวบการพิเศษที่ผู้ใช้ต้องการให้มี เพื่อให้ขบวนการมูลฐานสมบูรณ์ขึ้น โดยจะประกอบด้วยการกระทำดังต่อไปนี้

1) ตรวจสอบความถูกต้องของอินพุท
2) มีการใช้สูตรทางคณิตศาสตร์คำนวณ เช่น รวมราคาสินค้าในใบเสร็จ
3) มีเปลี่ยนค่าตามหน่วยของข้อมูล เช่น เปลี่ยนค่าเงินบาทเป็นเงินสกุลอื่น
4) มีการกรองและเลือกข้อมูลเพื่อจะเลือกดูเฉพาะข้อมูลที่อยู่ในความสนใจและสามารถนำไปเปรียบเทียบกับข้อมูลชุดอื่นได้
5) ต้องมีการวิเคราะห์เงื่อนไขให้ตรงมาข้อกำหนด เช่น ขบวนการตรรกของขบวนการมูลฐานต้องมีรูปแบบสำหรับลูกจ้างรายวันและรายเดือน
6) หนึ่งหรือหลายแฟ้มข้อมูลมีการปรับปรุง
7) หนึ่งหรือหลายแฟ้มข้อมูลและการอินเตอร์เฟสแฟ้มข้อมูลมีการอ้างอิง เช่น เมื่อมีการเพิ่มข้อมูลลูกจ้างจะมีการติดอินเตอร์เฟสแฟ้มข้อมูลเพื่อคำนวณค่าจ้างจากเงินเดือนเป็นค่าจ้างรายวันสำหรับลูกจ้างรายวัน
8) ข้อมูลหรือข้อมูลควมคุมมีการค้นข้อมูลเพื่อแสดงออกมา เช่นเมื่อต้องการดูรายการขั้นเงินเดือนของลูกจ้างที่เป็นไปได้ จะต้องมีการแสดงรายการข้อมูลขั้นเงินเดือนออกมาทั้งหมด

9) ดีไรฟ์ดาต้าจะถูกสร้างขึ้นจากข้อมูลที่มีอยู่เพื่อเป็นข้อมูลเพิ่มเติม เช่น มีการสร้างรหัสคนไข้ขึ้นเพื่อใช้งาน SMI103 for Smith SMI มาจาก Smith 103 มาจากคนไข้ลำดับที่ 103
10) พฤติกรรมของระบบมีการเปลี่ยนแปลง เช่น พฤติกรรมขบวนการมูลฐานสำหรับขันตอนการจ่ายค่าจ้างจะเปลี่ยนไปเมื่อถึงกำหนดวันที่ 15 และวันสุดท้ายของเดือน
11) มีเตรียมและแสดงข้อมูลออกนอกขอบเขต
12) มีการรับข้อมูลหรือข้อมควบคุมเข้ามาในแอพพลิเคชัน
13) ข้อมูลจะต้องมีการเรียงลำดับข้อมูลในการจัดเก็บ

และในขบวนการมูลฐานอาจรวมขบวนการต่างๆ เข้าด้วยกัน เช่น มีการตรวจสอบข้อมูล มีการกรองข้อมูล และจัดเรียงข้อมูลใหม่ อยู่ในขบวนการมูลฐานอันเดียว และสามารถสรุปขบวนการตรรกสำหรับส่วนประกอบทรานเซคชันได้ดังตารางที่ 2.3

ตารางที่ 2.3 ขบวนการตรรกสำหรับส่วนประกอบทรานเซคชัน

ขบวนการตรรก	ส่วนประกอบทรานเซคชัน		
	EI	EO	EQ
1. ตรวจสอบความถูกต้องของอินพุท	c	c	c
2. มีการใช้สูตรทางคณิตศาสตร์คำนวณ	c	m*	n
3. มีเปลี่ยนค่าตามหน่วยของข้อมูล	c	c	c
4. มีการกรองและเลือกข้อมูล	c	c	c
5. ต้องมีการวิเคราะห์เงื่อนไข	c	c	c
6. หนึ่งหรือหลายแฟ้มข้อมูลมีการปรับปรุง	m*	m*	n
7. หนึ่งหรือหลายแฟ้มข้อมูลและการอินเตอร์เฟส แฟ้มข้อมูลมีการอ้างอิง	c	c	m
8. ข้อมูลหรือข้อมูลควมคุมมีการค้นข้อมูลเพื่อแสดงออกมา	c	c	m
9. ดีไรฟ์ดาต้าจะถูกสร้างขึ้นจากข้อมูลที่มีอยู่	c	m*	n
10. พฤติกรรมของระบบมีการเปลี่ยนแปลง	m*	m*	n
11. มีเตรียมและแสดงข้อมูลออกนอกขอบเขต	c	m	m
12. มีการรับข้อมูลหรือข้อมควบคุมเข้ามาในแอพพลิเคชัน	m	c	c
13. ข้อมูลจะต้องมีการเรียงลำดับข้อมูลในการจัดเก็บ	c	c	c

โดยที่

m หมายถึง ส่วนประกอบทรานเซคชันต้องทำขบวนการตรรกนี้

m* หมายถึง ส่วนประกอบทรานเซคชันนี้ต้องทำขบวนการตรรกอย่างน้อย 1 ขบวนการ

c หมายถึง ส่วนประกอบทรานเซคชันสามารถทำขบวนการตรรกนี้

n หมายถึง ส่วนประกอบทรานเซคชันไม่สามารถทำขบวนการตรรกนี้

และก่อนที่จะเริ่มนับส่วนประกอบฟังก์ชันพื้นฐานก็ต้องกำหนดขอบเขตแอพพลิเคชัน (Identify the counting scope and application boundary) เป็นขั้นตอนการกำหนดแอพพลิเคชันว่าส่วนประกอบได้จะอยู่ในหรือนอกขอบเขตแอพพลิเคชันบ้างแสดงในรูปที่ 2.2 โดยจะมีกฎของการกำหนดขอบเขตแอพพลิเคชันดังต่อไปนี้

1) การกำหนดขอบเขตแอพพลิเคชันจะต้องกำหนดในมุมมองของผู้ใช้ ให้ผู้ใช้สามารถเข้าใจและอธิบายถึงเหตุผลได้

2) การกำหนดขอบเขตแอพพลิเคชันออกจากแอพพลิเคชันที่ทำงานร่วมกันนั้นต้องเป็นเหตุจากฟังก์ชันทางธุรกิจไม่ใช่เหตุผลทางเทคนิค

3) เมื่อจะเพิ่มเติมโครงการจะต้องดูว่าจะสามารถยังอยู่ในขอบเขตได้หรือเปล่า หรือจะต้องมีการปรับปรุงขอบเขต

หลังจากการกำหนดขอบเขตแอพพลิเคชัน หัวข้อย่อยต่อไปจะกล่าวถึงส่วนประกอบฟังก์ชันพื้นฐานและการกำหนดปัจจัยความซับซ้อนของส่วนประกอบโดยมีรายละเอียดดังต่อไปนี้

1) อินพุทภายนอก (External Inputs[EI])

อินพุทภายนอกเป็นขบวนการมูลฐานที่เป็นขบวนการของข้อมูล หรือ ข้อมูลควบคุมที่มาจากภายนอกขอบเขตแอพพลิเคชัน โดยอินพุทภายนอกจะเข้ามาบำรุงแฟ้มข้อมูลภายในหรือเปลี่ยนพฤติกรรมของระบบ และจะมีฎกการกำหนดอินพุทภายนอกดังนี้

(1) เป็นข้อมูลหรือข้อมูลควมคุมที่รับมาจากภายนอกขอบเขตแอพพลิเคชัน

(2) แฟ้มข้อมูลภายในจะมีการเข้ามาบำรุงเมื่อมีอินพุทภายนอกเข้ามา ซึ่งอินพุทภายนอกนั้นจะเป็นข้อมูลไม่ใช่ข้อมูลควมคุมที่เปลี่ยนพฤติกรรมของระบบ

และจะนับเป็นอินพุทภายนอกก็ต่อเมื่อตรงตามกฎข้อใดข้อหนึ่งต่อไปนี้

(1) ขบวนการตรรกของอินพุทภายนอกนั้นต้องไม่ซ้ำกับขบวนการตรรกอื่นในแอพพลิเคชันนี้

(2) เขตข้อมูล (Data element) นั้นต่างจากอินพุทภายนอกอื่นในแอพพลิเคชันนี้

(3) แฟ้มข้อมูลภายในและอินเตอร์เฟสแฟ้มข้อมูลที่อ้างอิงเป็นไฟล์อ้างอิงต่างจากอินพุทภายนอกอื่นในแอพพลิเคชันนี้

และจะจำแนกระดับความซับซ้อนของอินพุตภายนอก ขึ้นอยู่กับจำนวนแฟ้มข้อมูล (File Type Referenced [FTR]) และจำนวนเขตข้อมูล (Data Element Type[DET]) ที่มีการเข้ามาบำรุงแฟ้มหรืออ้างอิง แสดงตามตารางที่ 2.4 โดยมีกฎการนับจำนวนแฟ้มข้อมูล (FTR) ดังต่อไปนี้

(1) นับเป็นจำนวนแฟ้มข้อมูลทั้งหมดจากแฟ้มข้อมูลภายในที่มีการเข้ามาบำรุง

(2) นับเป็นจำนวนแฟ้มข้อมูลทั้งหมดจากแฟ้มข้อมูลภายในและอินเตอร์เฟสแฟ้มข้อมูลที่มีการอ่านข้อมูลระหว่างขบวนการของอินพุตภายนอก

(3) นับจำนวนแฟ้มข้อมูลเป็นหนึ่งจากแฟ้มข้อมูลภายในที่มีการเข้ามาบำรุงและอ่านจากอินพุตภายนอกอันเดียวกัน

และมีกฎการนับจำนวนเขตข้อมูลหรือจำนวนฟิลด์ (DET) ดังต่อไปนี้

(1) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งทุกครั้งเมื่อมีการกำหนดจากผู้ใช้ให้มีการเข้ามาในขอบเขตแอพพลิเคชันของอินพุตภายนอก ทั้งนี้อินพุทภายนอกที่ใส่เข้ามาเป็นแบบวงซ้ำในหนึ่งฟิลด์ให้นับครั้งเดียว

(2) ไม่นับฟิลด์ (field) ที่รับข้อมูลมาจากการดีไรฟ์หรือรับค่ามาจากระบบ และบรรจุข้อมูลนั้นในแฟ้มข้อมูลภายใน ระหว่างขบวนการมูลฐานซึ่งไม่มีการส่งผ่านข้อมูลข้ามแอพพลิเคชัน

(3) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งเมื่อมีการส่งข้อความ มีข้อผิดพลาด การยืนยันการทำงานเสร็จสิ้น หรือ จะต้องทำงานต่อไป จากระบบออกนอกขอบเขตแอพพลิเคชัน

(4) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งจากการกระทำเดียวหรือหลายการกระทำที่มาจากขบวนการตรรกเดียว เช่น การกดปุ่มตกลงเพื่อบันทึกอินพุตภายนอก หรือ การปุ่มตกลงและปุ่มยืนยันเพื่อบันทึกข้อมูล นับเป็นขบวนการตรรกเดียว

จากจำนวนแฟ้มข้อมูล (FTR) และจำนวนเขตข้อมูล (DET) ที่นับได้ก็ทำการกำหนดระดับความซับซ้อนของอินพุทภายนอก แสดงตามตารางที่ 2.4

ตารางที่ 2.4 ระดับความซับซ้อนของอินพุทภายนอก

จำนวนแฟ้มข้อมูล (FTR)	จำนวนเขตข้อมูล (DET)		
	1 - 4	5 - 15	> 15
< 2	ต่ำ	ต่ำ	ปานกลาง
2	ต่ำ	ปานกลาง	สูง
> 2	ปานกลาง	สูง	สูง

2) เอาท์พุทภายนอก (External Outputs[EO])

เอาท์พุทภายนอกเป็นขบวนการมูลฐานที่เป็นขบวนการจัดส่งข้อมูล หรือ ข้อมูลควบคุมออกนอกขอบเขตแอพพลิเคชัน โดยเอาท์พุทภายนอกจะทำหน้าที่แสดงข้อมูลสารสนเทศไปยังผู้ใช้ผ่านขบวนการตรรก ซึ่งขบวนการตรรกจะต้องมีใช้สูตรคณิตศาสตร์ มีการคำนวณค่าต่างๆ มีการสร้างดีไรฟ์ดาต้า มีการเข้ามาบำรุงแฟ้มข้อมูลภายใน หรือ ทำให้พฤติกรรมของระบบมีการเปลี่ยนแปลง อย่างน้อยหนึ่งอย่าง และจะมีฎกการกำหนดเอาท์พุทภายนอกดังนี้

(1) เป็นฟังก์ชันที่ส่งข้อมูลหรือข้อมูลควบคุมไปภายนอกแอพพลิเคชัน โดยมีขบวนการตรรกต่อไปนี้อย่างน้อยหนึ่งขบวนการตรรก

(1) มีการใช้สูตรคณิตศาสตร์หรือมีการคำนวณค่า

(2) มีการสร้างสร้างดีไรฟ์ดาต้า

(3) มีการเข้ามาบำรุงแฟ้มข้อมูลภายใน

(4) ทำให้พฤติกรรมของระบบมีการเปลี่ยนแปลง

และจะนับเป็นเอาท์พุทภายนอกก็ต่อเมื่อตรงตามกฎข้อใดข้อหนึ่งต่อไปนี้

(1) ขบวนการตรรกของเอาท์พุทภายนอกนั้นต้องไม่ซ้ำกับขบวนการตรรกอื่นในแอพพลิเคชันนี้

(2) เขตข้อมูล (Data element) นั้นต่างจากเอาท์พุทภายนอกอื่นในแอพพลิเคชันนี้

(3) แฟ้มข้อมูลภายในและอินเตอร์เฟสแฟ้มข้อมูลที่อ้างอิงเป็นไฟล์อ้างอิงต่างจากเอาท์พุทภายนอกอื่นในแอพพลิเคชันนี้

และจะจำแนกระดับความซับซ้อนของเอาท์พุตภายนอกที่ขึ้นอยู่กับจำนวนแฟ้มข้อมูล (FTR) และจำนวนเขตข้อมูล (DET) ที่ถูกเข้ามาบำรุงแฟ้มหรืออ้างอิง โดยมีกฎการนับจำนวนแฟ้มข้อมูล (FTR) ที่มีการใช้กฎร่วมการสอบถามภายนอก (EQ) ดังต่อไปนี้

(1) นับเป็นจำนวนแฟ้มข้อมูลทั้งหมดจากแฟ้มข้อมูลภายในและอินเตอร์เฟสแฟ้มข้อมูลที่มีการอ่านข้อมูลระหว่างขบวนการของเอาท์พุตภายนอก

และใช้กฎสำหรับเอาท์พุตภายนอกอย่างเดียว

(1) นับเป็นจำนวนแฟ้มข้อมูลทั้งหมดจากแฟ้มข้อมูลภายในที่มีการเข้าบำรุงข้อมูลระหว่างขบวนการของเอาท์พุตภายนอก

(2) นับจำนวนแฟ้มข้อมูลเป็นหนึ่งจากแฟ้มข้อมูลภายในที่มีการเข้ามาบำรุงและอ่านจากเอาท์พุตภายนอกอันเดียวกัน

และมีกฎการนับจำนวนเขตข้อมูลหรือจำนวนฟิลด์ (DET) สำหรับเอาท์พุตภายนอกและการสอบถามภายนอกดังต่อไปนี้

(1) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งทุกครั้งเมื่อมีการกำหนดจากผู้ใช้ให้มีการเข้ามาในขอบเขตแอปพิเคชันเพื่อให้ส่งข้อมูล (field) ออกไปหรือใช้ขบวนการตรรกส่งข้อมูล (field) ออกไป

(2) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งทุกครั้งเมื่อมีการกำหนดจากผู้ใช้ให้ส่งข้อมูล (field) ออกไปนอกแอพพลิเคชัน

(3) นับจำนวนเขตข้อมูลของขบวนการตรรกการรับข้อมูลเข้าเพื่อสร้างขบวนการตรรกส่งข้อมูลออกนอกขอบเขต ให้นับเป็นขบวนการตรรกเดียว

(4) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งเมื่อมีการส่งข้อความ มีข้อผิดพลาด การยืนยันการทำงานเสร็จสิ้น หรือ จะต้องทำงานต่อไป จากระบบออกนอกขอบเขตแอพพลิเคชัน

(5) นับเป็นจำนวนเขตข้อมูลเป็นหนึ่งจากการกระทำเดียวหรือหลายการกระทำที่มาจากขบวนการตรรกเดียว เช่น การกดปุ่มตกลงเพื่อแสดงรายการ หรือการคลิกเลือกจำนวนฟิลด์ที่จะแสดงแล้วกดปุ่มตกลงเพื่อแสดงรายการนับเป็นขบวนการตรรกเดียว

(6) ไม่นับฟิลด์ (field) ที่รับข้อมูลมาจากการดีไรฟ์หรือรับค่ามาจากระบบ และบรรจุข้อมูลนั้นในแฟ้มข้อมูลภายใน ระหว่างขบวนการมูลฐานซึ่งไม่มีการส่งผ่านข้อมูลข้ามแอพพลิเคชัน

(7) ไม่นับส่วนประกอบของรายงานเป็นจำนวนเขตข้อมูล เช่น หัวข้อรายงานหรือหัวข้อของคอลัมภ์

(8) ไม่นับส่วนประกอบต่อไปนี้เป็นจำนวนเขตข้อมูล เลขหน้า ตำแหน่งของแถวคำสั่งของหน้ารายงาน (หน้าต่อไป หน้าที่แล้ว) หรือ เวลาที่แสดงรายงาน

จากจำนวนแฟ้มข้อมูล (FTR) และจำนวนเขตข้อมูล (DET) ที่นับได้ก็ทำการกำหนดระดับความซับซ้อนของอินพุทภายนอก แสดงตามตารางที่ 2.5

ตารางที่ 2.5 ระดับความซับซ้อนของเอาท์พุทภายนอก

จำนวนแฟ้มข้อมูล (FTR)	จำนวนเขตข้อมูล (DET)		
	1 - 5	6 - 19	> 20
< 2	ต่ำ	ต่ำ	ปานกลาง
2 – 3	ต่ำ	ปานกลาง	สูง
> 3	ปานกลาง	สูง	สูง

3) การสอบถามภายนอก (External Inquiries[EQ])

การสอบถามภายนอกเป็นขบวนการมูลฐานที่เป็นขบวนการจัดส่งข้อมูล หรือข้อมูลควบคุมออกนอกขอบเขตแอพพลิเคชัน โดยการสอบถามภายนอกจะทำหน้าที่แสดงข้อมูลสารสนเทศไปยังผู้ใช้ผ่านขบวนการตรรก ซึ่งขบวนการตรรกจะต้องมีไม่ใช้สูตรคณิตศาสตร์ ไม่มีการคำนวณค่าต่างๆ ไม่มีการสร้างดีไรฟ์ดาต้า ไม่มีการเข้ามาบำรุงแฟ้มข้อมูลภายใน หรือ ไม่ทำให้พฤติกรรมของระบบมีการเปลี่ยนแปลง และจะมีฎกการกำหนดการสอบถามภายนอกดังนี้

(1) เป็นฟังก์ชันที่ส่งข้อมูลหรือข้อมูลควบคุมไปภายนอกแอพพลิเคชัน โดยมีขบวนการตรรกดังต่อไปนี้

(1) รับข้อมูลหรือข้อมูลควบคุมจากแฟ้มข้อมูลภายในและอินเตอร์เฟสแฟ้มข้อมูล

(2) ไม่มีการใช้สูตรคณิตศาสตร์หรือมีการคำนวณค่า

(3) ไม่มีการสร้างสร้างดีไรฟ์ดาต้า

(4) ไม่มีการเข้ามาบำรุงแฟ้มข้อมูลภายใน

(5) ไม่ทำให้พฤติกรรมของระบบมีการเปลี่ยนแปลง

และจะนับเป็นเอาท์พุทภายนอกก็ต่อเมื่อตรงตามกฎข้อใดข้อหนึ่งต่อไปนี้

(1) ขบวนการตรรกของการสอบถามภายนอกนั้นต้องไม่ซ้ำกับขบวนการตรรกอื่นในแอพพลิเคชันนี้

(2) เขตข้อมูล (Data element) นั้นต่างจากการสอบถามภายนอกอื่นในแอพพลิเคชันนี้

(3) แฟ้มข้อมูลภายในและอินเตอร์เฟสแฟ้มข้อมูลที่อ้างอิงเป็นไฟล์อ้างอิงต่างจากการสอบถามภายนอกอื่นในแอพพลิเคชันนี้

และจะจำแนกระดับความซับซ้อนของการสอบภายนอกที่ขึ้นอยู่กับ จำนวนแฟ้มข้อมูล (FTR) และจำนวนเขตข้อมูล (DET) ที่ถูกเข้ามาบำรุงแฟ้มหรืออ้างอิงตามตารางที่ 2.6 โดยมีกฎการนับจำนวนแฟ้มข้อมูล (FTR) ตามห้วข้อเอาท์พุทภายนอก (EO)

ตารางที่ 2.6 ระดับความซับซ้อนของส่วนการสอบถามภายนอก

จำนวนแฟ้มข้อมูล (FTR)	จำนวนเขตข้อมูล (DET)		
	1 - 4	5 - 15	> 15
1	ต่ำ	ต่ำ	ปานกลาง
2 – 3	ต่ำ	ปานกลาง	สูง
> 3	ปานกลาง	สูง	สูง

4) แฟ้มข้อมูลภายใน (Internal Logical Files[ILF])

แฟ้มข้อมูลภายในเป็นกลุ่มข้อมูลหรือข้อมูลควมคุมจากข้อกำหนดโดยผู้ใช้ ที่อยู่ภายในแอพพลิเคชัน โดยแฟ้มข้อมูลภายในเป็นข้อมูลที่รับการเข้ามาบำรุงจากขบวนการมูลฐาน และแฟ้มข้อมูลภายในจะมีกฎการกำหนดแฟ้มข้อมูลภายในดังนี้

(1) เป็นกลุ่มข้อมูลหรือข้อมูลควมคุมจากข้อกำหนดโดยผู้ใช้

(2) เป็นกลุ่มข้อมูลที่รับการเข้ามาบำรุงจากขบวนการมูลฐาน

และมีกฎการนับจำนวนเขตข้อมูลหรือจำนวนฟิลด์ (DET) ดังต่อไปนี้

(1) นับเป็นจำนวนเขตข้อมูลทั้งหมดเมื่อมีการกำหนดจากผู้ใช้ จากแฟ้มข้อมูลภายในหรืออินเตอร์เฟสแฟ้มข้อมูลที่มีการเข้ามาบำรุง หรือ รับข้อมูลออกไปของขบวนการมูลฐาน

(2) เมื่อมีสองแอพพลิเคชันเข้ามาบำรุงหรืออ้างอิงถึงแฟ้มข้อมูลภายใน หรืออินเตอร์เฟสแฟ้มข้อมูลอันเดียวกันและเป็นการมาบำรุงหรืออ้างอิงที่ต่างกัน ให้นับจำนวนเขตข้อมูล (DET) ตามจำนวนที่แอพพลิเคชันนั้นใช้งาน

(3) นับเป็นจำนวนเขตข้อมูลทั้งหมดเมื่อมีการกำหนดจากผู้ใช้ และรวมถึงจำนวนเขตข้อมูล (DET) ที่มีความสัมพันธ์กันในแฟ้มข้อมูลภายใน หรืออินเตอร์เฟสแฟ้มข้อมูล

และมีกฎการนับจำนวนแรคคอร์ดข้อมูล (FTR) ดังต่อไปนี้

(1) นับจำนวนแรคคอร์ดข้อมูลตามกลุ่มย่อย(กลุ่มความสัมพันธ์ของแฟ้มข้อมูล) ของแฟ้มข้อมูลภายในหรืออินเตอร์เฟสแฟ้มข้อมูล

(2) หรือถ้าไม่มีกลุ่มย่อยให้นับเป็นหนึ่งจำนวนแรคคอร์ดข้อมูล

และจะจำแนกระดับความซับซ้อนของแฟ้มข้อมูลภายในที่ขึ้นอยู่กับ จำนวนแรคคอร์ดข้อมูล หรือจำนวนตารางข้อมูล (Record Element Types[RET]) และจำนวนเขตข้อมูล (Data Element Type[DET]) แสดงตามตารางที่ 2.7

ตารางที่ 2.7 ระดับความซับซ้อนของแฟ้มข้อมูลภายใน

จำนวนแรคคอร์ดข้อมูล (RET)	จำนวนเขตข้อมูล (DET)		
	1 - 19	20 – 50	> 50
1	ต่ำ	ต่ำ	ปานกลาง
2 – 5	ต่ำ	ปานกลาง	สูง
> 5	ปานกลาง	สูง	สูง

5) อินเตอร์เฟสแฟ้มข้อมูล (External Interface Files[EIF])

อินเตอร์เฟสแฟ้มข้อมูลเป็นกลุ่มข้อมูลหรือข้อมูลควมคุมจากข้อกำหนดโดยผู้ใช้อ้างอิงในแอพพลิเคชัน แต่มีการเข้ามาบำรุงจากนอกแอพพลิเคชัน โดยอินเตอร์เฟสแฟ้มข้อมูลเป็นข้อมูลที่รับการอ้างอิงจากขบวนการมูลฐาน ซึ่งหมายถึงอินเตอร์เฟสแฟ้มข้อมูลและเป็นแฟ้มข้อมูลภายในสำหรับแอพพลิเคชันอื่น และแฟ้มข้อมูลภายในจะมีกฎการกำหนดแฟ้มข้อมูลภายในดังนี้

(1) เป็นกลุ่มข้อมูลหรือข้อมูลควมคุมจากข้อกำหนดโดยผู้ใช้

(2) เป็นกลุ่มข้อมูลที่รับการอ้างอิงจากขบวนการมูลฐาน

(3) เป็นกลุ่มข้อมูลที่ไม่มีการเข้ามาบำรุงจากแอพพลิเคชันที่กำลังนับฟังก์ชันอยู่นี้

(4) เป็นกลุ่มข้อมูลที่เป็นแฟ้มข้อมูลภายในของแอพพลิเคชันอื่น

มีกฎการนับจำนวนเขตข้อมูลหรือจำนวนฟิลด์ (DET) และกฎการนับจำนวนแรคคอร์ดข้อมูล หรือ จำนวนตารางข้อมูล (RET) ตามหัวข้อแฟ้มข้อมูลภายใน และจำแนกระดับความซับซ้อนตารางที่ 2.8

ตารางที่ 2.8 ระดับความซับซ้อนของอินเตอร์เฟสแฟ้มข้อมูล

จำนวนแฟ้มข้อมูล (RET)	จำนวนเขตข้อมูล (DET)		
	1 - 19	20 - 50	> 50
1	ต่ำ	ต่ำ	ปานกลาง
2 – 5	ต่ำ	ปานกลาง	สูง
> 5	ปานกลาง	สูง	สูง

2.1.3 แฟคเตอร์ที่ปรับแต่งค่า (Value Adjustment Factor[VAF])

มีการพิจารณาคุณลักษณะทั่วไปของซอฟต์แวร์ 14 คุณลักษณะที่มีผลต่อความซับซ้อนในการออกแบบ พัฒนา ติดตั้ง และสนับสนุนการพัฒนาซอฟแวร์ โดยการประมาณการระดับความสำคัญของคุณลักษณะต่างๆต่อการออกแบบและการใช้งาน ค่าระดับความสำคัญแบ่งออกเป็น 6 ระดับ ตั้งแต่ 0 ถึง 5 ตามตารางที่ 2.9 เพื่อใช้คำนวณถ่วงน้ำหนักค่าฟังก์ชันพอยต์ตามแฟคเตอร์ที่ปรับแต่งค่าของซอฟต์แวร์

คุณลักษณะทั่วไปของซอฟต์แวร์ทั้ง 14 คุณลักษณะ ได้แก่

1) การสื่อสารข้อมูล (Data Communications)
   คุณลักษณะของซอฟต์แวร์ในด้านการสื่อสาร เพื่อประโยชน์ในการรับหรือส่งข้อมูลภายในแอพพลิเคชัน และความคุมแอพพลิเคชัน
2) การกระจาย (Distributed Data Processing)
   คุณลักษณะของซอฟต์แวร์ในด้านการกระจายข้อมูลหรือฟังก์ชันการประมวลผลของแอพพลิเคชัน
3) ประสิทธิภาพ (Performance)
   คุณลักษณะของซอฟต์แวร์ด้านประสิทธิภาพการตอบสนองในการประมวลผล
4) สภาพการใช้งานหนัก (Heavily Used Configuration)
   คุณลักษณะของซอฟต์แวร์ด้านการรองรับสภาพการใช้งานขนาดหนัก
5) อัตราทรานแซคชัน (Transaction Rate)
   คุณลักษณะของซอฟต์แวร์ด้านการรองรับอัตราไหลเวียนของทรานแซคชันต่อช่วงเวลาที่มีทรานแซคชันสูง
6) การป้อนข้อมูลออนไลน์ (Online Data Entry)
   คุณลักษณะของซอฟต์แวร์ด้านการป้อนข้อมูลผ่านทางออนไลน์

7) ประสิทธิภาพของผู้ใช้ปลายทาง (End-User Efficiency)
คุณลักษณะของซอฟต์แวร์ด้านการออกแบบและพัฒนาโดยมุ่งเน้นถึงระสิทธิภาพการใช้งานของผู้ใช้ปลายทาง

8) การปรับปรุงข้อมูลออนไลน์ (Online Update)
คุณลักษณะของซอฟต์แวร์ด้านการปรับปรุงข้อมูลในแฟ้มข้อมูลภายในแบบออนไลน์

9) ความซับซอนในการประมวลผล (Complex Processing)
คุณลักษณะของซอฟต์แวร์ด้านความซับซอนในการประมวลผลที่มีความสลับซับซ้อน เช่น
(1) มีการความคุมการโต้ตอบและการตัดสินใจของผู้ใช้
(2) มีตรรกที่ครอบคลุมเงื่อนไขครบถ้วนและมีการคำนวณทางคณิตศาสตร์
(3) มีการประมวลผลทวนเมื่อมีทรานแซคชันไม่สมบูรณ์

10) ความสามารถนำกลับมาใช้ใหม่ (Reusability)
คุณลักษณะของโค้ดที่ใช้ในซอฟต์แวร์ มีการออกแบบและพัฒนาให้สามารถนำกลับมาเวียนใช้ไหม่ในซอฟต์แวร์อื่น

11) การติดตั้งง่าย (Installation Ease)
คุณลักษณะของซอฟต์แวร์ด้านการดัดแปลง และติดตั้งได้ง่าย ซึ่งต้องมีการเตรียมแผนในการดัดแปลงและติดตั้ง และมีการทดสอบ

12) การใช้งานง่าย (Operational Ease)
คุณลักษณะของซอฟต์แวร์ด้านออกแบบและพัฒนาให้ง่ายต่อการใช้งานได้แก่ การจัดเตรียมโพรซิเดอร์ในการเดินเครื่องใช้งานซอฟต์แวร์ การสำรองข้อมูลและการกู้คืนสภาพข้อมูล ซึ่งซอฟต์แวร์จะลดขั้นตอนการทำงานด้วยมือให้เหลือน้อยที่สุด

13) การติดตั้งระบบในหลายพื้นที่ (Multiple Sites)
คุณลักษณะของซอฟต์แวร์ที่มีการออกแบบและพัฒนา เพื่อให้สามารถติดตั้งใช้งานได้ในหลายพื้นที่ หรือในหลายองค์กร

14) ความสะดวกในการเปลี่ยนแปลง (Facilitate Change)
คุณลักษณะของซอฟต์แวร์ที่มีการออกแบบและพัฒนาเพื่อให้มีความสะดวกต่อการเปลี่ยนแปลงซอฟต์แวร์

สำหรับค่าระดับความสำคัญของคุณลักษณะของซอฟต์แวร์ทั้ง 6 ระดับได้แก่

1) ไม่มี หรือมีแต่ไม่มีความสำคัญ = 0

2) ไม่สำคัญ = 1

3) สำคัญน้อย = 2

4) สำคัญปานกลาง = 3

5) สำคัญมาก = 4

6) สำคัญมากที่สุด = 5

ผลรวมของค่าระดับความสำคัญของของคุณลักษณะของซอฟต์แวร์ 14 คุณลักษณะ จะเป็นแฟคเตอร์ที่ปรับแต่กค่า (Value Adjustment Factor) ซึ่งมีค่าอยู่ระหว่าง 0.65 ถึง 1.35 (เพิ่มลดขนาดของฟังก์ชันพอยต์ได้ +/- 35 เปอร์เซนต์)

ตารางที่ 2.9 ตารางการปรับค่าตามความซับซ้อนในการประมวลผล

ลำดับ	คุณลักษณะ	ระดับ
1	การสื่อสารข้อมูล	....
2	การกระจาย	....
3	ประสิทธิภาพ	....
4	สภาพการใช้งานหนัก	....
5	อัตราทรานแซคชัน	....
6	การป้อนข้อมูลออนไลน์	....
7	ประสิทธิภาพของผู้ใช้ปลายทาง	....
8	การปรับปรุงข้อมูลออนไลน์	....
9	ความซับซ้อนในการประมวลผล	....
10	ความสามารถในการนำกลับมาใช่ใหม่	....
11	การติดตั้งง่าย	....
12	การใช้งานง่าย	....
13	การติดตั้งระบบในหลายพื้นที่	....
14	ความสะดวกในการเปลี่ยนแปลง	....
รวม		....

2.2 การวิเคราะห์ฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ

มาตรวัดซอฟต์แวร์สำหรับโปรแกรมเว็บ ได้มีการเสนอบทความทางวิชาการออกมาหลายบทความ โดยในปี ค.ศ. 2000 Donald J. Reifer ได้เสนอมาตรวัดซอฟต์แวร์แบบวัตถุเว็บ (Web Object) โดยใช้วัตถุเว็บมาเป็นตัวทำนายขนาดของโปรแกรมเว็บ (Web Object Predictors) มีรายละเอียดตามตารางที่ 2.10

ตารางที่ 2.10 การคำนวณวัตถุเว็บ (Web Object Calulation)

ตัวทำนายขนาดของโปรแกรมเว็บ	ความซับซ้อน		
	ต่ำ	ปานกลาง	สูง
จำนวนบล็อก (Number of building blocks)	1	2	4
จำนวนส่วนประกอบจากซอฟต์แวร์การค้าสำเร็จรูป (Number of COTS components)	2	4	6
จำนวนมัลติมีเดียไฟล์ (Number of multimedia files)	1	2	4
จำนวนอ็อบเจ็กต์และแอพพลิเคชันพอยต์ (Number of object or application points)	*	*	*
จำนวนเว็บส่วนประกอบ (Number of Web components)	2	4	6
จำนวน เอ็กซ์เอ็มแอล , เอสจีเอ็มแอล , เอชทีเอ็มแอล และบรรทัดคำสั่งสอบถาม (Number of xml, sgml, html, and query lines)	3	5	8
จำนวนกราฟิกส์ไฟล์ (Number of graphics files)	2	4	6
จำนวนสคริปต์ (Number of scripts)	1	2	3
อื่นๆ (Other)	2	4	6

จากตารางที่ 2.10 จะเห็นว่ามาตรวัดซอฟต์แวร์สำหรับโปรแกรมเว็บจะใช้วัตถุเว็บมาเป็นตัวประมาณขนาดของเว็บ โดยจะนับจากจำนวนของวัตถุเว็บและระดับความซับซ้อนของวัตถุเว็บนั้น ซึ่งค่าที่ได้ออกมาจะเป็นขนาดของโปรแกรมเว็บที่เป็นจำนวนวัตถุเว็บ (Number of Web Object) เพื่อใช้คำนวณขั้นต่อไป

และในปี ค.ศ. 2002 Donald J. Reifer ได้เสนอมาตรวัดแบบวัตถุเว็บอีกครั้งโดยได้นำเอามาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์เข้ามานับรวมกับวัตถุเว็บด้วยดังรายละเอียดในตารางที่ 2.11

ตารางที่ 2.11 การคำนวณวัตถุเว็บ (Web Object Calulation)

ตัวทำนายขนาดของโปรแกรมเว็บ	ความซับซ้อน			หมายเหตุ
	ต่ำ	ปานกลาง	สูง	
ส่วนประกอบฟังก์ชันพอยต์ Internal Logical Files (ILF) External Interface Files (EIF) External Inputs (EI) External Outputs (EO) External Inquires (EQ)	 7 5 3 4 3	 10 7 4 5 4	 15 10 6 7 6	ใช้หลักการนับฟังก์ชันพอยต์ของ IFPUG
จำนวนมัลติมีเดียไฟล์ (multi-media files)	4	5	7	ประกอบด้วย -ไฟล์วีดีโอ, ออดีโอ, ภาพ/กราฟิกส์ -มัลติมีเดียไฟล์แบบต่างๆ
จำนวนเว็บบล็อก (Web building blocks)	3	4	6	ประกอบด้วย -หมวดสินค้าออนไลน์ -ส่วนประกอบซอฟต์แวร์การค้าสำเร็จรูป -อื่นๆ เช่น แอปเพล็ต, เอเจ็นต์
จำนวนสคริปต์ (scripts [animation, audio, video, visual, etc.])	2	3	4	ประกอบด้วย -สคริปต์ต่างๆ -มาโคร, คอนเทรนเนอร์
จำนวนลิงค์ เอ็กซ์เอ็มแอล, เอชทีเอ็มแอล และ บรรทัดคำสั่งสอบถาม (Number of Links xml, html, and query lines)	3	4	6	การนับนับจำนวนที่มองเห็นได้ ไม่เข้าไปนับระดับซอร์สโค้ด (Logical line count)

ซึ่งจะเห็นว่ามาตรวัดซอฟต์แวร์สำหรับโปรแกรมเว็บจะใช้วัตถุเว็บ และ ส่วนประกอบของฟังก์ชันพอยต์ร่วมกันเป็นตัวประมาณขนาดของโปรแกรมเว็บ โดยจะนับจากจำนวนของวัตถุเว็บและส่วนประกอบของฟังก์ชันพอยต์เป็นขนาดของโปรแกรมเว็บ ที่เรียกว่าจำนวนวัตถุเว็บ (Number of Web Object) จากนั้นจะคำนวณให้เป็นจำนวนบรรทัดคำสั่ง (SLOC) เพื่อใช้ในการคำนวณขั้นต่อไป

2.3 การประมาณขนาดของซอฟต์แวร์ด้วยแบบจำลองโคโคโม2

โคโคโม2 (COCOMO II) เป็นแบบจำลองที่พัฒนาต่อจากโคโคโม ซึ่งได้พัฒนาครั้งแรกในปี ค.ศ. 1981 โดยการศึกษาโครงการซอฟต์แวร์ที่ได้รับการพัฒนาจนเสร็จสิ้นแล้วจำนวน 83 โครงการ แล้วนำมาวิเคราะห์ปัจจัยที่มีผลต่อค่าใช้จ่ายของการพัฒนาโครงการ ซึ่งสามารถสรุปได้ว่ามี 15 ปัจจัยหรือเรียกว่าตัวขับค่าใช้จ่าย (Cost Drivers) ที่มีผลต่อการพัฒนาโครงการ จากนั้นทีมงานผู้พัฒนาโคโคโมได้ปรับปรุงอย่างต่อเนื่องเป็นโคโคโม2 ในปี ค.ศ.1991 ซึ่งแบบจำลองโคโคโม2 จะประกอบด้วย 3 แบบจำลองย่อยคือ แบบจำลองแอพพลิเคชันคอมโพซิชัน (Application Composition Model) แบบจำลองเออรีดิไซด์ (Early Design Model) และ แบบจำลองโพสอาร์คิเท็กเจอร์ (Post Architecture Model) ซึ่งในแต่ละแบบจำลองย่อยจะมีวิธีการประมาณค่าที่ต่างกันไปขึ้นอยู่กับว่าผู้ใช้สามารถใช้ข้อมูลพื้นฐานได้ละเอียดมากน้อยเพียงใด (ศรินทร์ วัชรบุราคำ, 2542)

2.3.1 แบบจำลองแอพพลิเคชันคอมโพซิชัน

เป็นแบบจำลองที่ใช้ประมาณค่าใช้จ่ายในการพัฒนาซอฟต์แวร์อย่างหยาบๆ โดยพิจารณาจากส่วนที่ซอฟต์แวร์นั้นติดต่อกับผู้ใช้ (User Interface) ซึ่งในแบบจำลองนี้ผู้ใช้ทราบเพียงจำนวนรายงาน และ จำนวนหน้าจอ (Screens) ที่ใช้ในการรับข้อมูล (Input) และการแสดงผล (Output) เท่านั้น แล้วจึงนำข้อมูลเหล่านี้ไปคำนวณหาค่า อ็อบเจ็กต์พอยต์ (Object Points) และจำนวนคนที่ใช้พัฒนาซอฟต์แวร์ต่อเดือน ดังสมการที่ 2-1 และ 2-2

$$NOP = \frac{(ObjectPoints) \times (100 - \%Reuse)}{100} \tag{2-1}$$

$$PM = \frac{NOP}{PROD} \tag{2-2}$$

โดยที่

NOP หมายถึง จำนวนอ็อบเจ็กต์พอยต์ที่สร้างใหม่ (New Object Points)

%Reuse หมายถึง จำนวนเปอร์เซ็นต์ของหน้าจอ (Screens) รายงาน (Reports)และโมดูลของสามจีแอล (3GL modules) ที่นำกลับมาใช้

PM หมายถึง จำนวนนักพัฒนาต่อเดือน (Person Months)

PROD หมายถึง ค่าอัตราการผลิต ซึ่งจะขึ้นอยู่กับความสามารถและประสบการณ์ของนักพัฒนา

ขั้นตอนการคำนวณหาค่าประมาณการมีดังนี้

1) การหาจำนวนอ็อบเจ็กต์พอยต์

การหาจำนวนอ็อบเจ็กต์พอยต์ นับจาก จำนวนหน้าจอ จำนวนรายงานจำนวนองค์ประกอบของสามจีแอล (3GL Components) ที่มีในแอพพลิเคชันที่จะพัฒนา

2) การจัดกลุ่มให้กับอ็อบเจ็กต์พอยต์

เมื่อได้จำนวนอ็อบเจ็กต์พอยต์ทั้งหมดจากขั้นตอนแรกแล้ว จึงนำอ็อบเจ็กต์พอยต์เหล่านั้นมาจัดแบ่ง ออกเป็นกลุ่มๆ คือ ง่าย (Simple) ปานกลาง (medium) และยาก (difficult) โดยแบ่งตามเงื่อนไขดังตารางที่ 2.12

ตารางที่ 2.12 การแบ่งกลุ่มของอ็อบเจ็กต์พอยต์

สำหรับหน้าจอ(Screens)				สำหรับรายงาน(Reports)			
จำนวนหน้าจอที่ปรากฏ	จำนวนแหล่งข้อมูลในตาราง (source of data tables)			จำนวนชนิดของรายงาน	จำนวนแหล่งข้อมูลในตาราง (source of data tables)		
	รวม < 4 (< 2 srvr < 3 clnt)	รวม < 8 (2-3 srvr 3-5 clnt)	รวม > 8 (>3 srvr <5 clnt)		รวม < 4 (< 2 srvr <3 clnt)	รวม < 8 (2-3 srvr 3-5 clnt)	รวม > 8 (> 3 srvr > 5 clnt
< 3	ง่าย	ง่าย	ปานกลาง	0 หรือ 1	ง่าย	ง่าย	ปานกลาง
3-7	ง่าย	ปานกลาง	ยาก	2 หรือ 3	ง่าย	ปานกลาง	ยาก
> 8	ปานกลาง	ยาก	ยาก	> 4	ปานกลาง	ยาก	ยาก

หมายเหตุ srvr หมายถึง ให้บริการ(Server), clnt หมายถึง รับบริการ (Client)

3) การให้น้ำหนักกับอ็อบเจ็กต์พอยต์ที่ได้จัดแบ่งกลุ่มจากข้อ 2 แล้ว โดยการให้น้ำหนักขึ้นอยู่กับกลุ่มที่แบ่งดังตารางที่ 2.13

ตารางที่ 2.13 การกำหนดน้ำหนักของอ็อบเจ็กต์พอยต์

ชนิดของอ็อบเจ็กต์พอยต์	การกำหนดน้ำหนัก		
	ง่าย	ปานกลาง	ยาก
หน้าจอ	1	2	3
รายงาน	2	5	8
จำนวนองค์ประกอบของสามจีแอล	-	-	10

4) นำจำนวนอ็อบเจ็กต์พอยต์ทั้งหมดมาคูณกับน้ำหนักที่ได้ในแต่ละกลุ่ม แล้วนำมาบวกรวมกัน ก็จะได้ค่า อ็อบเจ็กต์พอยต์ ในสมการที่ 2-1
5) การหาค่าอัตราการผลิตสามารถหาได้จากตารางที่ 2.14

ตารางที่ 2.14 ค่าอัตราการผลิต

ประสบการณ์และความสามารถของผู้พัฒนา (Developers experience and capability)	ต่ำมาก	ต่ำมาก	ปานกลาง	สูง	สูงมาก
ความสมบูรณ์และความสามารถของไอเคส (ICASE maturity and capability)	ต่ำมาก	ต่ำมาก	ปานกลาง	สูง	สูงมาก
ค่าอัตราการผลิต(Productivity Rate:PROD)	4	7	13	25	50

2.3.2 แบบจำลองเออรีดิไซด์

แบบจำลองเออรีดิไซด์ เป็นแบบจำลองที่มีความละเอียดของการประมาณค่าใช้จ่ายในการพัฒนาซอฟต์แวร์มากกว่าแบบจำลองแรก เนื่องจากผู้ใช้ทราบข้อมูลเพียงคร่าวๆเท่านั้น เช่น ทราบจำนวนฟังก์ชันพอยท์(Function points) และ ภาษาที่ใช้พัฒนา แต่ในการประมาณค่าใช้จ่าย (cost driver) 7 ลักษณะ เป็นข้อมูลพื้นฐานในการประมาณค่าใช้จ่ายซอฟต์แวร์ด้วย ซึ่งผู้ใช้ต้องเป็นผู้กำหนดระดับ(rating)ของตัวขับค่าใช้จ่ายทั้ง 7 ลักษณะ ดังนี้

1) ความเชื่อถือได้และความซับซ้อนของผลิตภัณฑ์ (Product Reliability and Complexity[RCPX]) พิจารณาจาก ความเชื่อถือได้ของผลิตภัณฑ์ซอฟต์แวร์ที่ต้องการ ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์ และ ความต้องการเอกสารที่ตรงกับวงจรชีวิต
2) ความต้องการนำกลับมาใช้ (Required Reuse[RUSE]) พิจารณาจากการออกแบบเพื่อมีการเตรียมนำกลับมาใช้ใหม่
3) แพลทฟอร์มที่แตกต่างกัน (Platform Difficulty[PDIF]) พิจารณาจากการกำจัดเวลาที่ใช้ในการประมวลผล, ขนาดของหน่วยความจำหลัก และการเปลี่ยนแพลทฟอร์ม
4) ความสามารถของบุคลากร (Personnel Capability[PERS]) พิจารณาจากความสามารถในการวิเคราะห์ ความสามารถของโปรแกรมเมอร์ และการทำงานอย่างต่อเนื่องของบุคลากร

5) ประสบการณ์ของบุคลากร (Personnel Experience[PREX]) พิจารณาจาก ประสบการณ์การใช้แอพพลิเคชัน ประสบการณ์การใช้แพลทฟอร์ม และ ประสบการณ์การใช้โปรแกรมภาษาและเครื่องมือ

6) อุปกรณ์อำนวยความสะดวก (Facilities[FCIL]) พิจารณาจาก การใช้เครื่องมือซอฟต์แวร์ และ การพัฒนาในหลายสถานที่

7) กำหนดการการพัฒนา (Schedule[SCED]) พิจารณาจาก กำหนดการการพัฒนาที่ต้องการ

การประมาณค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ในแบบจำลองนี้ จะประมาณระยะเวลาที่ใช้ในการพัฒนาซอฟต์แวร์ จำนวนคนที่ใช้พัฒนาต่อเดือน เป็นต้น ซึ่งสมการสำหรับประมาณการดังสมการที่ 2-3 ถึง 2-10 และคำอธิบายในตารางที่ 2.15

สูตรคำนวณจำนวนคนที่ใช้พัฒนาต่อเดือนคือ

$$PM = A \times [Size]^{B} \prod_{i=1}^{7} EM_{i} + PM_{AT} \tag{2-3}$$

โดยที่

$$PM_{AT} = \frac{ASLOC\left[\frac{AT}{100}\right]}{ATPROD} \tag{2-4}$$

และ

$$B = 1.01 + 0.01 \sum_{J=1}^{5} SF_{j} \tag{2-5}$$

สูตรคำนวณขนาดของซอฟต์แวร์คือ

$$Size = \overline{Size}\left[1 + \frac{BRAK}{100}\right] \tag{2-6}$$

โดยที่

$$\overline{\text{Size}} = \text{KNSLOC} + \text{KASLOC} \times \left[\frac{100 - \text{AT}}{100} \right] \times (\text{AAM}) \tag{2-7}$$

$$\text{AAM} \begin{cases} \dfrac{\text{AA} + \text{AAF} \times (1 + 0.02 \times \text{SU} \times \text{UNFM})}{100}, \text{AAF} \le 0.05 \\ \dfrac{\text{AA} + \text{AAF} + \text{SU} \times \text{UNFM}}{100}, \text{AAF} > 0.05 \end{cases} \tag{2-8}$$

$$\text{AAF} = 0.4 \times \text{DM} + 0.3 \times \text{CM} + 0.3 \times \text{IM} \tag{2-9}$$

สูตรคำนวณจำนวนระยะเวลาที่ใช้พัฒนาซอฟ์แวร์คือ

$$\text{TDEV} = \left[3.0 \times (\text{PM})^{(0.33+0.2\times(\text{B}-1.01))} \right] \frac{\text{SECD\%}}{100} \tag{2-10}$$

ตารางที่ 2.15 สัญลักษณ์และคำอธิบายของสมการจากเออรีดิไซด์

สัญลักษณ์	คำอธิบาย
A	ค่าคงที่ กำหนดให้เป็น 2.5
AA	การประเมินสิ่งที่สามารถนำมาใช้ได้(Assessment and assimilation)
AT	เปอร์เซ็นต์ขององค์ประกอบ(Components)ที่สามารถแปลงได้โดยอัตโนมัติ (automatically translated)
ATPROD	ตัวคูณของการแปลงโดยอัตโนมัติ(Automatic translation productivity)
BRAK	เปอร์เซ็นต์ของการเปลี่ยนแปลงที่มีในความต้องการ(Requirements)
CM	เปอร์เซ็นต์ของการเปลี่ยนแปลงโปรแกรม(Percentage of code modified)
DM	เปอร์เซ็นต์ของการเปลี่ยนแปลงการออกแบบ(Percentage of design modified)
EM	ตัวคูณความพยายาม(Effort Multipliers : RCPX, RUSE, PDIF, PERS, PREX, FCIL, SCED)
IM	เปอร์เซ็นต์ของการเปลี่ยนแปลงการรวมและการทดสอบระบบ (Percentage of integration and test modified)
KASLOC	ขนาดของโปรแกรมที่ได้รับการปรับปรุง มีหน่วยเป็น หนึ่งพันบรรทัด
KNSLOC	ขนาดของโปรแกรมที่สร้างใหม่ มีหน่วยเป็น หนึ่งพันบรรทัด
PM	จำนวนนักพัฒนาต่อเดือน
SECD%	เปอร์เซ็นต์ของการลดและเพิ่มเวลาที่ใช้พัฒนาซอฟต์แวร์
SF	ค่าสเกลของปัจจัย(Scale Factors : PREC, FLEX, RESL, TEAM, PMAT)
SU	ระดับความเข้าใจของผู้พัฒนาที่มีต่อโปรแกรม(Software Understanding)
TDEV	เวลาที่ใช้พัฒนาซอฟต์แวร์มีหน่วยเป็นเดือน(Time to develop)
UNFM	ความไม่คุ้นเคยของโปรแกรมเมอร์ที่มีต่อซอฟต์แวร์(Programmer Unfamiliarity with Software)

2.3.3 แบบจำลองโพสอาร์คิเท็กเจอร์

แบบจำลองโพสอาร์คิเท็กเจอร์ เป็นแบบจำลองที่ใช้ประมาณค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ ซึ่งขยายขีดความสามารถของแบบที่สอง เนื่องจากผู้ใช้ต้องทราบข้อมูลเกี่ยวกับการพัฒนาซอฟต์แวร์ละเอียดขึ้น โดยผู้ใช้ต้องเป็นผู้ให้ข้อมูลเกี่ยวกับจำนวนบรรทัดคำสั่งของซอฟต์แวร์ที่จะพัฒนา และ ระดับของตัวขับค่าใช้จ่ายจากเดิม 7 ลักษณะเป็น 17 ลักษณะ เป็นข้อมูล

พื้นฐานในการประมาณค่าใช้จ่ายซอฟต์แวร์ซึ่งตัวขับค่าใช้จ่ายทั้ง 17 ลักษณะแบ่งออกได้เป็น 4 กลุ่มใหญ่ๆ ดังนี้

1) ปัจจัยกลุ่มผลิตภัณฑ์(Product Factors)

(1) ความเชื่อถือได้ของผลิตภัณฑ์ซอฟต์แวร์ที่ต้องการ (Required Software Reliability[RELY]) ซึ่งจะพิจารณาจากความสูญเสียที่เกิดขึ้นเมื่อระบบไม่สามารถทำงานได้

(2) ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ (Data Base Size [DATA]) พิจารณาจากขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ ซึ่งคำนวณได้จากขนาดของฐานข้อมูลหารด้วยขนาดของโปรแกรม

(3) ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์ (Complexity[CPLX]) การพิจารณาความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์จะพิจารณา 5 อย่างด้วยกัน

- การดำเนินการควบคุม (Control Operation)
  การดำเนินการควบคุม หมายถึง กระบวนการที่ใช้ควบคุมการทำงานของโปรแกรมหรือลักษณะการทำงานของโปรแกรมในซอฟต์แวร์ที่จะพัฒนา
- การดำเนินการคำนวณ (Computation Operation)
  การดำเนินการคำนวณ หมายถึง วิธีการหรือรูปแบบการคำนวณที่มีใช้ในซอฟต์แวร์ที่จะพัฒนา
- การดำเนินการที่เกี่ยวกับอุปกรณ์ (Device-dependent Operation)
  การดำเนินการที่เกี่ยวกับอุปกรณ์ หมายถึง วิธีหรือกระบวนการที่ซอฟต์แวร์ที่จะพัฒนาใช้งานอุปกรณ์ เช่นการสั่งให้เครื่องพิมพ์เอกสารพิมพ์รายงานการจัดเก็บข้อมูลในฮาร์ดดิสก์ เป็นต้น
- การดำเนินการเกี่ยวกับการจัดการกับข้อมูล (Data Management Operation)
  การดำเนินการเกี่ยวกับการจัดการกับข้อมูล หมายถึง วิธีการจัดการเก็บข้อมูล และการใช้งานข้อมูลซอฟต์แวร์ที่จะพัฒนา
- การดำเนินการส่วนต่อประสานกับผู้ใช้ (User interface Management operation)
  การปฏิบัติการส่วนต่อประสานกับผู้ใช้ หมายถึง วิธีหรือรูปแบบติดต่อกับผู้ใช้ของซอฟต์แวร์ที่จะพัฒนา

(4) ความต้องการเอกสารที่ตรงกับวงจรชีวิต (Documentation match to life-cycle needs: [DOCU]) พิจารณาถึงความละเอียดของเอกสารในทุกวงจรชีวิต

(5) ความต้องการที่จะนำผลิตภัณฑ์ซอฟต์แวร์กลับมาใช้ใหม่ (Required Reusability [RUSE]) พิจารณาจากการออกแบบผลิตภัณฑ์ซอฟต์แวร์ที่มีการเตรียมสำหรับการนำกลับมาใช้ใหม่

2) ปัจจัยกลุ่มแพลทฟอร์ม (Platform Factors)

(1) การจำกัดเวลาที่ใช้ในการประมวลผล (Execution Time Constraint [Time]) พิจารณาจากเวลาที่ใช้ในการกระทำการของซอฟต์แวร์ระบบ

(2) การจำกัดหน่วยความจำหลัก (Main Storage Constraint [STOR]) พิจารณาขนาดของหน่วยความจำหลักซึ่งจะขึ้นกับความสามารถของซอฟต์แวร์ระบบ

(3) การเปลี่ยนแพลทฟอร์ม (Platform Volatility [PVOL]) แพลทฟอร์มในที่นี้รวมทั้งฮาร์ดแวร์และซอฟต์แวร์ ซึ่งจะพิจารณาจากระยะเวลาที่สามารถใช้งานระบบได้ก่อนมีการเปลี่ยนแปลง

3) ปัจจัยกลุ่มบุคลากร (Personal Factor)

(1) ความสามารถในการวิเคราะห์ (Analyst Capability [ACAP]) พิจารณาจากการวิเคราะห์และออกแบบระบบของทีมที่พัฒนา

(2) ความสามารถของโปรแกรมเมอร์ (Programmer Capability [PCAP]) พิจารณาจากความสามารถของทีมที่พัฒนาโปรแกรม

(3) ประสบการณ์ใช้แอพพลิเคชัน (Applications Experience [APEX]) พิจารณาจากประสบการณ์ที่เคยใช้แอพพลิเคชันประเภทเดียวกับที่จะพัฒนาระบบใหม่ โดยจะพิจารณาจากประสบการณ์โดยรวมของทีมพัฒนา

(4) ประสบการณ์แพลทฟอร์ม (Platform Experience [PLEX) พิจารณาจากการใช้แพลทฟอร์มได้เป็นอย่างดีของทีมที่พัฒนา เช่น มีการใช้รูปภาพ การใช้ส่วนต่อประสานกับผู้ใช้การใช้ฐานข้อมูลหรือ การใช้เครือข่าย

(5) ประสบการณ์โปรแกรมภาษาและเครื่องมือ (Language and Tool Experience [LTEX]) พิจารณาจากประสบการณ์ของทีมที่พัฒนาว่าสามารถใช้เครื่องมือทำให้ตรงตามความต้องการ และเวลาที่ใช้ในการพัฒนาระบบ

(6) ความต่อเนื่องของบุคลากร (Personal Continuity [PCON]) พิจารณาจากการหมุนเวียนของบุคลากรในทีมพัฒนา

4) ปัจจัยกลุ่มโครงการ (Project Factor)

(1) การใช้เครื่องมือซอฟต์แวร์ (Use of Software tools [TOOL]) พิจารณาจากเครื่องมือที่ใช้ในการพัฒนาระบบว่ามีขีดความสามารถเพียงใด

(2) การพัฒนาในหลายสถานที่ (Multisite Development [SITE]) พิจารณาสถานที่และรูปแบบการสื่อสารในระหว่างการพัฒนาระบบ

(3) กำหนดตารางการพัฒนาที่ต้องการ (Required Development Schedule [SCED]) พิจารณาจากระยะเวลาที่ใช้ในการพัฒนาระบบ

โดยค่าตัวขับค่าใช้จ่ายทั้ง 17 ค่านั้นการกำหนดระดับสามารถดูรายละเอียดได้ที่ ภาคผนวก ข. การประมาณค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ในแบบจำลองนี้ จะประมาณระยะเวลาที่ใช้ในการพัฒนาซอฟต์แวร์ จำนวนคนที่ใช้พัฒนาต่อเดือน เป็นต้น ซึ่งมีสมการสำหรับประมาณการดังสมการที่ 2-11 ถึง 2-18 และคำอธิบายในตารางที่ 2.16

สูตรคำนวณจำนวนคนที่ใช้พัฒนาต่อเดือนคือ

$$PM = A \times [Size]^{B} \prod_{i=1}^{17} EM_i + PM_{AT} \tag{2-11}$$

โดยที่

$$PM_{AT} = \frac{ASLOC\left[\frac{AT}{100}\right]}{ATPROD} \tag{2-12}$$

และ

$$B = 1.01 + 0.01 \sum_{J=1}^{5} SF_j \tag{2-13}$$

สูตรคำนวณขนาดของซอฟต์แวร์คือ

$$Size = \overline{Size}\left[1 + \frac{BRAK}{100}\right] \tag{2.14}$$

โดยที่

$$\overline{\text{Size}} = \text{KNSLOC} + \text{KASLOC} \times \left[\frac{100 - \text{AT}}{100} \right] \times (\text{AAM}) \tag{2-15}$$

$$\text{AAM} \begin{cases} \dfrac{\text{AA} + \text{AAF} \times (1 + 0.02 \times \text{SU} \times \text{UNFM})}{100}, \text{AAF} \le 0.05 \\ \dfrac{\text{AA} + \text{AAF} + \text{SU} \times \text{UNFM}}{100}, \text{AAF} > 0.05 \end{cases} \tag{2-16}$$

$$\text{AAF} = 0.4 \times \text{DM} + 0.3 \times \text{CM} + 0.3 \times \text{IM} \tag{2-17}$$

สูตรคำนวณจำนวนระยะเวลาที่ใช้พัฒนาซอฟต์แวร์คือ

$$\text{TDEV} = \left[3.0 \times (\text{PM})^{(0.33+0.2\times(\text{B}-1.01))} \right] \frac{\text{SECD\%}}{100} \tag{2-18}$$

ตารางที่ 2.16 สัญลักษณ์และคำอธิบายของสมการจากโพสอาร์คิเท็กเจอร์

สัญญาลักษณ์	คำอธิบาย
A	ค่าคงที่ กำหนดให้เป็น 2.5
AA	การประเมินสิ่งที่สามารถนำมาใช้ได้ (Assessment and assimilation)
AT	เปอร์เซ็นต์ขององค์ประกอบที่สามารถแปลงได้โดยอัตโนมัติ (Percentage of components are automatically translated)
ATPROD	ตัวคูณของการแปลงอัตโนมัติ (Automatic translation productivity)
BRAK	เปอร์เซ็นต์ของการเปลี่ยนแปลงที่มีความต้องการ (Requirements)
CM	เปอร์เซ็นต์ของการเปลี่ยนแปลงโปรแกรม(Percentage of code modified)
DM	เปอร์เซ็นต์ของการเปลี่ยนแปลงการออกแบบ(Percentage of design modified)
EM	ตัวคูณความพยายาม (Effort Multipliers: RELY, DATA, CPLX, RUSE, DOCU,TIME ,STOR,PVOL,ACAP, PCAP, APEX,PLXP, LTEX,PCON,TOOL,SITE,SCED) ดังแสดงในตารางที่ 2.17
IM	เปอร์เซ็นต์ของการเปลี่ยนแปลงการรวมและการทดสอบระบบ (percentage of integration and test modified)
KASLOC	ขนาดของโปรแกรมที่ได้รับการปรับปรุง มีหน่วยเป็น หนึ่งพันบรรทัด
PM	จำนวนนักพัฒนาต่อเดือน
SECD%	เปอร์เซ็นต์ของการลดและเพิ่มเวลาที่ใช้พัฒนาซอฟต์แวร์
SF	ค่าสเกลของปัจจัย (scale factors : PREC, FLEX,RESL,TEAM,PMAT)
SU	ระดับความเข้าใจของผู้พัฒนาที่มีต่อโปรแกรม (Software understanding)
TDEV	เวลาที่ใช้พัฒนาซอฟต์แวร์มีหน่วยเป็นเดือน (Time to develop)
UNFM	ความไม่คุ้นเคยของโปรแกรมเมอร์ที่มีต่อซอฟต์แวร์ (programmer unfamiliarity with software)

ตารางที่ 2.17 ค่าตัวคูณความพยายามในแต่ละระดับ

ตัวขับค่าใช้จ่าย	ระดับ					
	ต่ำมาก	ต่ำ	ปานกลาง	สูง	สูงมาก	สูงที่สุด
RELY	0.75	0.88	1.00	1.15	1.39	
DATA		0.93	1.00	1.09	1.19	
CPLX	0.75	0.88	1.00	1.15	1.30	1.66
RUSE		0.91	1.00	1.14	1.29	1.49
DOCU	0.89	0.95	1.00	1.06	1.13	
TIME			1.00	1.11	1.31	1.67
STOR			1.00	1.06	1.21	1.57
PVOL		0.87	1.00	1.15	1.30	
ACAP	1.50	1.22	1.00	0.83	0.67	
PCAP	1.37	1.16	1.00	0.87	0.74	
APEX	1.22	1.10	1.00	0.89	0.81	
PLEX	1.25	1.10	1.10	0.88	0.81	
PCON	1.24	1.10	1.00	0.92	0.84	
LTEX	1.22	1.10	1.00	0.91	0.84	
TOOL	1.24	1.12	1.00	0.86	0.72	
SITE	1.25	1.10	1.00	0.92	0.84	0.78
SCED	1.29	1.10	1.00	1.00	1.00	

จากหัวข้อทั้งหมดที่กล่าวมาจะเห็นได้ว่าการพัฒนาโปรแกรมเว็บในปัจจุบัน จะเป็นการพัฒนาโปรแกรมเว็บแบบพลวัต (Dynamic web) ที่มีการเชื่อมต่อกันระบบฐานข้อมูลหรือในบางโครงการใช้โปรแกรมเว็บเป็นโปรแกรมระบบฐานข้อมูล การใช้มาตรวัดแบบฟังก์ชันพอยต์ที่มีการพัฒนาอย่างต่อเนื่อง ที่สามารถเป็นใช้มาตรวัดซอฟต์แวร์สำหรับโปรแกรมฐานข้อมูลจึงสามารถนำมาปรับใช้เป็นมาตรวัดซอฟต์แวร์สำหรับโปรแกรมเว็บที่ทำงานร่วมกับฐานข้อมูลได้ โดยที่จะนำเอาส่วนประกอบใดของเว็บที่เป็นส่วนประกอบที่มีผลต่อขนาดของโปรแกรม และ สามารถนับออกมาได้ในช่วงการกำหนดความต้องการของซอฟต์แวร์ เข้ามาเป็นส่วนประกอบในการนับสำหรับมาตรวัดซอฟต์แวร์ใหม่ที่จะทำการพัฒนาขึ้น

บทที่ 3
วิธีดำเนินการวิจัย

มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ เป็นมาตรวัดขนาดของซอฟต์แวร์ที่ใช้การคำนวณแบบฟังก์ชันพอยต์เป็นหลัก แต่มีการเพิ่มส่วนประกอบเว็บ (Web Component) เพื่อปรับให้เหมาะสมกับโปรแกรมเว็บ และในงานวิจัยนี้จะทำการศึกษาโปรแกรมเว็บที่พัฒนาจากโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล โดยการรวบรวมโปรแกรมเว็บที่พัฒนาเสร็จสมบูรณ์แล้วมาเป็นโปรแกรมต้นแบบจำนวน 10 โครงการ แล้วสร้างสมการประมาณขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่ง ซึ่งสมการดังกล่าวนี้จะใช้ประมาณขนาดของโปรแกรมเว็บที่จะทำการพัฒนาในอนาคต

3.1 ระเบียบวิธีวิจัย

3.1.1 ศึกษาค้นคว้าและรวบรวมงานวิจัยที่เกี่ยวข้อง

3.1.2 การรวมรวมโปรแกรมต้นแบบ

3.1.3 การวิเคราะห์ฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บจากความต้องการซอฟต์แวร์

3.1.4 สร้างสมการประมาณขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่ง

3.1.5 พัฒนามาตรวัดซอฟต์แวร์

3.1.6 ประเมินผลมาตรวัดโดยข้อมูลทดสอบ

3.1.7 การคำนวณจำนวนความพยายามและระยะเวลาการพัฒนาด้วยแบบจำลองโคโคโม2

3.2 มาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ

มาตรวัดซอฟต์แวร์สำหรับโปรแกรมเว็บ ในยุคเริ่มต้นโปแกรมเว็บยังเป็นโปรแกรมเว็บแบบสถิต (Static Web) จึงยังใช้เพียงมาตรวัดที่เป็นวัตถุเว็บ (Web Object) เท่านั้น แต่เมื่อมีการนำระบบฐานข้อมูลเข้ามาใช้กับโปรแกรมเว็บ ทำให้โปรแกรมเว็บปัจจุบันเป็นโปรแกรมเว็บแบบพลวัต (Dynamic web) จึงทำให้โครงสร้างของโปรแกรมภาษามีความเหมือนกับโครงสร้างของโปรแกรมภาษาอื่นที่ทำงานร่วมกับฐานข้อมูล จึงได้นำเอามาตรวัดแบบฟังก์ชันพอยต์เข้ามาใช้กับโปรแกรมเว็บ โดยงานวิจัยเกี่ยวกับมาตรวัดสำหรับโปรแกรมเว็บนั้นได้มีการเสนอกันในหลายรูปแบบ และจนถึงขณะนี้ยังไม่มีการกำหนดมาตรวัดมาตรฐานสำหรับโปรแกรมเว็บ(Shu, ed, 2004)

งานวิจัยนี้จึงได้นำส่วนประกอบเว็บที่เป็นมัลติมีเดียไฟล์ (Multi-Media Files MMF) นำเข้าไปเป็นส่วนประกอบในการนับฟังก์ชันพอยต์ (Function Point Count) ตารางที่ 2.1 จะเห็นว่ามีการนับฟังก์ชันพอยต์ตั้งแต่ขันตอน เสนอโครงร่างโครงการ (Proposal) แต่ขั้นตอนการเสนอโครงร่างไม่ได้เป็นขั้นตอนการพัฒนาโครงการทางวิศวกรรมซอฟต์แวร์ ทำให้การนับฟังก์ชันพอยต์เริ่มที่ขั้นตอน การกำหนดความต้องการซอฟต์แวร์ (Requirements) การออกแบบ (Design) การเขียนโปรแกรม (Construction) การส่งมอบ (Delivery) และ การบำรุงรักษา (Corrective Maintenance) ตามลำดับซึ่งในแต่ละขั้นตอนจะมีรายละเอียดของโครงการมากขึ้นเรื่อยๆ แต่ข้อมูลที่จะใช้นับฟังก์ชันพอยต์นั้นจะเริ่มต้นที่ขั้นตอนกำหนดความต้องการ ซึ่งในขั้นตอนนี้จะได้ความต้องการขั้นสุดท้าย (Final Requirements) และอาจมีข้อมูลอื่นประกอบด้วยเช่น โมเดลของข้อมูลและขบวนการ (Data and process models) วินโดว์ส(Windows) หน้าจอภาพ (screens) หรือ หน้าจอภาพ(reports) แต่สำหรับการนับฟังก์ชันพอยต์ใช้เพียงความต้องการซอฟต์แวร์ก็เป็นข้อมูลเบื้องต้นก็เพียงพอแล้ว (IFPUG,2005) และขั้นตอนกำหนดความต้องการเป็นขั้นตอนแรกที่เริ่มทำการประมาณขนาดซอฟต์แวร์

งานวิจัยนี้จะเน้นการวิจัยการนับฟังก์ชันพอยต์ในขั้นตอนกำหนดความต้องการซอฟต์แวร์ เพราะจะได้ข้อมูลตั้งแต่แรกเริ่มโครงการ ถึงแม้ในขั้นตอนนี้จะมีข้อมูลน้อยกว่าขั้นตอนการพัฒนาขั้นต่อๆไป แต่ก็เป็นข้อมูลที่เพียงพอต่อการนับฟังก์ชันพอยต์ ซึ่งฟังก์ชันพอยต์ที่ได้จะนำไปวางแผนพัฒนาโครงการในทันทีที่เริ่มโครงการ และการนับฟังก์ชันพอยต์สามารถปรับปรุงได้ในขั้นตอนการพัฒนาขั้นต่อๆไปได้อีก

จากความต้องการซอฟต์แวร์ที่ได้ก็จะได้ข้อมูลพื้นฐานในการนับฟังก์ชันพอยต์ ที่เป็นส่วนประกอบข้อมูล (Count the data functions) ส่วนประกอบทรานแซคชัน (Count the transactional functions) และส่วนประกอบการออกแบบหน้าเว็บหรือมัลติมีเดียไฟล์ (Multi-Media Files MMF) ซึ่งมัลติมีเดียไฟล์นี้รวมรูปภาพ, ภาพเคลื่อนไหว, ไฟล์เสียง และไฟล์วีดีโอ ที่ใช้ในการออกแบบหน้าเว็บไว้ด้วย

งานวิจัยนี้จึงได้เสนอมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบของเว็บ (Function Point with Web Component) ที่เป็นการเพิ่มเติมมัลติมีเดียไฟล์เข้าไปเป็นส่วนประกอบหนึ่งในการนับฟังก์ชันพอยต์เป็นหนึ่งในส่วนประกอบทรานเซคชัน คือ จะเป็นมัลติมีเดียไฟล์ที่แสดงออกมาบนหน้าเว็บ ซึ่งสามารถที่จะสามารถออกแบบได้ในขั้นตอนกำหนดความต้องการ และมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบของเว็บแสดงตามตารางที่ 3.1 ประกอบด้วย 5 ส่วนประกอบของฟังก์ชันพอยต์ และ 1 ส่วนประกอบจากมัลติมีเดียไฟล์

ตารางที่ 3.1 การนับค่าฟังก์ชันพอยต์ร่วมกับส่วนประกอบของเว็บ

คอมโพเน้น	ความซับซ้อน			หมายเหตุ
	ต่ำ	ปานกลาง	สูง	
แฟ้มข้อมูลภายใน (ILF)	7	10	15	การนับค่าฟังชันพอยต์นับตามข้อกำหนดของ International Function Point Users Group (IFPUG)
อินเตอร์เฟสแฟ้มข้อมูล (EIF)	5	7	10	
อินพุทภายนอก (EI)	3	4	6	
เอาท์พุทภายนอก (EO)	4	5	7	
การสอบถามภายนอก (EQ)	3	4	6	
มัลติมีเดียไฟล์ (MMF)	1	2	4	กำหนดความซับซ้อนจากชนิดของไฟล์มัลติมีเดีย คือ Multi-Media Files (graphics , animation , audio , video , etc.) -JPEG,GIF, PNG (ต่ำ) - Animation, A2b music, Microsoft Picture It (ปานกลาง) -AIFF Audio, Liquid Audio, Streaming Audio/Video (สูง)

3.2.1 การนับค่าฟังก์ชันพอยต์ร่วมกับส่วนประกอบของเว็บ

การนับค่าฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ การนับจะเป็นไปตามหลักของการนับค่าฟังก์ชันพอยต์โดยยึดตามหนังสือคู่มือฝึกนับฟังก์ชันพอยต์ (Function Point Counting Practices Manual[FPCPM]) จัดทำโดย IFPUG ซึ่งได้กล่าวในหัวข้อ 2.1.1 ปัจจุบันออกเป็นเวอร์ชัน 4.2.1 และเพิ่มเติมการนับของมัลติมีเดียไฟล์ตามตารางที่ 3.1 ซึ่งมีขั้นตอนการนับดังต่อไปนี้

1) กำหนดประเภทของการนับ

นับตามแนวทาง FPCPM

2) กำหนดขอบเขตการนับและขอบเขตแอพพลิเคชัน

นับตามแนวทาง FPCPM เพิ่มเติมโดยให้โปรแกรมเว็บที่พัฒนาและฐานข้อมูลที่

อยู่เครื่องแม่ข่าย (Server) เดียวกันหรืออยู่คนละเครื่อง ให้อยู่บนขอบเขตเดียวกัน (Reifer, 2002)

3) การนับส่วนประกอบข้อมูล

นับตามแนวทาง FPCPM

4) การนับส่วนประกอบทรานเซคชัน

นับตามแนวทาง FPCPM เพิ่มการนับมัลติมีเดียไฟล์ (MMF) โดยมัลติมีเดียไฟล์เป็นมัลติมีเดียไฟล์ที่แสดงออกบนหน้าโปรแกรมเว็บทที่กำลังทำการนับ และจะมีกฎการกำหนดมัลติมีเดียไฟล์ดังต่อไปนี้

(1) เป็นมัลติมีเดียไฟล์ที่แสดงบนหน้าเว็บจะมาจากไฟล์ที่อยู่ภายในขอบเขตแอพพลิเคชัน

และสามารถนักส่วนประกอบมัลติมีเดียไฟล์ได้ตามกฎต่อไปนี้

(1) นับมัลติมีเดียไฟล์ที่แสดงบนเว็บ 1 ไฟล์เป็น 1 หน่วย

(2) ไม่นับมัลติมีเดียไฟล์ที่มีลักษณะเป็นปุ่มเดียวที่ทำหน้าเป็นลิงค์ (Link or Navigator)

(3) นับมัลติมีเดียไฟล์ที่แสดงหน้าเว็บแบบวนซ้ำ (Repeat Field) ให้นับเป็น 1 หน่วย

(4) นับมัลติมีเดียไฟล์ที่เป็นภาพต่อให้นับเป็น 1 หน่วย และกำหนดปัจจัยความซับซ้อนได้ดังนี้

- จำนวนภาพต่อ 1-20 ภาพ ระดับต่ำ
- จำนวนภาพต่อ 20-40 ภาพ ระดับ
- จำนวนภาพต่อ มากกว่า 40 ภาพ ระดับสูง

(5) มัลติมีเดียไฟล์ที่นับแล้วจะไม่นับอีกเมื่อนำไปแสดงหน้าเว็บอื่น ในลักษณะโค้ดอินคลูด (Include Code)

(6) ไม่นับมัลติมีเดียไฟล์ลิงค์มาจากเว็บอื่น

5) การกำหนดปัจจัยความซับซ้อนของส่วนประกอบ

กำหนดตามแนวทาง FPCPM เพิ่มเติมการทำหนดความซับซ้อนของมัลติมีเดียไฟล์ (MMF) โดยมีกฎการกำหนดความซับซ้อนดังต่อไปนี้

(1) ระดับความซับซ้อนและค่าถ่วงน้ำหนักมีค่าตามตารางที่ 3.1

(2) มัลติมีเดียไฟล์สามารถกำหนดระดับความซับซ้อนได้ดังนี้ (Reifer, 2002)

- ระดับต่ำ ประกอบด้วย ไฟล์รูปภาพประเภท JPEG, GIF และ PNG

- ระดับปานกลาง ประกอบด้วย ภาพเคลื่อนไหว ไฟล์เสียงแบบ A2b และไฟล์ภาพที่ทำมา จาก Microsoft Picture It
- ระดับสูงประกอบด้วย ไฟล์เสียงแบบ AIFF และ Liquid และไฟล์เสียง และ ไฟล์วีดีโอแบบสตรีมมิ่ง (Streaming)

และถ้ามัลติมีเดียไฟล์ไม่สามารถเลือกระดับความซับซ้อนได้ให้เลือกกลุ่มไฟล์ที่มีลักษณะไกล้เคียง

6) การพิจารณาแฟคเตอร์ที่ปรับแต่งค่า

จะไม่มีการพิจารณาแฟคเตอร์ที่ปรับแต่งค่า เพราะจำนวนความพยายามและระยะเวลาพัฒนาที่จะคำนวณด้วยแบบจำลองโคโคโม2 จะไม่พิจารณาแฟคเตอร์ที่ปรับแต่งค่าของฟังก์ชันพอยต์ แต่จะใช้การพิจารณาแฟคเตอร์ที่ปรับแต่งค่า ของแบบจำลองโคโคโม2 เอง

7) การคำนวณฟังก์ชันพอยต์

คำนวนตามจากสมาการที่ 1-1 ในหัวข้อ 2.1.2

และเมื่อทำการรวมค่าฟังก์ชันพอยค์จากทุกส่วนประกอบ ก็จะได้ขนาดของโปรแกรมที่เป็นฟังก์ชันพอยต์ ซึ่งค่าที่ได้นี้จะนำไปประมาณขนาดของโครงการ หรือจัดทำมาตรวัดซอฟต์แวร์สำหรับโครงการด้วยแบบจำลองโคโคโม2 แต่ก่อนที่จะนำไปคำนวณในแบบจำลองโคโคโม2 จะต้องแปลงขนาดที่เป็นฟังก์ชันพอยต์ให้เป็นค่าจำนวนบรรทัดคำสั่งเสียก่อน ทำให้จะต้องมีการสร้างสมการประมาณจำนวนบรรทัดคำสั่งของโปรแกรมภาษาที่กำลังพัฒนา

3.3.2 การวิเคราะห์ความถดถอยและสหสัมพันธ์อย่าง่าย (Simple Regression Analysis and Correlation) (กัลยา วานิชย์บัญชา, 2544)

เป็นการศึกษาถึงความสัมพันธ์ระหว่างตัวแปร 2 ตัว หรือลักษณะที่สนใจศึกษา 2 ลักษณะ โดยที่ต้องทราบค่าของตัวแปรตัวหนึ่งหรือต้องกำหนดค่าของตัวแปรตัวหนึ่งไว้ล่วงหน้า เช่น ถ้าศึกษาถึงความสัมพันธ์ระหว่างรายจ่ายกับรายได้ ยอดขายกับค่าโฆษณา ฯลฯ ซึ่งต้องทราบหรือกำหนดรายได้ และค่าโฆษณาไว้ล่วงหน้า เช่น ทราบว่าเงินเดือนพนักงานทำความสะอาดของบริษัทแห่งหนึ่งเป็น 2,000, 2,500, 3,000, 3,500 และ 4,000 บาท ผู้วิเคราะห์จะต้องสอบถามพนักงานทำความสะอาดที่มีเงินเดือนดังกล่าวถึงรายจ่ายต่อเดือน จึงจะสามารถหาความสัมพันธ์ระหว่างรายได้กับรายจ่ายได้ หรือในการหาความสัมพันธ์ระหว่างยอดขายกับค่าโฆษณาจะต้องทราบถึงงบประมาณในการโฆษณาที่บริษัทกำหนดไว้หรือใช้ไปจริง แล้วจึงจะทราบถึงยอดขายโดยจะเรียก รายได้ และ ค่าโฆษณา ซึ่งเป็นตัวแปรที่ต้องกำหนดค่าไว้ล่วงหน้าว่า ตัวแปรอิสระ (Independent Variable) และมักจะใช้สัญลักษณ์ X ส่วนยอดขายกับรายจ่ายจะเรียกว่า ตัวแปรตาม

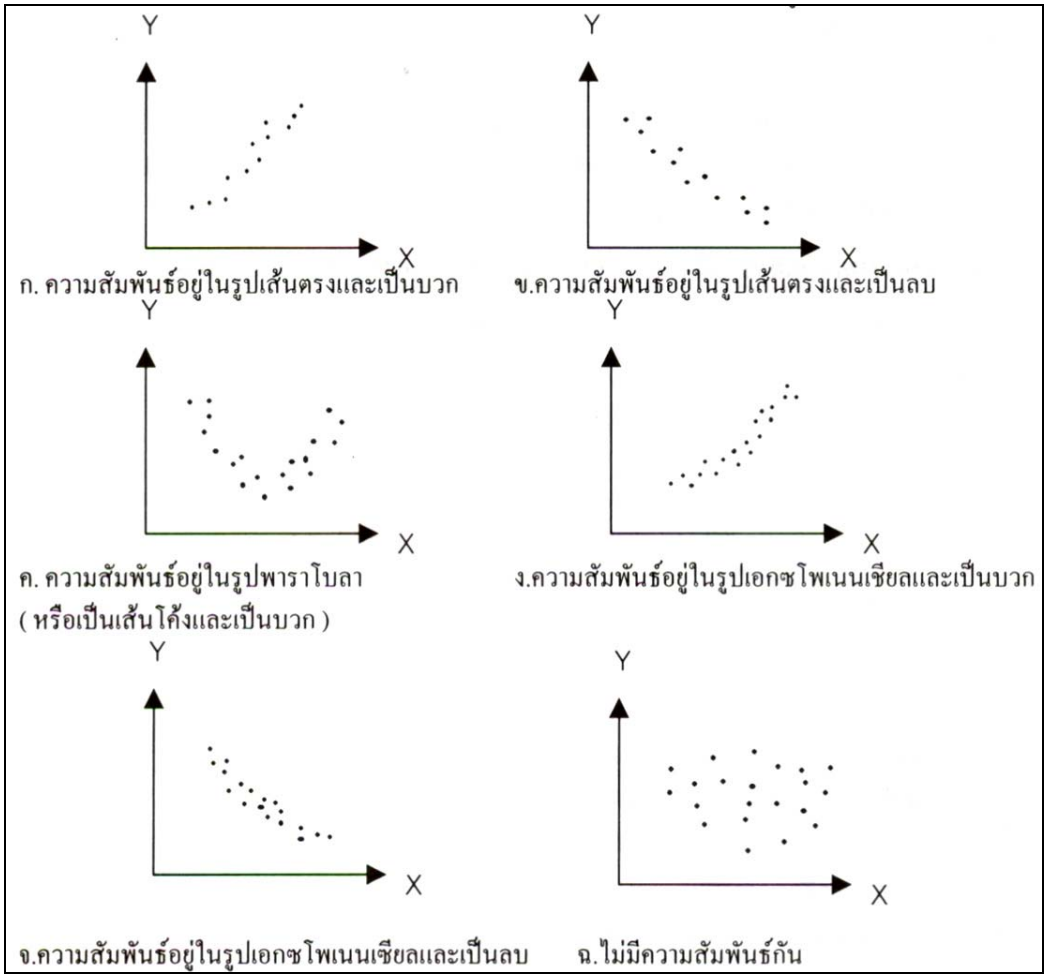
(Dependent Variable) และใช้ลัญลักษณ์ Y ซึ่งหมายถึง ยอดขายเป็นตัวแปรที่ขึ้นอยู่กับค่าโฆษณา และรายจ่ายเป็นตัวแปรที่ขึ้นกับรายได้

1) วัตถุประสงค์ของการวิเคราะห์ความถดถอยและสหสัมพันธ์

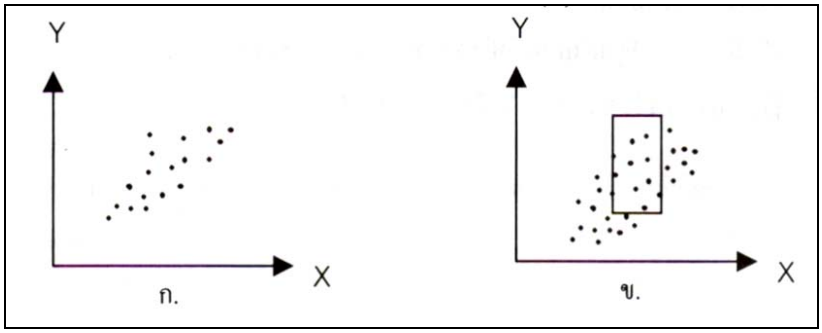
การวิเคราะห์ความสัมพันธ์ของตัวแปรต่างๆ มีวัตถุประสงค์ดังนี้

(1) เพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรว่ามีความสัมพันธ์กันมากน้อยเพียงใด ถ้า X และ Y มีความสัมพันธ์กันมากแสดงว่า ถ้า X มีค่าเปลี่ยนแปลงไปจะมีผลกระทบต่อค่าของ Y เป็นอย่างมาก

(2) ใช้ความสัมพันธ์ที่วิเคราะห์ได้มาประมาณค่าหรือพยากรณ์ค่า Y ในอนาคต เมื่อกำหนดค่า X นั้นในขั้นแรกจะนำเอาข้อมูลของตัวแปรทั้งสองมาเขียนกราฟแสดงความสัมพันธ์ซึ่งเรียกกราฟนี้ว่า แผนภาพการกระจาย (Scatter Diagram) ผู้วิเคราะห์จะต้องพิจารณาจากแผนภาพการกระจายว่าความสัมพันธ์ของตัวแปรทั้งสองจะอยู่ในรูปแบบใดเช่น เส้นตรง พาราโพลา เส้นโค้ง อื่นๆ ฯลฯ โดยที่จะต้องสามารถเขียนความสัมพันธ์ให้อยู่ในรูปแบบทางคณิตศาสตร์ได้ตัวอย่างแผนภาพการกระจายของตัวแปร X และ Y ได้แสดงไว้ในรูปที่ 3.1 จากรูปที่ 3.1 ก. และ ข. แสดงความสัมพันธ์ของ X และ Y ในลักษณะเชิงเส้นโดยที่รูป 3.1 ก. ความสัมพันธ์เป็นบวก ซึ่งหมายถึงถ้าค่า X เพิ่มขึ้น Y จะมีค่าเพิ่มขึ้นด้วย แต่ถ้า X ลดลงจะทำให้ Y ลดลงด้วย ส่วนรูปที่ 3.1 ข. แสดงถึงความสัมพันธ์ที่เป็นลบ หมายถึง X เพิ่มจะทำให้ Y มีค่าลดลง หรือถ้า X มีค่าลดลงจะมีผลให้ Y มีค่าเพิ่มขึ้น ตัวอย่างเช่นสนใจศึกษาความสัมพันธ์ระหว่างราคาขายบ้าน (Y) กับขนาดของพื้นที่บ้าน (X) ดังแสดงในรูปที่ 3.2 จากรูปที่ 3.2 ก แสดงถึงความสัมพันธ์ในรูปเส้นตรงที่เป็นบวก คือ ถ้าพื้นที่ของบ้านขนาดใหญ่ ราคาขายบ้านจะสูงด้วย แต่รูปที่ 3.2 ข ความสัมพันธ์เป็นบวกเช่นกัน แต่เนื่องจากมีบ้านที่มีขนาดเดียวกันแต่มีราคาต่างกัน ซึ่งอาจจะเกิดจากปัจจัยอื่นๆ ที่มีผลต่อราคาบ้าน เช่น สถานที่ จำนวนชั้น จำนวนห้องนอน จำนวนห้องน้ำเป็นต้น ในที่นี้จะศึกษาเฉพาะความสัมพันธ์ของตัวแปร X และ Y ในรูปเชิงเส้นหรือเส้นตรงเท่านั้น จึงเรียกการวิเคราะห์ความถดถอยอย่างง่ายที่ความสัมพันธ์ของตัวแปรอยู่ในรูปเชิงเส้นว่า การวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย (Simple Liner Regression Analysis)



รูปที่ 3.1 ตัวอย่างแผนภาพการกระจายซึ่งแสดงความสัมพันธ์ระหว่าง X และ Y
(กัลยา วานิชย์บัญชา, 2544)

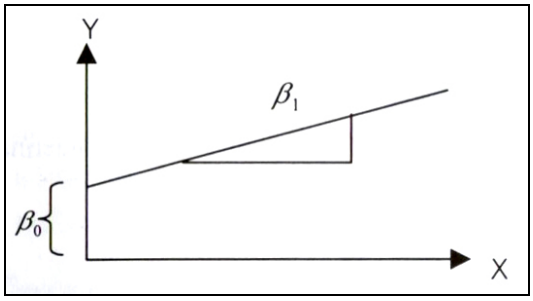


รูปที่ 3.2 ความสัมพันธ์ระหว่างราคาขายบ้าน (Y) กับขนาดของพื้นที่บ้าน (X) ก และ ข
(กัลยา วานิชย์บัญชา, 2544)

2) การวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย (Simple Liner Regression Analysis)

เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปร 2 ตัว ที่ความสัมพันธ์ในรูปเชิงเส้นซึ่งสามารถแสดงความสัมพัธ์ในรูปสมการเส้นและรูปที่ 3.3 ดังนี้

$$Y_i = \beta_0 + \beta_1 X_i + e_i \qquad ; i=1, 2,\ldots., N \qquad (3\text{-}1)$$



รูปที่ 3.3 สมการเชิงเส้น (กัลยา วานิชย์บัญชา, 2544)

โดยที่

Y คือ ตัวแปรตาม (Dependent Variable) เนื่องจากค่าของ Y อยู่กับค่าของ X

X = ตัวแปรอิสระ (Independent Variable)

β_0 = ส่วนตัดแกน Y หรือ คือค่าของ Y เมื่อ X มีค่าเป็นศูนย์

e = ความคลาดเคลื่อนอย่างสุ่ม

β_1 = ความชัน (slope) ของเส้นตรง ซึ่งเป็นค่าที่แสดงถึงอัตราการเปลี่ยนแปลง Y เมื่อ Xเปลี่ยนไป 1 หน่วย และเรียก β_1 ว่าสัมประสิทธ์ความถดถอย (Regression Coefficient) ค่าของ β_1 อาจจะเป็น

$\beta 1 > 0$ แสดงว่า X และ Y ความสัมพันธ์ในทางเดียวกันคือถ้า X เพิ่ม Y จะเพิ่มด้วย แต่ถ้า X ลดลง Y ลดลงด้วย

$\beta 1 < 0$ แสดงว่า X และ Y ความสัมพันธ์ในทางตรงข้ามคือถ้า X เพิ่มขึ้น Y ลดลง แต่ถ้า Xลดลง Y จะเพิ่มขึ้น

$\beta 1$ มีค่าเข้าใกล้ศูนย์ แสดงว่าค่า X และ Y มีความสัมพันธ์กันน้อย

$\beta 1 = 0$ แสดง X และ Y ไม่มีความสัมพันธ์กัน

3) สมมติฐานหรือเงื่อนไปของการวิเคราะห์ความถดถอย

(1) ค่า X จะต้องเป็นค่าที่กำหนดไว้ล่วงหน้าหรือทราบค่า

(2) ความคลาดเคลื่อน e_i เป็นตัวแปรที่มีค่าเฉลี่ย = 0 หรือ $E(e_i) = 0$ ค่าแปรปรวนของ e_i มีค่าเท่ากันทุกค่าของ i และมีค่าเท่ากับค่าแปรปรวนของ Y นั่นคือ $V(e_i) = V(Y) = \sigma^2_{y.x} = \sigma^2$

(3) e_i และ e_j เป็นอิสระกัน นั่นคือ $Cov(e_i , e_j) = E(e_i , e_j) = 0 ; i \neq j$

(4) e_i มีการแจกแจงแบบปกติที่มีค่าเฉลี่ยเป็นศูยน์และค่าแปรปรวน σ^2 นั่นคือ

$e_i \sim normal (0 , \sigma^2)$

จากข้อสมมุติข้างต้น จะได้ว่า

$Y_i \sim normal (E(Y_i) , \sigma^2)$

โดยที่

$$E(Y_i) = E(\beta_0 + \beta_1 X_i + e_i)$$
$$= \beta_0 + \beta_1 X_i + E(e_i)$$
$$= \beta_0 + \beta_1 X_i \text{ เนื่องจาก } E(e_i) = 0$$

4) การประมาณค่าพารามิเตอร์ของสมการความถดถอย

เมื่อพิจารณาจากแผนภาพการกระจาย ซึ่งแสดงความสัมพันธ์ระหว่าง X และ Y แล้วพบว่า X และ Y สัมพันธ์กันในรูปเส้นตรง จะต้องคำนวณหาค่า β_0และ β_1 ซึ่งจะทำให้ทราบถึงความสัมพันธ์ระหว่าง X และ Y ว่ามีความสัมพันธ์ตามกันหรือตรงข้ามกันและความสัมพันธ์นั้นมากหรือน้อยเพียงใด ถ้า β_1 มีค่ามากแสดงว่า Y มีความสัมพันธ์กับ X มากด้วย

การที่จะหาค่า β_0และ β_1 ได้จำเป็นต้องทราบค่า X และ Y ทุกค่าที่ได้เกิดขึ้นในอดีตเช่น ถ้า X = รายได้ของคนกรุงเทพมหานคร (กทม.) Y = รายจ่ายของคนกทม. การหาค่า β_0 และ β_1 จะต้องทราบถึงรายได้และรายจ่ายของคนกทม ทุกคน ซึ่งเป็นไปได้ยากในทางปฏิบัติเราจึงใช้ข้อมูลตัวอย่างขนาด n ในการประมาณค่า β_0และ β_1 ดังนั้นค่าประมาณของ Y

$$Y_i = \beta_0 + \beta_1 X_i \text{ หรือ } \hat{y} = a + bX_i \qquad (3\text{-}2)$$

โดยที่

$\hat{\beta}_0 = a, \hat{\beta}_1 = b$

จากนั้นก็จะทำการประมาณค่า β_0 และ β_1 โดยวิธีกำลังสองน้อยที่สุด จะได้ค่า β_0 และ β_1 ดังต่อไปนี้

$$\beta_1 = b = \frac{\sum X_i Y_i - \frac{(\sum X_i)(\sum Y_i)}{n}}{\sum X_i^2 - \frac{(\sum X_i)^2}{n}} \quad (3\text{-}3)$$

$$\beta_0 = a = \bar{y} - b\bar{x} \quad (3\text{-}4)$$

หรือ

$$b = \frac{SS_{XY}}{SS_{XX}} \text{ และ } a = \bar{y} - b\bar{x} \quad (3\text{-}5)$$

โดยที่

$$SS_{XX} = \sum (X_i - \bar{x})^2 = \sum_1^n X_i^2 - \frac{(\sum_1^n X_i)^2}{n} \quad (3\text{-}5\text{-}1)$$

$$SS_{XY} = \sum (X_i - \bar{x})(Y_i - \bar{y}) = \sum_1^n X_i Y_i - \frac{(\sum_1^n X_i)(\sum_1^n Y_i)}{n} = \sum (X_i - \bar{x})Y_i \quad (3\text{-}5\text{-}2)$$

$$SS_{YY} = \sum (Y_i - \bar{y})^2 = \sum_1^n Y_i^2 - \frac{(\sum_1^n Y_i)^2}{n} \quad (3\text{-}5\text{-}3)$$

การที่ประมาณค่า β_0 และ β_1 ด้วยค่า a และ b โดยใช้วิธีกำลังสองน้อยที่สุดจะทำให้

(1) ผลรวมของค่าคลาดเคลื่อนในการประมาณค่า Y_i ด้วย $\overline{Y}_i$ เป็นศูนย์ คือ $SS_{YY} = \sum (Y_i - \bar{y})^2 = \sum e_i = 0$

(2) จุดที่ $(\bar{x}, \bar{y})$ เป็นจุดที่อยู่บนเส้นความถดถอย

(3) $\sum (Y_i - \bar{y})^2$ มีค่าต่ำที่สุด

5) สัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient)

จากหัวข้อ 1 ถึง 3 เป็นการวิเคราะห์เพื่อหาความสัมพันธ์ของตัวแปร X และ Y และเป็นการทดสอบว่า X และ Y มีความสัมพันธ์กันหรือไม่ โดยการทดสอบสัมประสิทธิ์ความถดถอยค่า (β_1) โดยใช้ค่าประมาณ (b) ซึ่งไม่สามารถระบุได้ว่า X และ Y มีความสัมพันธ์กันมากหรือน้อย เนื่องจากค่า b ที่ได้มีหน่วยตามค่าของ Y เช่น Y หน่วยเป็นบาท ค่า b จะมีหน่วยเป็นบาทด้วย แต่ถ้าค่า Y มีหน่วยเป็นล้านบาท ค่า b จะมีหน่วยเป็นล้านบาทด้วย การที่ค่า b ที่มีหน่วยเป็นบาทมีค่ามากกว่าค่า b ที่มีหน่วยเป็นล้านบาท ไม่ได้หมายความว่าความสัมพันธ์ X และ Y ที่ Y มีหน่วยเป็นบาทจะมีความสัมพันธ์กันมากกว่ากรณีที่ Y มีหน่วยเป็นล้านบาท นั้นคือการที่กำหนดให้หน่วยของY แตกต่างกันจะทำให้ค่า b แตกต่างกันด้วย (X และ Y ไม่จำเป็นต้องมีหน่วยเหมือนกัน)

สำหรับสถิติที่ใช้วัดความสัมพันธ์ระหว่าง X และ Y ว่ามากหรือน้อยนั้นจะเรียกว่า สัมประสิทธิ์สหสัมพันธ์ (ρ) ซึ่งในกรณีที่ค่าของ Y ขึ้นกับ X เพียงตัวเดียวจะเรียกว่าสัมประสิทธิ์สหสัมพันธ์อย่างง่าย (Simple Correlation Coefficient) โดยที่ ρ จะไม่มีหน่วย จึงสามารถใช้วัดความสัมพันธ์ระหว่าง X และ Y ได้ว่ามีความสัมพันธ์มากหรือน้อยเพียงใด เนื่องจากค่า ρ จะมีค่าสูงสุดเป็น 1 และต่ำสุดเป็น -1

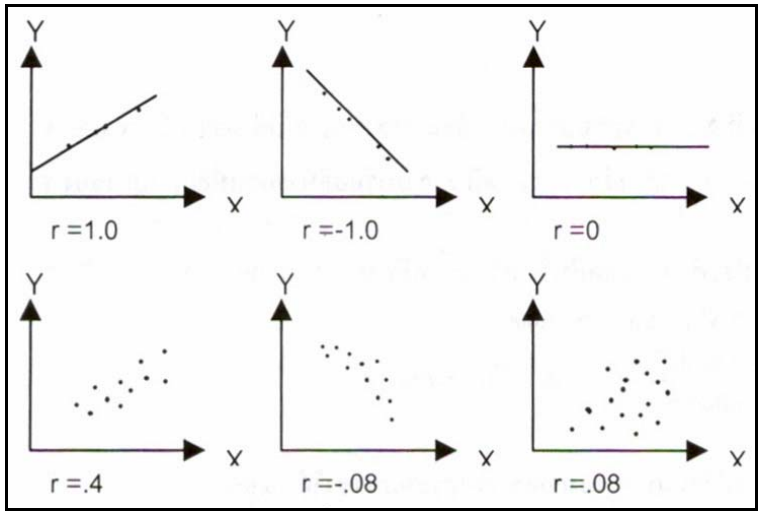
นอกจากนั้นความถดถอยและสหสัมพันธ์ยังมีวิธีการในการเก็บข้อมูลที่แตกต่างกัน คือสำหรับเรื่องความถดถอยตัวแปรอิสระ X จะต้องถูกกำหนดไว้ล่วงหน้า ในขณะที่เป็นตัวแปรตามหรือ Y เป็นตัวแปรสุ่ม แต่สหสัมพันธ์ทั้ง X และ Y จะเป็นตัวแปรสุ่มทั้งคู่

เนื่องจากเราใช้ข้อมูลตัวอย่าง จึงประมาณค่า ρ ด้วยค่า r โดยที่ r คือสัมประสิทธิ์สหสัมพันธ์ตัวอย่าง

ความหมายของค่า r มีความหมาบดังต่อไปนี้และตามรูปที่ 3.4

- ค่า r เป็นลบ แสดงว่า X และ Y มีความสัมพันธ์ในทิศทางตรงข้าม คือถ้า X เพิ่ม Y จะลด แต่ถ้า X ลด Y จะเพิ่ม
- ค่า r เป็นบวกแสดงว่า X และ Y มีความสัมพันธ์ในทางเดียวกัน คือ ถ้า X เพิ่ม Y จะเพิ่มด้วยแต่ถ้า X ลด Y จะลดด้วย
- ถ้า r มีค่าเข้าใกล้ 1 หมายถึง X และ Y สัมพันธ์ในทิศทางเดียวกันและมีความสัมพันธ์กันมาก
- ถ้า r มีค่าเข้าใกล้ -1 หมายถึง X และ Y สัมพันธ์ในทิศทางตรงกันข้ามและมีความสัมพันธ์กันมาก
- ถ้า r = 0 แสดงว่า X และ Y ไม่มีความสัมพันธ์กัน
- ถ้า r เข้าใกล้ 0 แสดงว่า X และ Y มีความสัมพันธ์กันน้อย

ในทางปฏิบัติการที่จะพิจารณาว่าตัวแปร X และ Y มีความสัมพันธ์กันมากหรือน้อย จะพิจารณาจากค่า r (ไม่ใช่จากค่า b) เนื่องจาก r ไม่มีหน่วย และมีขอบเขตคือมีค่าต่ำสุด = -1 ค่าสูงสุด = 1 นอกจากนั้น r และ b จะมีเครื่องหมายเดียวกัน คือ เป็นบวกเหมือนกันหรือเป็นลบเหมือนกัน เนื่องจากทั้ง r และ b เป็นค่าแสดงถึงความสัมพันธ์ระหว่าง X และ Y



รูปที่ 3.4 การแสดงค่าของ r ที่มีค่า -1 < r < 1 (กัลยา วานิชย์บัญชา, 2544)

ความสัมพันธ์ระหว่างสัมประสิทธ์การตัดสินใจ และ สัมประสิทธิ์สหสัมพันธ์ สำหรับเมื่อมี 2 ตัวแปร สัมประสิทธิ์สหสัมพันธ์จะเท่ากับรากที่สองของสัมประสิทธ์การตัดสินใจ หรือ $r = \sqrt{r^2}$

โดยสัมประสิทธิ์การตัดสินใจ หมายถึงสัดส่วนที่ตัวแปร X สามารถอธิบายการเปลี่ยนแปลงของ Y ได้ ดังนั้นถ้า r^2 มีค่ามากแสดงว่า Y และ X มีความสัมพันธ์กันมาหรือ X สามารถอธิบายการเปลี่นยแปลงของค่า Y ได้มาโดยที่

$$\text{สัมประสิทธ์การตัดสินใจ} = R = r^2 = b\frac{SS_{XY}}{SS_{YY}} \qquad (3\text{-}6)$$

3.3 การเก็บรวบรวมและการวิเคราะห์โปรแกรมต้นแบบ

ในการวิจัยนี้ ได้เก็บรวมรวมข้อมูลด้วยโปรแกรมเว็บที่พัฒนาจากด้วยโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอลจากแหล่งเปิดเผยซอร์สโค้ด และ หนังสือตำรา รวม 10 โปรแกรม สร้างสมการประมาณขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่ง เพื่อใช้เป็นสมการมาตรฐาน และนำสมการมาตรฐานใช้ในการประมาณขนาดของโปรแกรมเว็บที่จะพัฒนาขึ้นใหม่ต่อไป โดยจะกล่าวถึง การเก็บรวบรวมและนับฟังก์ชันพอยต์ และ การสร้างสมการประมาณขนาดด้วยวิธีการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย ในหัวข้อ 3.3.1 และ 3.3.2

3.3.1 การเก็บรวบรวมโปรแกรมเว็บและนับฟังก์ชันพอยต์

ข้อมูลของโปรแกรมต้นแบบที่เก็บรวบรวมมา จะทำการจดบันทึกข้อมูลการนับฟังก์ชันพอยต์ ซึ่งรายการที่จะเก็บข้อมูลจะประกอบด้วย ชื่อโปรแกรม แหล่งข้อมูล ประเภทของเว็บ จำนวนฟังก์ชันพอยต์ แสดงในตารางที่ 3.2 ถึง 3.11 โดยมีรายละเอียดสรุปดังนี้

1) โปรแกรมตู้เพลง

(1) แหล่งข้อมูลจาก www.deeserver.net, www.mocye.com

(2) ประเภทเว็บข้อมูลสารสนเทศ

ตารางที่ 3.2 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมตู้เพลง

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	2	x3	-	x4	-	x6	6
เอาท์พุทภายนอก(EO)	-	x4	-	x5	-	x7	-
การสอบถามภายนอก(EQ)	2	x3	-	x4	-	x6	6
แฟ้มข้อมูลภายใน(ILF)	1	x7	-	x10	-	x15	7
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	2	x1	-	x2	1	x4	6
รวมฟังก์ชันพอยต์(UPF)							25

2) โปรแกรมร้านหนังสือออนไลน์ 1

(1) แหล่งข้อมูลจากหนังสือ Sams Teach Yourself PHP4 in 24 Hours

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์

ตารางที่ 3.3 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านหนังสือออนไลน์ 1

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	3	x3	1	x4	1	x6	19
เอาท์พุทภายนอก(EO)	-	x4	-	x5	-	x7	-
การสอบถามภายนอก(EQ)	5	x3	-	x4	-	x6	15
แฟ้มข้อมูลภายใน(ILF)	4	x7	-	x10	-	x15	28
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	4	x1	-	x2	-	x4	4
รวมฟังก์ชันพอยต์(UPF)							66

3) โปรแกรมจัดการภาพ

(1) แหล่งข้อมูลจากเว็บไซต์ http://www.3dsrc.com

(2) ประเภทเว็บข้อมูลสารสนเทศ

ตารางที่ 3.4 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมจัดการภาพ

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	16	x3	-	x4	-	x6	48
เอาท์พุทภายนอก(EO)	-	x4	-	x5	-	x7	-
การสอบถามภายนอก(EQ)	8	x3	-	x4	1	x6	30
แฟ้มข้อมูลภายใน(ILF)	4	x7	-	x10	-	x15	28
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	8	x1	-	x2	-	x4	8
รวมฟังก์ชันพอยต์(UPF)							114

4) โปรแกรมร้านสินค้าออนไลน์ 1

(1) แหล่งข้อมูลจากเว็บไซต์ http://www.professionals.co.th

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์

ตารางที่ 3.5 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 1

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	7	x3	-	x4	-	x6	21
เอาท์พุทภายนอก(EO)	2	x4	-	x5	-	x7	8
การสอบถามภายนอก(EQ)	11	x3	-	x4	-	x6	33
แฟ้มข้อมูลภายใน(ILF)	4	x7	-	x10	-	x15	28
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	12	x1	-	x2	-	x4	12
รวมฟังก์ชันพอยต์(UPF)							102

5) โปรแกรมร้านสินค้าออนไลน์ 2

(1) แหล่งข้อมูลจากเว็บไซต์ http://www.zippydesign.com

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์

ตารางที่ 3.6 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 2

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	12	x3	3	x4	-	x6	48
เอาท์พุทภายนอก(EO)	-	x4	-	x5	-	x7	-
การสอบถามภายนอก(EQ)	4	x3	5	x4	-	x6	32
แฟ้มข้อมูลภายใน(ILF)	4	x7	-	x10	-	x15	28
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	1	x1	-	x2	-	x4	1
รวมฟังก์ชันพอยต์(UPF)							109

6) โปรแกรมร้านสินค้าออนไลน์ 3

(1) แหล่งข้อมูลจากเว็บไซต์ http://www.ebizzi.net

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์และเว็บข้อมูลสารสนเทศ

ตารางที่ 3.7 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 3

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	15	x3	-	x4	-	x6	45
เอาท์พุทภายนอก(EO)	-	x4	-	x5	1	x7	7
การสอบถามภายนอก(EQ)	11	x3	1	x4	-	x6	37
แฟ้มข้อมูลภายใน(ILF)	4	x7	-	x10	-	x15	28
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	4	x1	-	x2	-	x4	4
รวมฟังก์ชันพอยต์(UPF)							121

7) โปรแกรมร้านหนังสือออนไลน์ 2

(1) แหล่งข้อมูลจากหนังสือ คัมภีร์ พีเอสพี โดย กิตติ ภักดีวัฒนะกุล

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์และเว็บข้อมูลสารสนเทศ

ตารางที่ 3.8 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านหนังสือออนไลน์ 2

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	13	x3	6	x4	1	x6	69
เอาท์พุทภายนอก(EO)	-	x4	-	x5	-	x7	-
การสอบถามภายนอก(EQ)	10	x3	-	x4	-	x6	30
แฟ้มข้อมูลภายใน(ILF)	9	x7	-	x10	-	x15	63
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	4	x1	-	x2	-	x4	5
รวมฟังก์ชันพอยต์(UPF)							166

8) โปรแกรมระบบจองห้องพัก

(1) แหล่งข้อมูลเว็บไซต์ http://www.mejai.com

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์และเว็บข้อมูลสารสนเทศ

ตารางที่ 3.9 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมระบบจองห้องพัก

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	16	x3	2	x4	2	x6	68
เอาท์พุทภายนอก(EO)	-	x4	1	x5	4	x7	33
การสอบถามภายนอก(EQ)	13	x3	2	x4	-	x6	47
แฟ้มข้อมูลภายใน(ILF)	10	x7	-	x10	-	x15	70
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	12	x1	-	x2	-	x4	19
รวมฟังก์ชันพอยต์(UPF)							237

9) โปรแกรมระบบตำแหน่งงานออนไลน์

(1) แหล่งข้อมูลจากเว็บไซต์ http://www.narkDeveloper.com

(2) ประเภทเว็บข้อมูลสารสนเทศ

ตารางที่ 3.10 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมระบบตำแหน่งงานออนไลน์

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	20	x3	3	x4	-	x6	72
เอาท์พุทภายนอก(EO)	-	x4	-	x5	1	x7	7
การสอบถามภายนอก(EQ)	40	x3	1	x4	11	x6	190
แฟ้มข้อมูลภายใน(ILF)	10	x7	1	x10	-	x15	80
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	18	x1	-	x2	-	x4	18
รวมฟังก์ชันพอยต์(UPF)							367

10) โปรแกรมร้านสินค้าออนไลน์ 4

(1) แหล่งข้อมูลจากเว็บไซต์ http://www.phpshop.org

(2) ประเภทเว็บพาณิชย์อิเลกทรอนิคส์

ตารางที่ 3.11 การนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมร้านสินค้าออนไลน์ 4

ส่วนประกอบซอฟต์แวร์	ความซับซ้อน						จำนวนฟังก์ชันพอยต์(UPF)
	ต่ำ		ปานกลาง		สูง		
อินพุทภายนอก(EI)	43	x3	8	x4	12	x6	233
เอาท์พุทภายนอก(EO)	1	x4	2	x5	-	x7	14
การสอบถามภายนอก(EQ)	58	x3	12	x4	6	x6	258
แฟ้มข้อมูลภายใน(ILF)	20	x7	3	x10	-	x15	170
อินเตอร์เฟสแฟ้มข้อมูล(EIF)	-	x5	-	x7	-	x10	-
มัลติมีเดียไฟล์(MMF)	6	x1	-	x2	-	x4	6
รวมฟังก์ชันพอยต์(UPF)							681

3.3.2 การสร้างสมการประมาณขนาดด้วยวิธีการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย

จากหัวข้อ 3.3.1 จะทำการนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ ต่อจากนั้นก็จะเป็นการสร้างสมการประมาณขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่ง ด้วยวิธีการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย (Simple Regression Analysis)

ซึ่งจะใช้วิธีการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่ายจากหัวข้อ 3.2.2 ใช้จำนวนบรรทัดคำสั่งที่นับจากโปรแกรม Code Count Pro ตามรูปที่ 3.5 และจำนวนฟังก์ชันพอยต์ จะได้สมการตามการคำนวณและข้อมูลในตารางที่ 3.12 ดังต่อไปนี้

$$SS_{XX} = \sum_{1}^{n} X_i^2 - \frac{(\sum_{1}^{n} X_i)^2}{n} = 737087 - \frac{(1988)^2}{10} = 341863.6$$

$$SS_{XY} = \sum_{1}^{n} X_i Y_i - \frac{(\sum_{1}^{n} X_i)(\sum_{1}^{n} Y_i)}{n} = 16989078 - \frac{(1988)(45097)}{10} = 8023794.4$$

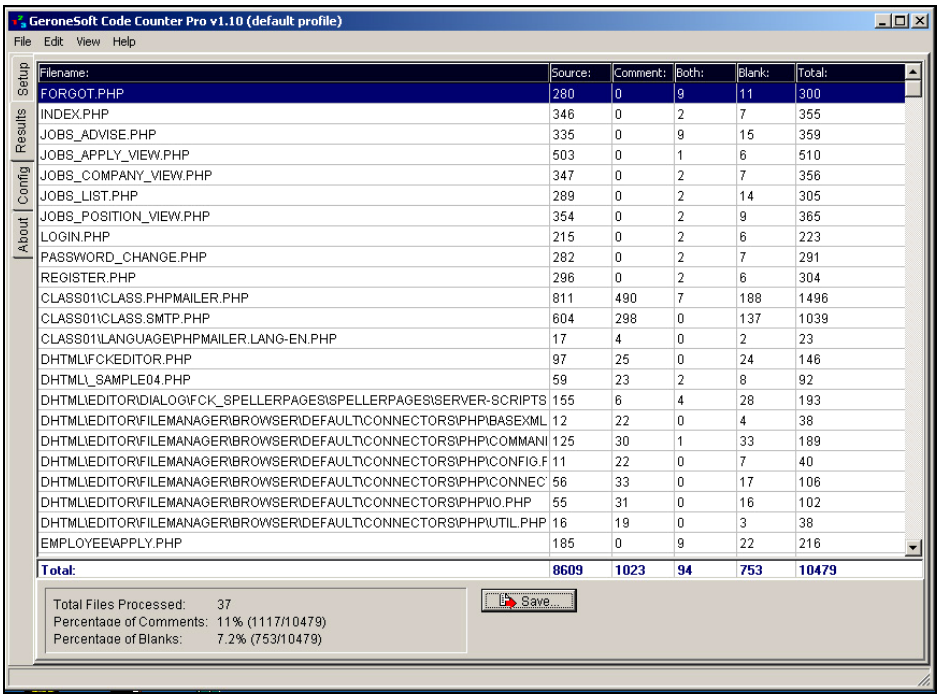
$$b = \frac{SS_{XY}}{SS_{XX}} = \frac{8023794.4}{341872.6} = 23.47$$

$$a = \bar{y} - b\bar{x} = \frac{45097}{10} - 23.47(\frac{1988}{10}) = -156.28$$

จะได้สมการประมาณขนาดซอฟต์แวร์คือ **SLOC = 23.47(UFP)-156.28** (3-7)

ตารางที่ 3.12 ตารางการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย

App.	UFP X_i	SLOC Y_i	X_i^2	X_iY_i
1	25	511	625	12775
2	66	1349	4356	89034
3	114	2204	12996	251256
4	102	2276	10404	232152
5	109	2488	14641	302016
6	121	2496	27556	609884
7	166	3674	51529	1051010
8	237	5796	56169	1373652
9	367	8609	134689	3159503
10	681	15694	463761	10687614
	$\sum X = 1988$	$\sum Y = 45097$	$\sum X^2 = 737087$	$\sum XY = 16989078$



รูปที่ 3.5 โปรแกรม Count Code Pro

3.4 โปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ

ในการวิจัยครั้งนี้เป็นการการพัฒนามาตรวัดแบบร่วมกับส่วนประกอบเว็บ สำหรับโปรแกรมภาษาพีเอชพีร่วมกับฐานข้อมูลมายเอสคิวแอล (PHP Language with Mysql Database) ซึ่งในการพัฒนาโปรแกรมนั้นย่อมจะต้องมีส่วนประกอบเว็บและซอร์สโค้ดจำนวนมาก ทำให้การที่นับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บเป็นไปด้วยความยากลำบาก ทำให้งานวินัยนี้ได้พัฒนาโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บขึ้นเพื่อช่วยนับฟังก์ชันพอยต์ คำนวณค่าความซับซ้อนของแต่ละส่วนประกอบ และบันทึกลงฐานข้อมูลได้ โดยพัฒนามาจากโปรแกรมภาษาวิชวลเบสิก เวอร์ชัน 6 (Microsoft Visual Basic 6) มีส่วนประกอบและการทำงานดังต่อไปนี้

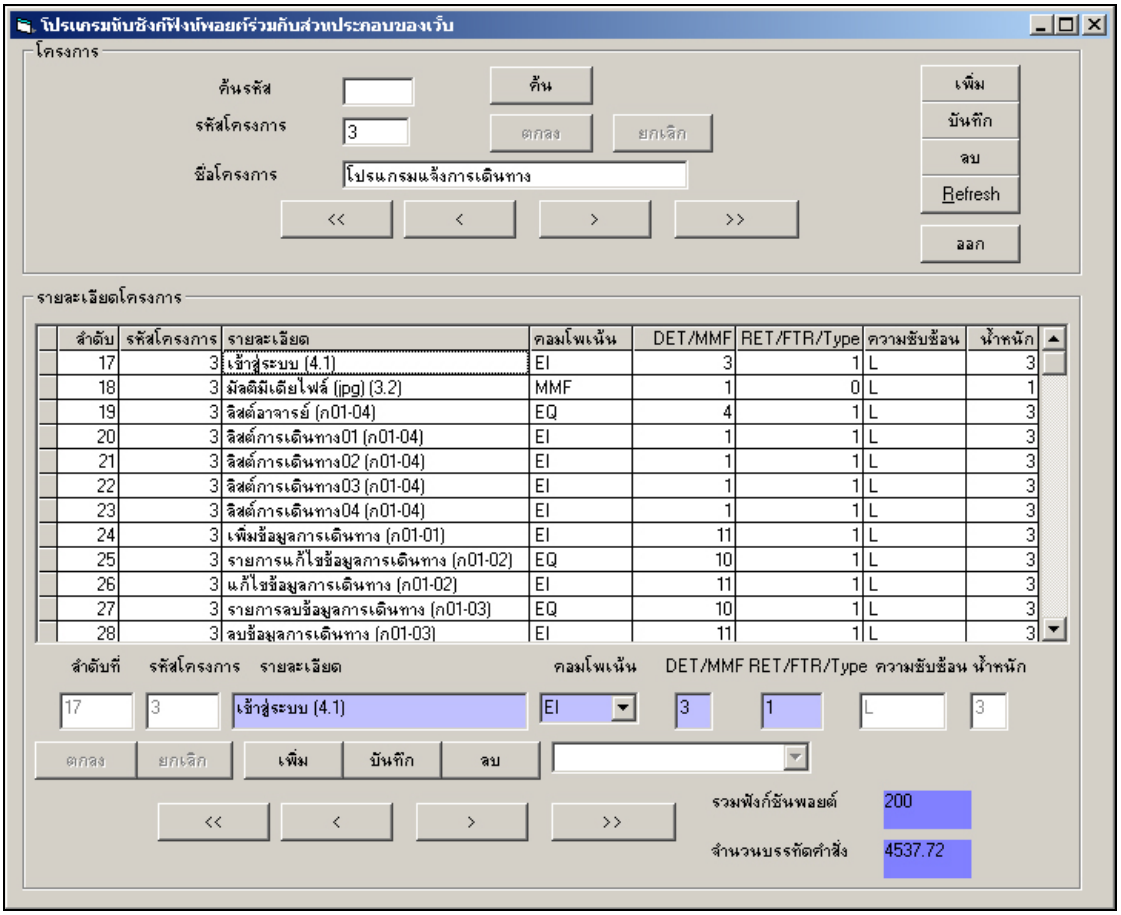
3.4.1 ฐานข้อมูล

1) ตารางโครงการ ประกอบด้วยเก็บรหัสโครงการ (p_id) และชื่อโครงการ (p_name)
2) ตารางฟังกชัยพอยต์ ประกอบด้วยรหัสส่วนประกอบ(d_id) รหัสโครงการ (id) รายละเอียดส่วนประกอบ (Description) ชนิดของคอมโพเน้น (Type) เขตข้อมูล (DET) แฟ้มข้อมูล (FTR)/เรคคอร์ดข้อมูล (RET) ความซับซ้อน (Complexity) และ ค่าถ่วงน้ำหนัก (Weight) รายละเอียดตามรูปที่ 3.7

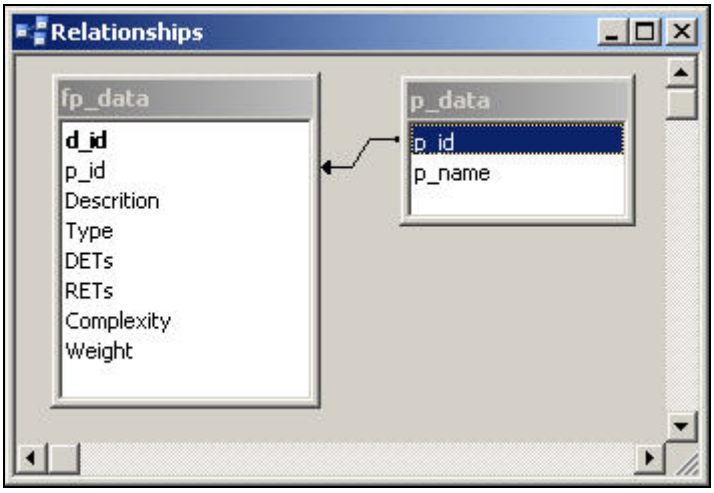
3.4.2 รูปแบบและการทำงานของโปแกรม

1) รูปแบบของโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บมีส่วนประกอบแสดงตามรูปที่3.6

2) การทำงานโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บเป็นโปรแกรมที่จะจัดเก็บ รายละเอียดส่วนประกอบฟังก์ชันพอยต์ (Component) เป็นรายโครงการโดยแต่ละโครงการสามารถ เพิ่ม ลบ และแก้ไขได้ ส่วนประกอบได้และโปรแกรมยังจะคำนวณค่าถ่วงน้ำหนักให้เมื่อทำงานป้อนข้อมูลเขตข้อมูล (DET) แฟ้มข้อมูล (FTR)/เรคคอร์ดข้อมูล (RET) สำหรับ อินพุทภายนอก เอาท์พุทภายนอก การสอบถามภายนอก แฟ้มข้อมูลภายใน และ อินเตอร์เฟสแฟ้มข้อมูล และ จำนวนมัลติมีเดียไฟล์ ชนิดมัลติมีเดียไฟล์ สำหรับ มัลติมีเดียไฟล์ (MMF)



รูปที่ 3.6 โปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ

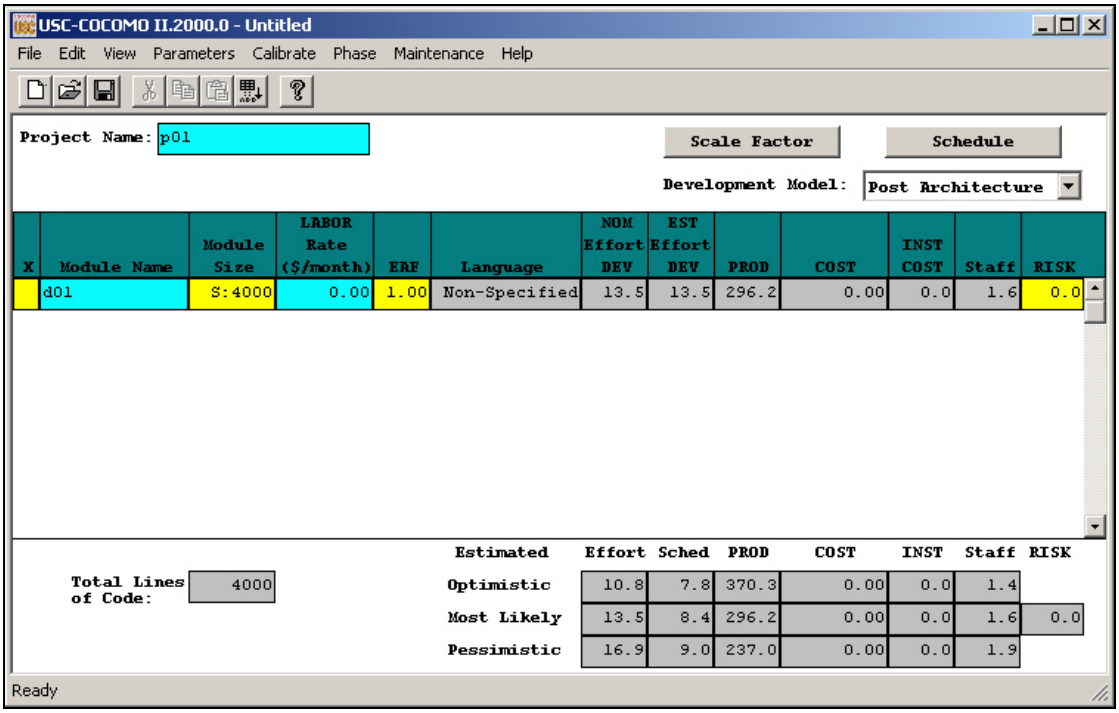


รูปที่ 3.7 ฐานข้อมูลโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ

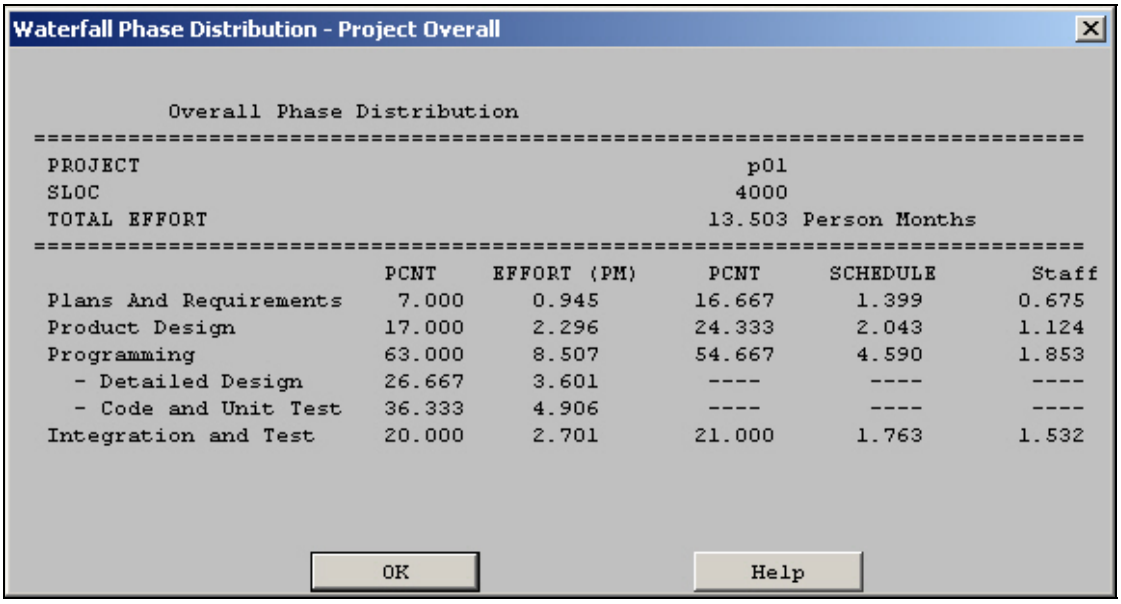
3.5 โปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000

มาตรวัด (Measurement Tool) ที่จะนำใช้ในงานวิจัยจะใช้โปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 ที่พัฒนาโดย University of Southern California เป็นโปรแกรมที่ใช้หลักการของแบบจำลอง โคโคโม2 มีรูปแบบเป็นส่วนต่อประสานกราฟิกส์ (GUI) สามารถคำนวณในแบบจำลองเออรีดิไซด์ และแบบจำลองโพสอาร์คิเท็กเจอร์ โปรแกรมนี้จะรับค่าจำนวนบรรทัดคำสั่ง และค่าตัวคูณความพยายาม(Effort Multipliers) ค่าสเกลของปัจจัย(Scale Factors) และค่าอื่นที่จำเป็น แล้วโปรแกรมจะทำการคำนวณจำนวนความพยายาม และระยะเวลาพัฒนา ออกมา นอกจากนั้นยังจะคำนวณขั้นตอนการพัฒนาโปรแกรมแบบน้ำตก (waterfall) ออกมาได้ รูปแบบของโปรแกรมแสดงในรูปที่ 3.8 และขั้นตอนการพัฒนาโปรแกรน้ำตก (waterfall) แสดงในรูปที่ 3.9 รายละเอียดของโปรแกรมแสดงในภาคผนวก ค.

โดยจำนวนความพยายาม (Effort) ระยะเวลาพัฒนา (Duration) และขั้นตอนการพัฒนาโปรแกรม ที่ได้จากโปรแกรมนี้จะมาคำนวณออกจะนำมาเป็นมาตรวัดการพัฒนาโปรแกรมหรือโครงการ นอกจากนี้โปรแกรมนี้ยังใช้กับการพัฒนาโปรแกรมแบบปรับปรุงและนำกลับมาใช้ (Adaptation and Reuse) ได้ด้วย



รูปที่ 3.8 โปรแกรมโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000



รูปที่ 3.9 ขั้นตอนการพัฒนาโปรแกรมแบบน้ำตก (waterfall)

บทที่ 4
ผลการวิเคราะห์ข้อมูลและอภิปรายผล

4.1 การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บด้วยวิธี Cross-Validation

การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บจะใช้วิธี Cross-Validation ที่ใช้หลักการแยกข้อมูลฝึก (Train Data) แยกชุดข้อมูลออกเป็น ชุดข้อมูลฝึก (Train Set) และ ชุดข้อมูลทดสอบจาก (Test Set) แล้วทำการทดสอบชุดข้อมูลฝึกด้วยชุดข้อมูลทดสอบและบันทึกผลในการทดสอบ จากนั้นก็เลือกชุดข้อมูลฝึกและชุดข้อมูลทดสอบชุดใหม่ ตามหลักการจัดหมู่ (Combination) โดยอัตราส่วนของชุดข้อมูลฝึกและชุดข้อมูลทดสอบจะขึ้นอยู่กับจำนวนรอบในการทดสอบ ซึ่งในการทดสอบในงานวิจัยนี้มีข้อมูลทดสอบ 10 ชุดข้อมูล จำนวนรอบในการทดสอบ 10 รอบ จะได้อัตราส่วนของชุดข้อมูลฝึกและชุดข้อมูลทดสอบ เป็น 9 ต่อ 1 และเมื่อทำการทดสอบด้วยวิธี Cross-Validation เสร็จก็จะได้ค่าต่างๆ ตามตารางที่ 4.1

ตารางที่ 4.1 ตารางความคลาดเคลื่อนสัมบูรณ์และความคลาดเคลื่อนกำลังสอง

รอบการทดสอบ	จำนวน บรรทัดจริง	จำนวนบรรทัดที่ ประมาณได้	ค่าความคลาด เคลื่อนสัมบูรณ์	ค่าความคลาด เคลื่อนกำลังสอง
1	511	411.799	99.201	9840.8384
2	1349	1400.608	51.608	2663.3857
3	2204	2562.809	358.809	128743.8985
4	2276	2232.144	43.856	1923.3487
5	2488	2389.903	98.097	9623.0214
6	2496	2708.713	212.713	45246.8204
7	3674	3747.434	73.434	5392.5526
8	5796	5360.917	435.083	189297.2169
9	8609	8423.596	185.404	34374.6432
10	15694	16300.285	606.285	367581.5012

และจากตารางที่ 4.1 สามารถคำนวณหาค่าประสิทธิทางสถิติอื่นอีกดังต่อไปนี้

ค่าสัมประสิทธิ์สหสัมพันธ์ = 0.9983

ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย = 216.449

รากของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย = 281.9019

ความคลาดเคลื่อนสัมพัทธ์สัมบูรณ์ = 5.8783 %

รากของความคลาดเคลื่อนสัมพัทธ์กำลังสอง = 5.841 %

4.2 ประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ กับ มาตรวัดแบบฟังก์ชันพอยต์

จากตารางที่ 3.12 ตารางการวิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย จะเป็นการสร้างสมการถดถอยจากจำนวนบรรทัดคำสั่ง และ จำนวนฟังก์ชันพอยต์จากการนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ แต่ในหัวข้อนี้สร้างสมการถดถอยจากการนับแบบฟังก์ชันพอยต์ปกติและคำนวณค่าประสิทธิภาพทางสถิติ เพื่อเปรียบเทียบกับค่าประสิทธิภาพของสมการถดถอยจากการนับแบบฟังก์ชันพอยต์ร่วมกับฟังก์ชันพอยต์ แสดงตามการคำนวณและตารางที่ 4.2

ตารางที่ 4.2 ตารางความคลาดเคลื่อนสัมบูรณ์และความคลาดเคลื่อนกำลังสอง

App.	UFP X_i	SLOC Y_i	X_i^2	X_iY_i
1	19	511	361	9709
2	62	1349	3844	83638
3	106	2204	11236	233624
4	90	2276	8100	204840
5	108	2488	11664	268704
6	117	2496	13689	292032
7	161	3674	25921	591514
8	218	5796	47524	1263528
9	349	8609	121801	3004541
10	675	15694	455625	10593450
	$\sum X = 1905$	$\sum Y = 45097$	$\sum X^2 = 699765$	$\sum XY = 16545580$

$$SS_{XX} = \sum_{1}^{n} X_i^2 - \frac{(\sum_{1}^{n} X_i)^2}{n} = 699765 - \frac{(1905)^2}{10} = 336862.5$$

$$SS_{XY} = \sum_{1}^{n} X_i Y_i - \frac{(\sum_{1}^{n} X_i)(\sum_{1}^{n} Y_i)}{n} = 165455580 - \frac{(1905)(45097)}{10} = 7954601.5$$

$$b = \frac{SS_{XY}}{SS_{XX}} = \frac{7954601.5}{336862.5} = 23.61$$

$$a = \bar{y} - b\bar{x} = \frac{45097}{10} - 23.61(\frac{1905}{10}) = 11.27$$

จะได้สมการประมาณขนาดซอฟต์แวร์คือ **SLOC = 23.61(UFP)+11.27** (4-1)

ตารางที่ 4.3 ตารางเปรียบเทียบค่าประสิทธิภาพทางสถิติ

ค่าประสิทธิภาพทางสถิติ	ฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ	ฟังก์ชันพอยต์ปกติ
ค่าสัมประสิทธิ์สหสัมพันธ์	0.9983	0.9953
ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย	216.449	368.2293
รากของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย	281.9019	509.0552
ความคลาดเคลื่อนสัมพัทธ์สัมบูรณ์	5.8783 %	10.0003 %
รากของความคลาดเคลื่อนสัมพัทธ์กำลังสอง	5.841 %	10.5476 %

จากสมการถดถอยและผลการทดสอบประสิทธิภาพด้วยวิธี Cross-Validation ของการนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บและการนับฟังก์ชันพอยต์ปกติ จะเห็นได้ว่าสมการที่ได้คือ SLOC = 23.47(UFP)-156.28 และ SLOC = 23.61(UFP)+11.27 ตามลำดับ โดยค่าคงที่ที่ได้คือค่าความชันของเส้นตรงจะมีค่าใกล้เคียงส่วนค่าคงที่อีกตัวหนึ่งจะมีค่าแตกต่างกัน แต่อย่างไรค่าคงที่ทั้ง 2 ไม่บอกประสิทธิภาพของสมการที่ได้มา ต้องทำการวัดประสิทธิภาพจากค่าประสิทธิภาพทางสถิติจากการทำ Cross-Validation ของการนับฟังก์ชันทั้ง 2 แบบ โดยการนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บจะมีค่าดีกว่าการนับแบบฟังก์ชันพอยต์ปกติ ตามตารางที่ 4.3 ซึ่งจะสรุปได้ดังนี้

ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย และ รากของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย ที่เป็นค่าความคาดเลื่อนของจำนวนบรรทัดคำสั่งที่คำนวณได้ จะเห็นได้ว่าจำนวนบรรทัดที่ความคาดเลื่อนจากสมการถดถอยของฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บมีค่าน้อยกว่า สมการสมการถดถอยของฟังก์ชันพอยต์ปกติประมาณหนึ่งเท่าตัวหรือ 50 เปอร์เซ็นต์

ความคลาดเคลื่อนสัมพัทธ์สัมบูรณ์ และ รากของความคลาดเคลื่อนสัมพัทธ์กำลัง มีค่าเป็นเปอร์เซ็นต์ซึ่งยิ่งมีค่าเปอร์เซ็นต์น้อยความคลาดเลื่อนก็น้อยด้วย และจากการทดสอบค่าทางสถิติทั้ง 2 ค่าดังกล่าว ค่าที่ได้จากฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บน้อยกว่า ค่าที่ได้ฟังก์ชันพอยต์ปกติประมาณ 4 เปอร์เซ็นต์ซึ่งเป็นค่าที่มากพอสมควร

ค่าสัมประสิทธิ์สหสัมพันธ์ที่บอกถึงความสัมพันธ์กันของตัวแปรอิสระ และ ตัวแปรตาม คือ จำนวนบรรทัดคำสัง กับ จำนวนฟังก์ชันพอยต์ ค่าสัมประสิทธิ์สหสัมพันธ์ของฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ มีค่ามากกว่าค่าสัมประสิทธิ์สหสัมพันธ์ของฟังก์ชันพอยต์ปกติ จากค่าประสิทธิภาพทางสถิติของทั้ง 2 การนับฟังก์ชันพอยต์จะเห็นได้ว่าประสิทธิภาพทางสถิติของการฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บมีค่าที่ดีกว่าทุกค่า

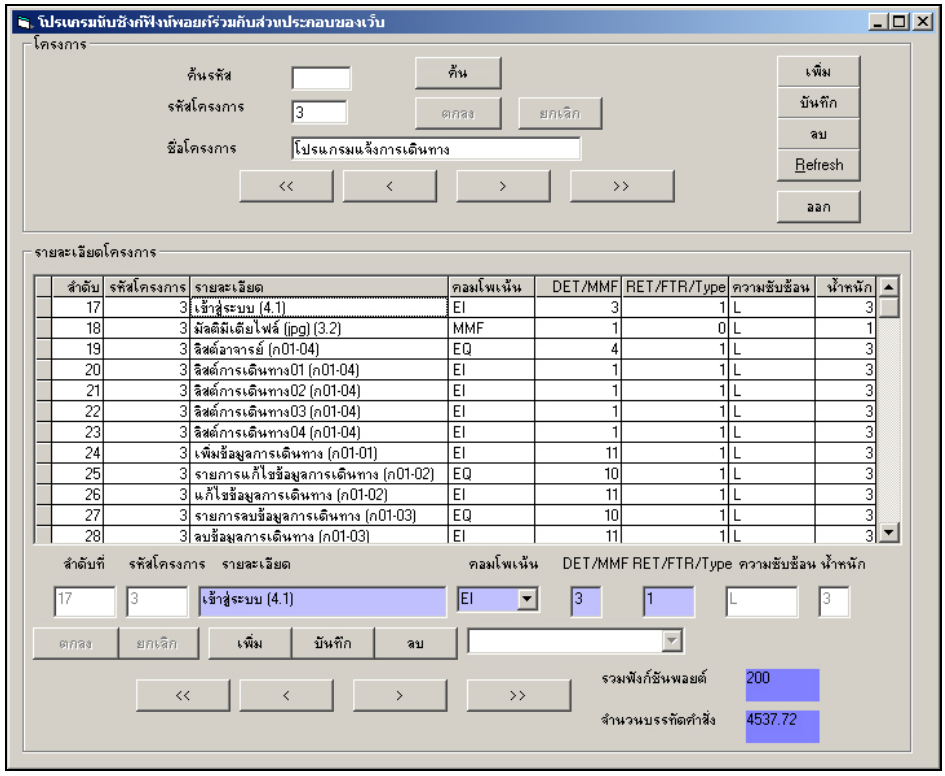
4.3 การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บด้วยการทดสอบด้วยโปรแกรมที่พัฒนาขึ้นจริง

การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บนี้ จะทำโดยการนำพัฒนาโปแกรมเว็บขึ้นมาใหม่ 1 โปรแกรม แล้วจะทำการกำหนดความต้องการของโปแกรม แล้วทำการนับฟังก์ชันพอยต์และคำนวณหาขนาดของโปรแกรมที่เป็นจำนวนบรรทัดคำสั่ง ด้วยโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บที่พัฒนาขึ้น แล้วเปรียบเทียบค่าที่คำนวณกับค่าจริงของโปรแกรมที่พัฒนาขึ้น จากนั้นจะคำนวณจำนวนความพยายามและระยะเวลาพัฒนาจากโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 เพื่อเป็นแนวทางในการพัฒนาโปรแกรม

โปรแกรมเว็บที่จะมาเป็นข้อมูลทดสอบเป็น โปรแกรมระบบแจ้งการเดินทางที่พัฒนาขึ้นมาเอง โดยมีรายละเอียดความต้องการของซอฟต์แวร์จะแสดงในภาคผนวก ง. และจำนวนฟังก์ชันพอยต์และจำนวนบรรทัดคำสั่งที่คำนวณได้จากโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บแสดงในภาพที่ 4.1 และรายละเอียดเพิ่มเติมในภาคผนวก จ. โดยได้ค่าฟังก์ชันพอยต์ที่นับได้เท่ากับ 200 และมีจำนวนบรรทัดคำสั่งเท่ากับ 4537.72 คิดเป็น 4538 บรรทัด

และเมื่อได้จำนวนบรรทัดคำสั่งแล้วก็นำค่าดังกล่าวไปคำนวณโดยโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 โดยกำหนดค่าตัวคูณความพยายามตามคุณลักษณะของรูปแบบโปรแกรมทีมงานนักพัฒนา เครื่องมือในการพัฒนาโปรแกรม และ ฯลฯ รายละเอียดตามหัวข้อ 2.3 แสดงตาม

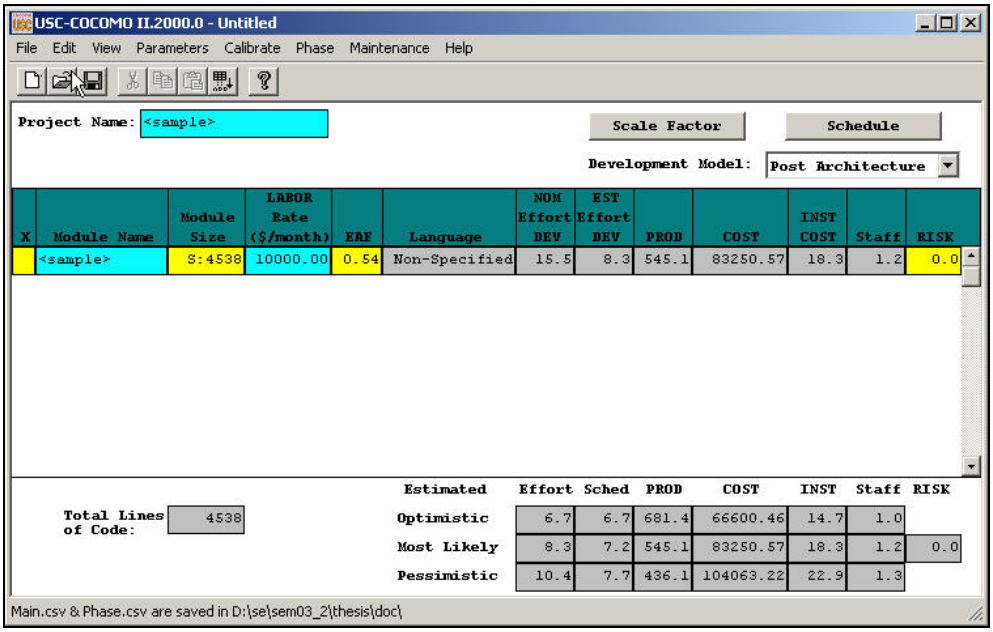
ตารางที่ 4.5 และกำหนดค่าสเกลของปัจจัย ระดับ ปานกลางทั้งหมด จะได้ค่าจำนวนความพยายาม 8.3 จำนวนนักพัฒนาต่อเดือน (Person-Months) ระยะเวลาพัฒนา 7.2 เดือน และจำนวนนักพัฒนา 1.2 คน แสดงตามรูปที่ 4.2 และสามารถจะแสดงเป็นขั้นตอนพัฒนาโปรแกรมแบบน้ำตก (Waterfall Phase) แสดงตามตารางที่ 4.4



รูปที่ 4.1 การนับฟังก์ชันพอยต์จากโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ

ตารางที่ 4.4 ขั้นตอนพัฒนาโปรแกรมแบบน้ำตก (Waterfall Phase)

Overall	Schedule (%)	Schedule (Months)	Effort (%)	Effort (PM)	Staff (P)
Plans And Requirements	16.85%	1.212804	7.00%	0.582754	0.480501
Product Design	24.42%	1.7583	17.00%	1.41526	0.804902
Programming	54.31%	3.909829	62.73%	5.222392	1.335708
Integration and Test	21.27%	1.531232	20.27%	1.687406	1.101992



รูปที่ 4.2 ผลการคำนวณของโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000

การวัดประสิทธิภาพมาตรวัดแบบฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บด้วยการทดสอบด้วยโปรแกรมที่พัฒนาขึ้นจริง โดยที่โปรแกรมที่พัฒนาขึ้นมีจำนวนบรรทัดคำสั่งเท่ากับ 4630 และสามารถคำนวณจากโปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ ได้จำนวนบรรทัดคำสั่งเท่ากับ 4538 บรรทัดคำสั่ง มีความคลาดเคลื่อนเป็น 92 บรรทัดคำสั่ง คิดเป็น 1.98 เปอร์เซ็นต์

ส่วนการคำนวณจำนวนความพยายามและระยะเวลาพัฒนาจากโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 ได้ค่าจำนวนความพยายาม 8.3 จำนวนนักพัฒนาต่อเดือน (Person-Months) ระยะเวลาพัฒนา 7.2 เดือน จำนวนนักพัฒนา 1.2 คน และขั้นตอนพัฒนาโปรแกรมแบบน้ำตก (Waterfall Phase) ตามตารางที่ 4.4 จะใช้ในการเป็นแนวทางในการพัฒนาโปรแกรม ส่วนการวัดประสิทธิภาพของค่าที่ได้จากแบบจำลองโคโคโม2 นั้นมีความซับซ้อนและต้องเก็บข้อมูลเป็นจำนวนมากจึงไม่สามารถวัดออกมาในงานวิจัยนี้ แต่ค่าที่ประมาณได้จากแบบจำลองโคโคโม2 มีการใช้งานกับอย่างแพร่หลายและมีการพัฒนาปรับปรุงอยู่ตลอดเวลา จึงมีความน่าเชื่อถือใช้เป็นแนวทางในการวางแผนพัฒนาโครงการได้

ตารางที่ 4.5 ค่าตัวคูณความพยายามสำหรับโปรแกรมแจ้งการเดินทาง

ตัวคูณความพยายาม	ความหมาย	ระดับ
RELY	ความเชื่อถือได้ของผลิตภัณฑ์ซอฟต์แวร์ที่ต้องการ	สูง
DATA	ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์	ปานกลาง
DOCU	ความต้องการเอกาสรที่ตรงกับวงจรชีวิต	ปานกลาง
CPLX	ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์	ปานกลาง
RUSE	ความต้องการที่จะนำผลิตภัณฑ์ซอฟต์แวร์กลับมาใช้ใหม่	ต่ำมาก
TIME	การจำกัดเวลาที่ใช้ในการประมวลผล	ปานกลาง
STOR	การจำกัดหน่วยความจำหลัก	ปานกลาง
PVOL	การเปลี่ยนแพลทฟอร์ม	ต่ำ
ACAP	ความสามารถในการวิเคราะห์	ปานกลาง
PCAP	ความสามารถของโปรแกรมเมอร์	ปานกลาง
PCON	ความต่อเนื่องของบุคลากร	สูงมาก
APEX	ประสบการณ์ใช้แอปพลิเคชัน	สูง
LTEX	ประสบการณ์โปรแกรมภาษาและเครื่องมือ	สูง
PLEX	ประสบการณ์แพลทฟอร์ม	สูง
TOOL	การใช้เครื่องมือซอฟต์แวร์	ปานกลาง
SITE	การพัฒนาในหลายสถานที่	ปานกลาง
SCED	กำหนดตารางการพัฒนาที่ต้องการ	ปานกลาง

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยครั้งนี้ได้เสนอการนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ ซึ่งเป็นการเพิ่มเติมส่วนประกอบเว็บที่เป็นมัลติมีเดียไฟล์ เพื่อจะช่วยนับส่วนประกอบเว็บเข้าไปเป็นพารามิเตอร์หนึ่งช่วยให้การคำนวณมีพารามิเตอร์มากขึ้น และทำให้การคำนวณมีประสิทธิภาพมากขึ้น คือ เมื่อการนับฟังก์ชันพอยต์สำหรับโปรแกรมเว็บนอกจากส่วนประกอบพื้นฐานของฟังก์ชันพอยต์ทั้ง 5 ส่วนประกอบก็ควรจะทำการนับส่วนประกอบของเว็บเข้าไปด้วย จากนั้นก็วิเคราะห์ความถดถอยเชิงเส้นอย่างง่าย เพื่อสร้างสมการถดถอย แล้วทำการทดสอบสมการซึ่งจะเห็นได้ว่าค่าประสิทธิภาพทางสถิติของสมการถดถอยจากการนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บมีค่าที่ดีกว่า ค่าประสิทธิภาพทางสถิติของสมการถดถอยจากการนับฟังก์ชันพอยต์ปกติ

งานวิจัยครั้งนี้ไม่เพียงแต่คำนึงถึงค่าประสิทธิภาพทางสถิติเท่านั้น แต่เป็นการนำการนับฟังก์ชันพอยต์เข้ามาทำการนับโปรแกรมเว็บซึ่งจากงานวิจัยอื่นๆ ยังไม่ได้ให้ความสำคัญกับการใช้งานแบบฟังก์ชันพอยต์ (Shu, ed, 2004) และ IFPUG ก็ยังไม่ออกคู่มือสำหรับการนับฟังก์ชันพอยต์สำหรับโปรแกรมเว็บโดยเฉพาะ (IFPUG, 2005) โดยงานวิจัยนี้ได้ทำการเลือกโปรแกรมเว็บที่จะมีลักษณะโครงสร้างทางภาษาที่เอื้ออำนวยเข้าการนับฟังก์ชันพอยต์ โดยได้เลือกโปรแกรมภาษาพีเอชพี(PHP)ซึ่งเป็นโปรแกรมที่นิยมและมีการใช้งานกันอย่างแพร่หลาย โดยโปรแกรมภาษานี้มีลักษณะโครงสร้างเป็นโปรแกรมเชิงโครงสร้าง (Structure Programming) โปรแกรมพัฒนาโดยการเขียนคำสั่งบรรทัดเป็นหลัก ซึ่งจะทำให้การประมาณขนาดของซอฟต์แวร์ที่เป็นจำนวนบรรทัดคำสั่งมีความสอดคล้องกันมากขึ้น อีกทั้งยังเป็นโปรแกรมแบบเปิดเผยซอร์สโค้ด (Open Sources) ทำให้มีการพัฒนาใช้งานกันมากในกลุ่มกันพัฒนาโปรแกรม ทำให้มีการแจกจ่ายและแลกเปลี่ยนซอร์สโค้ดในกลุ่มพัฒนาโปรแกรมเว็บภายใต้เงื่อนไข GNU General Public License ทำให้มีข้อมูลในงานวิจัยที่หลากหลายขึ้น

นอกจากการนำเอาโปรแกรมภาษาพีเอชพีมานับฟังก์ชันพอยต์เท่านั้น ยังมีการนำเอาระบบข้อมูลมายเอสคิวแอล (Mysql Database) เพื่อทำให้เหมาะสมกับการนับแบบฟังก์ชันพอยต์มากขึ้น ซึ่งในคู่มือการนับฟังก์ชันพอยต์(FPCPM) ของ IFPUG ในเวอร์ชันล่าสุดเวอร์ชัน 4.2.1 มีรูปแบบการนับที่อ้างอิงกับฐานข้อมูลเป็นส่วนใหญ่ เพราะโปรแกรมในปัจจุบันจะทำการจัดเก็บข้อมูลลง

ระบบฐานข้อมูลแทนการเก็นในรูปแบบไฟล์ ซึ่งการที่นำเอาโปรแกรมภาษาพีเอชพี่ร่วมกับฐานข้อมูลมายเอสคิวแอลทำให้มีโครงสร้างเหมาะกับการนับฟังก์ชันพอยต์มากขึ้น

และงานวิจัยนี้ได้พัฒนาโปรแกรมมือนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บเพื่อทำการเก็บข้อมูลและคำนวณค่าถ่วงน้ำหนัก และเมื่อใช้งานร่วมกับโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 จะเป็นมาตรวัดซอฟต์แวร์แบบฟังก์ชันพอยต์ (Measurement Tool) ซึ่งสามารถวัดขนาดซอฟต์แวร์ที่กำลังจะพัฒนา และวัดความก้าวของโครงการเมื่อทำการคำนวนระยะเวลาการพัฒนา (Duration) จากโปรแกรมทั้งสองอันนี้

5.2 การประยุกต์ผลการวิจัย

จากพัฒนาเครื่องมือนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ และใช้งานร่วมกับโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 สามารถใช้เป็นมาตรวัดซอฟต์แวร์สำหรับโครงการทำจะพัฒนาในอนาคต โดยเริ่มจากการวัดขนาดของซอฟต์แวร์ โดยการนวัดขนาดของซอฟต์แวร์ออกมาเป็นในรูปแบบของฟังก์ชันหรือฟังก์ชันพอยต์ โดยใช้โปรแกรมนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บช่วยในการนับ และ คำนวณขนาดของซอฟต์แวร์จากฟังก์ชันพอยต์มาเป็นจำนวนบรรทัดคำสั่ง (SLOC) และเมื่อได้จำนวนบรรทัดคำสั่งแล้วจะใช้โปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 คำนวณค่าต่างๆที่จำเป็นในการวางแผน วัดผลการดำเนิน และค่าความเสี่ยง ของโครงการ เช่น คำนวณความพยายาม (Effort) ระยะเวลาพัฒนา (Duration) จำนวนนักพัฒนา (Staff) และจำนวนผลผลิตที่ต้องทำ (Productivity)

จากค่าต่างๆทั้งหมดที่กล่าวมากนั้นจะเป็นข้อมูลพื้นฐานในการพัฒนาโครงการ จากเครื่องมือทั้ง 2 อันที่กล่าวมาแล้วจะช่วยให้ผู้ดูแลโครงการสามารถที่จะดำเนินโครงการเป็นไปด้วยราบรื่นและเสร็จตามกำหนดเวลา

5.3 ข้อเสนอแนะในการวิจัยต่อไป

จากเครื่องมือนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บและโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 ใช้ในงานวิจัยครั้งนี้จะเห็นได้ว่าในการทำงานจะต้องมีการส่งผ่านค่าจำนวนบรรทัดคำสั่งระหว่างโปรแกรม จึงยังไม่สะดวกในการใช้งาน ทำให้จะต้องมีการพัฒนาเครื่องมือนับฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บ บรรจุแบบจำลองโคโคโม2 และสามารถคำนวณข้อมูลต่างๆที่จะใช้ในการพัฒนาโครงการ โดยอาจจะมีการแสดงข้อมูลออกมีในรูปกราฟชนิด เพื่อจะสามารถเข้าใจถึงข้อมูลแต่ละชนิดได้ดียิ่งขึ้นด้วย

รายการอ้างอิง

กัลยา วานิชย์บัญชา. (2544). **หลักสถิติ**. พิมพ์ครั้งที่ 6. กรุงเทพ: โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

น้ำฝน อัศวเมฆิน.(2542). **การพัฒนาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์.** วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์ มหาวิทยาลัย.

พรศักดิ์ เกริกกวิน. (2539). **การพัฒนามาตรวัดซอฟต์แวร์โดยใช้เทคนิคฟังก์ชันพอยด์**. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

ศรินทร์ วัชรบุราคำ. (2542). **การพัฒนาเครื่องมือช่วยประเมินค่าใช้จ่ายของการพัฒนาซอฟต์แวร์โดยเทคนิคโคโคโม 2**. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

Agarwal, R., Kumart, M., Yogesh, Mallick, S., Bharadwaj, R.M., and Anantwar, D.(2001). Estimating software projects. **Software Engineering Notes**: 26(4):60-67.

Boehm, B.W., et al. (2000). **Software cost estimation with COMOMO II**. New Jersey: Prentice-Hall.

Dreger, J. B. (1989). **Function point analysis**. New Jersey: Prentice-Hall.

Garmus, D, and Herron, D. (2001) . **Function point analysis**.(5th ed.). Boston: Addison-Wesley.

International Function Point Users Group.(2002). **IT Measurement Practical advice from the Experts.** Boston: Addison-Wesley.

International Function Point Users Group. (2005). **Function Point Counting Practices Manual**. (n.p.).

Reifer, D.J. (2000). Web development: Estimating Quick-to-Market Software. **IEEE Software**.17(8): 57-64.

Reifer, D.J. (2002). Estimating Web Development Costs: There are Differences. **CrossTalk.** (June):13-17.

Ruhe, M., Jeffery, R.,and Wieczorek, I. (2003). Using Web Objects for Estimating Software Development Effort for Web Applications. **Proceedings of the Ninth International Software Metrics Symposium,** 30-37.

Suh, Woojong(editor).(2004). **Web engineering: Principles and Techniques**. Hershey,USA: Idea Group Publishing.

ภาคผนวก ก

บทความผลงานวิจัยที่นำเสนอในการประชุมวิชาการ

“KMITL International Conference on Science and Applied Science 2006”

DEVELOPMENT OF A MEASUREMENT TOOL FOR PHP LANGUAGE WITH MYSQL DATABASE

Chaknarin Kongcharoen and Pichayotai Mahatthanapiwat

School of Computer Engineering Suranaree University of Technology

Nakhon Ratchasima 30000, Thailand
E-mail : csncrk@ku.ac.th

ABSTRACT

The estimate of effort and schedule for Web development is a difficult task because there are many Web components. Function points are used to quantify the amount of functionality of a software product based on its requirement. However, they are widely used for general programming languages, This paper shows how to apply them with PHP languages with MYSQL Database in order to quantity the functionality for Web development. After calculating the size of the program in SLOC, we can use COCOMOII to estimate the effort and schedule length.

KEYWORDS Function Point Analysis, Source lines of code, COCOMO II , Web Component

1. INTRODUCTION

Function Point Analysis (FPA)[1][3] is an attempt to overcome difficulties associated with Source lines of code (SLOC) as a measure of software size, and to assist in developing a mechanism to predict effort associated with software development. Practically, we will translate FPs into Source lines of code and use another model (e.g.,COCOMO II) that uses SLOC as the independent variable to derive estimates of schedule.

This paper describes such a process which was modified for a web development from PHP Language with MYSQL database environment. FPA was used to produce a level of effort estimate in terms of SLOC, which can be add a web component (e.g., picture , multimedia file) to FPs components. The Constructive Cost Model II (COCOMO II) was used with FPA results to derive an estimate of resource requirements, which can be refined to account for complexities within the development environment.

2. FUNCTION POINT WITH WEB COMPONENT ANALYSIS

Function Points with Web Component Analysis are an extension of function points. Web components in the Web applications are central to the size of such applications. Table 1 shows each of the five FPs and one Web component category with the associated weight factors representing level of complexity.

2.1 Function Point with Web Component Categories

The base of our estimate are still function points. To extended them, we have shown that they still can be used to accurately estimate the size of web applications[4]. To do this, a Multi-Media Files need to be added to take the new factors that drive the size of web applications into account. To count Function Points with Web Component, we evaluate the following six components of a web system based upon user requirements and page layouts:

Table 1 Function Point with Web Component Factors

Components	Functions Levels			Notes
	Low	Average	High	
Internal Logical Files (ILF)	7	10	15	Unadjusted function points computed using these standard weightings using IFPUG counting conventions
External Interface Files (EIF)	5	7	10	
External Inputs (EI)	3	4	6	
External Outputs (EO)	4	5	7	
External Inquires (EQ)	3	4	6	
Multi-Media Files (MMF)	1	2	4	Multi-media files (graphics , animation , audio , video , etc.) -JPEG,GIF (Low) -A2b music, Microsoft picture (Average) -PCX Image, XIF: Image, AIFF Audio, Liquid Audio, Streaming Audio/Video (High)

2.2 Function Point with Web Component Calculations

Sample FPs calculations are shown in Table 2. To translate FPs into SLOC, a SLOC estimate is provided for each FPs category to take into account category-specific requirements common in PHP Language with MYSQL database development. It should be noted that the SLOC values presented in the table were derived from the case study of PHP Language with MYSQL database development projects into Source Lines of Code per Function point (SLOC/FP). The values of SLOC/FP should not be used as absolute guidelines until the values can be calibrated to a much larger sample of case study. It should also be noted that the definition of SLOC used is expected by COCOMO II. If a different parametric model had been used, the definition of SLOC would have to be modified accordingly.

The Function Point with Web Component Calculations are show in Table 2.

2.3 COCOMO II

COCOMO II provides an estimate of the effort and schedule required to complete a software development project. The primary cost driver is Delivered Source Instructions (DSI). The SLOC estimate derived in the FPA can be used as the DSI estimate in COCOMO II calculations. Using this approach, we can convert the level of effort estimate in FPA (i.e., SLOC) into personnel and schedule requirements as shown in Table 3.

COCOMO II is composed of three submodels i.e., Application Composite Model, Early Design Model and Post Architecture Model. In this research, we will use Post Architecture Model and Early Design Model

These two models are used in the development of Application Generator, System Integration, or Infrastructure developments. The Post-Architecture is a detailed model that is used once the project is ready to develop and sustain a fielded system. The system should have a life-cycle architecture package, which provides detailed information on cost driver inputs, and enables more accurate cost estimates. The Early Design model is a high-level model that is used to explore of architectural alternatives or incremental development strategies. This level of detail is consistent with the general level of information available and the general level of estimation accuracy needed.[2]

Both the Post-Architecture and Early Design models use the same functional form to estimate the amount of effort and schedule it will take to develop a software project. These nominal-schedule (NS) formulas exclude the cost driver for Required Development Schedule, SCED. The amount of effort in person-months, PM_{NS}, is estimated by the formula:

$$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_i \qquad \text{Eq.1}$$

$$\text{where} \quad E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

The amount of schedule, $TDEV_{NS}$, it will take to develop the product is estimated by the formula:

$$TDEV_{NS} = C \times (PM_{NS})^{F} \qquad \text{Eq.2}$$

$$\text{where} \quad F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$$

$$F = D + 0.2 \times [E - B]$$

The value of n, the number of effort multipliers, EM_i, is 17 for the Post-Architecture model effort multipliers, EM_i, and 7 for the Early Design model. SF_j stands for the exponential scale factors. The values of A, B, C, and D are A = 2.94, B = 0.91, C = 3.67 and D = 0.28

Table 2 Calculation the Function Point Count

Function	**Type** [a]	**Value**
Index Page Books Cart	EQ	6
	EI	17
	MMF	2
News	EQ	3
	MMF	1
News Admin	EQ	9
	EI	16
Books Detail	EQ	3
	MMF	1
Books Order	EI	13
Books Cart Admin	EQ	9
	EI	23
Book Database	ILF	21
Order Book Database	ILF	14
Mailing Database	ILF	14
News Database	ILF	14
Total unadjusted function point count		166
Source lines of code (SLOC) [b]		166*23
		3818

[a] ILF = Internal Logical Files ; EIF = External Interface Files
EI = External Inputs ; EO = External Outputs
EQ = External Inquires ; MMF = Multi-Media Files
[b] SLOC / UFP ≈ 23

Table 3 Calculation the COCOMO II

Simple Calculations
Size = KSLOC $\times\left(1+\frac{\text{REVL}}{100}\right)$; Percentage of Requirement Evolution and Volatility (REVL)
Size = 3.818 $\times\left(1+\frac{10}{100}\right)$; REVL = 5 = 4.008 KSLOC
$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$ where $E = B + 0.01 \times \sum_{j=1}^{5} SF_j$, A = 2.94, B = 0.91
$PM_{NS} = 2.94 \times 4.2^{E} \times \prod_{i=1}^{n} EM_i$ = 13.5 person-months
$TDEV_{NS} = C \times (PM_{NS})^{F}$ where $F = D + 0.2 \times [E - B]$, B = 0.91, C = 3.67, D = 0.28
$TDEV_{NS} = 3.67 \times (27.1)^{F}$ = 8.4 months

3. CONCLUSIONS

Managers of software development projects need cost estimation tools to help plan and control costs. FPA and COCOMO were developed with general language environments in mind. PHP languages with MYSQL Database present significant opportunities for achieving envelopment productivity improvements. Tools such as FPA can be extended with web component for using in these new environments and then used in conjunction with COCOMO II

The resulting tool is not only useful for projecting costs before a project begins, but also for assessing the sensitivities of effort and schedule due to changes in software development parameters (e.g., skill-level of programmers). Your estimates will improve as you become more familiar with the application and tools.

REFERENCES

[1] Albrecht, Alan J. and Gaffney, John E., Jr. **1983** Software Function Points, Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Transactions on Software Engineering*, Vol. Se-9, No. 6, pp.639-648.

[2] Boehm, Barry W., Abts, Chris, Brown, A. Winsor, Chulani, Sunita, Clark, Bradford K., Horowtiz, Ellis, Madachy, Ray, Reifer, Donald and Steece, Bert. **2000**, *Software Cost Estimation with COCOMO II*, New Jresey, Prentice-Hall.

[3] Garmus, David. and David Herron. **2001**, *Function Point Analysis Measurement Practices for Successful Software Projects*, New York, Addison-Wesley.

[4] Reifer, Donald J. **2002** Estimating Web Development Costs:There Are Differences, *Crosstalk*, pp.13-17

ภาคผนวก ข

การกำหนดค่าตัวคูณความพยายาม (Effort Multipliers) สำหรับแบบจำลองโพสอาร์คิเท็กเจอร์ (Post Architecture Model)

แบบจำลองโพสอาร์คิเท็กเจอร์ (The Post-Architecture Model)

แบบจำลองโพสอาร์คิเท็กเจอร์ เป็นแบบจำลองที่ใช้ประมาณค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ ซึ่งขยายขีดความสามารถมากขึ้น เนื่องจากผู้ใช้ต้องทราบข้อมูลเกี่ยวกับการพัฒนาซอฟต์แวร์ละเอียดขึ้น โดยผู้ใช้ต้องเป็นผู้ให้ข้อมูลเกี่ยวกับจำนวนบรรทัดคำสั่งของซอฟต์แวร์ที่จะพัฒนา และ ระดับของตัวขับค่าใช้จ่ายจากเดิม 7 ลักษณะเป็น 17 ลักษณะ เป็นข้อมูลพื้นฐานในการประมาณค่าใช้จ่ายซอฟต์แวร์ซึ่งตัวขับค่าใช้จ่ายทั้ง 17 ลักษณะแบ่งออกได้เป็น 4 กลุ่มใหญ่ๆ ดังนี้

1. ปัจจัยกลุ่มผลิตภัณฑ์ (Product Factors)

1.1 ความเชื่อถือได้ของผลิตภัณฑ์ซอฟต์แวร์ที่ต้องการ (Required Software Reliability [RELY])

หมายถึง ซอฟต์แวร์ต้องสามารถทำงานได้ในช่วงเวลาที่กำหนดอย่างต่อเนื่อง และถูกต้อง โดยไม่มีเหตุผิดปกติๆ เช่น การคำนวณผิดพลาด การไม่ทำงานตามขั้นตอน การหยุดทำงานโดยที่ยังไม่จบการทำงานหรือไม่ถูกสั่งให้หยุดเป็นต้น โดยการพิจารณาความน่าเชื่อถือของผลิตภัณฑ์ซอฟต์แวร์ตามแนวทางของโคโคโม2 จะพิจารณาจากความเสียหายที่เกิดขึ้นหลังจากซอฟต์แวร์ไม่สามารถทำงานได้ เพื่อประเมินระดับของความพยายามที่จะใช้ในการพัฒนาซอฟต์แวร์ ดังแสดงในตารางที่ ข.1

ตารางที่ ข.1 ความแตกต่างของแต่ละระดับของ RELY

ระดับ	เงื่อนไข
ต่ำที่สุด	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความไม่สะดวกในการทำงานแต่สามารถดำเนินงานต่อไปได้
ต่ำ	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายเพียงเล็กน้อยในการแก้ไขให้กลับมาทำงานได้เหมือนเดิม
ปานกลาง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เสียเวลาหรือเสียค่าใช้จ่ายปานกลางในการแก้ไขให้กลับมาได้เหมือนเดิม
สูง	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความสูญเสียทางการเงิน
สูงมาก	ถ้าซอฟต์แวร์ไม่สามารถทำงานได้ตามปกติจะทำให้เกิดความเสี่ยงต่อชีวิตมนุษย์

1.2 ขนาดของฐานข้อมูลที่ใช้ในผลิตภัณฑ์ซอฟต์แวร์ (Data Base Size[DATA])

หมายถึง ตัวขับค่าใช้จ่ายที่ได้จากสมการในการคำนวณคือ D/P

โดยที่ D คือ ขนาดของฐานข้อมูล (ไบต์)

P คือ ขนาดของซอฟต์แวร์ (จำนวนบรรทัดคำสั่งของโปรแกรม)

ซึ่งจะพบว่าขนาดของฐานข้อมูลมีความสำคัญสำหรับการพิจารณา เนื่องจากถ้าขนาดของฐานข้อมูลมีขนาดใหญ่มากก็จะทำให้ได้ค่าจาก D/P มาก ซึ่งหมายถึงว่าข้อมูลที่ใช้ในการทดสอบซอฟต์แวร์ก็ต้องมีขนาดใหญ่มากขึ้นด้วย ดังแสดงในตารางที่ ข.2

ตารางที่ ข.2 ความแตกต่างของแต่ละระดับของ DATA

ระดับ	เงื่อนไข
ต่ำ	D/P น้อยกว่า 10
ปานกลาง	D/P มากกว่าเท่ากับ 10 แต่น้อยกว่า 100
สูง	D/P มากกว่าเท่ากับ 100 แต่น้อยกว่า 1000
สูงมาก	D/P มากกว่า 100

1.3 ความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์ (Complexity[CPLX])

การพิจารณาความซับซ้อนของผลิตภัณฑ์ซอฟต์แวร์จะพิจารณา 5 อย่างด้วยกัน

1) การดำเนินการควบคุม (Control Operation)

การดำเนินการควบคุม หมายถึง กระบวนการที่ใช้ควบคุมการทำงานของโปรแกรมหรือลักษณะการทำงานของโปรแกรมในซอฟต์แวร์ที่จะพัฒนา ดังแสดงในตารางที่ ข.3

2) การดำเนินการคำนวณ (Computation Operation)

การดำเนินการคำนวณ หมายถึง วิธีการหรือรูปแบบการคำนวณที่มีใช้ในซอฟต์แวร์ที่จะพัฒนา ดังแสดงในตารางที่ ข.4

ตารางที่ ข.3 ความแตกต่างของแต่ละระดับของ Control Operation

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการ(programming operation)ของซอฟต์แวร์ ส่วนใหญ่จะเป็นการดำเนินการแบบตรงไปข้างหน้า (straight-line code) แต่ก็มีลักษณะการดำเนินการแบบเป็นเงื่อนไขแต่เป็นเงื่อนไขที่ไม่ซับซ้อน(non-nested structured) เช่น DO, CASE, IF-THEN-ELSE นอกจากนี้อาจมีการใช้โมดูลง่ายๆ เช่น การทำโปรซีเดอร์คอร์(procedure call) เป็นต้น
ต่ำ	การดำเนินการของซอฟต์แวร์จะมีโครงสร้างที่ซับซ้อน(nesting of structured programming) และโดยส่วนใหญ่จะเป็นแบบมีเงื่อนไข เช่น มีลูป WHILE ซ้อน IF-THEN-ELSE ซ้อน IF-THEN-ELSE เป็นต้น
ปานกลาง	การดำเนินการของซอฟต์แวร์โดยส่วนใหญ่จะเป็นแบบมีเงื่อนไขที่ซับซ้อน จึงมีการใช้ตารางตัดสินใจ(decision table) หรือ มีการเรียกใช้โปรแกรมหรือส่วนของโปรแกรมแบบ call back หรือแบบการส่งข้อความ(message passing) รวมทั้งมีการดำเนินการที่สนับสนุนการประมวลผลแบบกระจายบนเครือข่าย (distribute processing) เช่น ระบบ client/server
สูง	โครงสร้างของการดำเนินการมีความซับซ้อนมากซึ่งประกอบด้วยการดำเนินการที่เป็นแบบเงื่อนไขซ้อนเงื่อนไข หรือมีการใช้คิว (queue) และกองซ้อน (stack) เป็นตัวควบคุมลำดับการดำเนินการ หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความเหมือนกันหรือใกล้เคียงกัน(homogeneous) คือ มียี่ห้อหรือเวอร์ชันเดียวกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด (soft real time)
สูงมาก	การดำเนินการจะมีลักษณะเป็นแบบการเวียนบังเกิด(recursive) หรือมีการประมวลผลแบบกระจายซึ่งมีอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีความต่างกัน(heterogeneous) คือ มียี่ห้อหรือเวอร์ชันต่างกัน หรือมีตัวประมวลผลเพียงตัวเดียวและเป็นแบบทันทีทันใด(hard real time)
สูงที่สุด	การดำเนินการของซอฟต์แวร์จะต้องควบคุมการทำงานของฮาร์ดแวร์หรืออุปกรณ์โดยตรง(Microcode level control) หรือมีตารางการใช้ทรัพยากร (resource)ที่สามารถแก้ไขได้โดยอัตโนมัติตามความเหมาะสม หรือมีการประมวลผลแบบกระจาย(distribute processing) ที่ได้ผลลัพธ์แบบทันทีทันใด (hard real time)

ตารางที่ ข.4 ความแตกต่างของแต่ละระดับของ Computation Operation

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการคำนวณสามารถหาค่าจากสมการพีชคณิตหรือสมการเส้นตรง เช่น ทำการบวก การลบ การคูณ การหาร ตัวอย่างเช่น A=B+C*(D-E)
ต่ำ	การดำเนินการคำนวณสามารถหาคาจากสมการที่มีความยากระดับกลาง (moderate-level expressions) คือ นอกจากทำการบวก การลบ การคูณ การหาร แล้วยังสามารถทำการยกกำลัง การหาราก การคำนวณหาค่าตรีโกณมิติ ตัวอย่างเช่น D = SQRT (B*2 – 4*A*C)
ปานกลาง	การดำเนินการคำนวณมีการใช้รูทีน(routine)ที่เป็นมาตรฐานทางคณิตศาสตร์และสถิติ เช่นการหาส่วนเบี่ยงเบนมาตรฐาน การหาค่า Z เป็นต้นนอกจากนี้ยังคำนวณหาค่าของเมตริกซ์(matrix)และเวกเตอร์(vector)ได้
สูง	การดำเนินการคำนวณสามารถวิเคราะห์เชิงตัวเลข (numerical analysis) ขั้นพื้นฐานได้ เช่น การหารคำตอบของสมการโดยวิธีการของนิวตัน(Newton's Method)หรือโดยวิธีการของ Guassian การแก้สมการเชิงอนุพันธ์ เป็นต้น
สูงที่สุด	การดำเนินการคำนวณสามารถวิเคราะห์โครงสร้างเชิงตัวเลขที่มีโครงสร้างไม่แน่นอน เช่น การวิเคราะห์ระดับความสูงของเสียงรบกวน(highly accurate analysis of noisy) การวิเคราะห์ข้อมูลแบบ stochastic

3) การดำเนินการที่เกี่ยวกับอุปกรณ์ (Device-dependent Operation)

การดำเนินการที่เกี่ยวกับอุปกรณ์ หมายถึง วิธีหรือกระบวนการที่ซอฟต์แวร์ที่จะพัฒนาใช้งานอุปกรณ์ เช่นการสั่งให้เครื่องพิมพ์เอกสารพิมพ์รายงานการจัดเก็บข้อมูลในฮาร์ดดิสก์ เป็นต้น ดังแสดงในตารางที่ ข.5

ตารางที่ ข.5 ความแตกต่างของแต่ละระดับของ Device-dependent Operation

ระดับ	เงื่อนไข
ต่ำมาก	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์เป็นคำสั่งที่ใช้งานง่าย หรือเป็นภาษาระดับสูง เช่น การใช้คำสั่ง อ่าน (read) เขียน (write) ในภาษาพาสคาล (Pascal) ซึ่งเป็นคำสั่งที่ใช้ สำหรับการอ่านและบันทึกข้อมูล
ต่ำ	คำสั่งที่ใช้ดำเนินการเกี่ยวกับอุปกรณ์ สามารถใช้คำสั่งที่เป็นพื้นฐานที่ภาษามีการเตรียมไว้ให้ เช่น คำสั่งเขียนข้อมูลลงแฟ้มข้อมูล (put) หรือ อ่านข้อมูลในแฟ้มข้อมูล (get) ในภาษา C โดยที่ผู้พัฒนาไม่จำเป็นต้องมีความรู้เกี่ยวกับอุปกรณ์ที่ใช้
ปานกลาง	ผู้พัฒนาจะต้องมีความรู้อย่างเจาะจงเกี่ยวกับอุปกรณ์ที่เป็นตัวรับข้อมูลและตัวแสดงผล (input/output) โดยการใช้งานอุปกรณ์ต้องทำ 3 ขั้นดังนี้คือ เลือกอุปกรณ์ที่จะใช้งาน ตรวจสอบสถานะการทำงานอุปกรณ์นั้นและตรวจสอบความผิดพลาดในการทำงานของอุปกรณ์
สูง	การดำเนินการที่เกี่ยวข้องกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะกระทำในระดับกายภาพ (physical) เช่น การแปลงตำแหน่งของหน่วยความจำทางกายภาพ เพื่อช่วยในการค้นหาและการอ่าน และการกำหนดรูปแบบการซ้อนทับ (overlap) ในหน่วยความจำของอุปกรณ์
สูงมาก	การดำเนินการที่เกี่ยวกับอุปกรณ์ของตัวรับข้อมูลและตัวแสดงผลจะมีรูทีน (routines) สำหรับจัดการเมื่อเกิดความผิดพลาดในขณะส่งและรับข้อมูล มีการเกี่ยวกับการส่งข้อมูลในสายการสื่อสารหรือมีระบบการวัดประสิทธิภาพของอุปกรณ์แบบเข้มงวด (performance-intensive embedded system) เช่น มีการบันทึกการทำงานของอุปกรณ์ที่ทำงานผิดพลาดหรือไม่ทำงาน

4) การดำเนินการเกี่ยวกับการจัดการกับข้อมูล (Data Management Operation)
การดำเนินการเกี่ยวกับการจัดการขข้อมูล หมายถึง วิธีการจัดการเก็บข้อมูลและการใช้งานข้อมูลของซอฟต์แวร์ที่จะพัฒนา ดังแสดงในตารางที่ ข.6

ตารางที่ ข.6 ความแตกต่างของแต่ละระดับของ Data Management Operation

ระดับ	เงื่อนไข
ต่ำมาก	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลมีการใช้แถวลำดับ (array) ในหน่วยความจำหลัก เพื่อเก็บข้อมูล และสามารถใช้การสอบถาม (query) และการแก้ไข (update) ได้
ต่ำ	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้งานแฟ้มข้อมูลเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ แฟ้มข้อมูลสามารถใช้การสอบถามและการแก้ไขได้
ปานกลาง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถใช้แฟ้มข้อมูลได้หลายแฟ้มข้อมูลโดยจะเป็นแฟ้มส่งข้อมูลออกเพียงแฟ้มเดียว และไม่สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลได้ จะไม่มีการดำเนินการกับแฟ้มข้อมูลที่เกิดในระหว่างการทำงาน เช่น log file
สูง	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลสามารถทำทริกเกอร์ (trigger) โดยใช้ข้อมูลเพียงชุดเดียวสามารถทำการเปลี่ยนโครงสร้างของข้อมูลที่ซับซ้อน (Complex data restructuring) เช่น การเปลี่ยนคีย์ การเปลี่ยนความสัมพันธ์ระหว่างแฟ้มข้อมูล เป็นต้น
สูงที่สุด	การดำเนินการเกี่ยวกับการจัดการกับข้อมูลเป็นแบบพลวัต (dynamic relational) คือ สามารถเปลี่ยนแปลงโครงสร้างบางอย่างโดยอัตโนมัติตามสภาพของข้อมูลที่เก็บได้ มีโครงสร้างเชิงวัตถุ (object structures) มีการจัดการข้อมูลด้วยภาษาธรรมชาติ (natural language data management)

5) การดำเนินการส่วนต่อประสานกับผู้ใช้ (User interface Management operation)
การปฏิบัติการส่วนต่อประสานกับผู้ใช้ หมายถึง วิธีหรือรูปแบบซอฟต์แวร์ที่จะพัฒนาติดต่อกับผู้ใช้ ดังแสดงในตารางที่ ข.7

ตารางที่ ข.7 ความแตกต่างของแต่ละระดับของ Data Management Operation

ระดับ	เงื่อนไข
ต่ำมาก	มีโปรแกรมช่วยในการสร้าง (generators) รูปแบบการรับข้อมูลเข้า (input form) และการทำรายงาน (report) แบบง่าย เช่นมี wizard ช่วย
ต่ำ	มีการใช้ตัวสร้างส่วนต่อประสานกับผู้ใช้ (user interface) ที่ไม่ยุ่งยาก เช่น ใช้ตัวสร้าง GUI (graphic user interface builders)
ปานกลาง	มีการใช้งานส่วนต่อประสานกับผู้ใช้เป็นแบบวินโดว์ ที่ไม่มีความซับซ้อนมาก เช่น มีการติดต่อแบบ window 95 เป็นต้น
สูง	มีการใช้งานส่วนต่อประสานกับผู้ใช้แบบวินโดร์ซึ่งสามารถใช้เสียงเป็นตัวรับและแสดงผลและยังใช้ติดต่อแบบมัลติมีเดียอย่างง่ายได้ด้วย
สูงมาก	มีการใช้ภาพ 2 มิติ หรือ 3 มิติ ที่มีความซับซ้อนไม่มาก มีการใช้ภาพที่เป็นพลวัต (dynamic graphic) และมีการใช้มัลติมีเดีย (multimedia) ในการติดต่อกับผู้ใช้
สูงที่สุด	มีการใช้มัลติมีเดียที่ซับซ้อน เช่น วีดิโอคอนเฟอร์เร็น (video conference) และมีการใช้ภาพเสมือนจริง (virtual reality) ในการติดต่อกับผู้ใช้

1.4 ความต้องการเอกาสรที่ตรงกับวงจรชีวิต (Documentation match to life-cycle needs [DOCU])

หมายถึง ตัวขับค่าใช้จ่ายที่อยู่ในรูปของการจัดทำเอกสารประกอบการพัฒนาซอฟต์แวร์ได้ครบทุกขั้นตอนทั้งโครงการ ซึ่งขั้นตอนหรือรูปแบบของการจัดทำเอกสารก็อยู่กับผู้พัฒนาว่าใช้วงจรชีวิต (life-cycle) แบบใดในการพัฒนาซอฟต์แวร์ เช่น วอลเตอร์ฟอร์โมเดล (waterfall Model) อินครีเมนต์ดิวิลอปต์โมเดล (Increment Development Model) สไปรรอลโมเดล (Spiral Model) โปรโตไทป์ปิง (Prototyping Model) เป็นต้น ซึ่งการกำหนดระดับขั้นกับเอกสารที่จัดทำครบทุกขั้นตอนและเพียงพอกับความต้องการใช้งานเพียงใด ดังแสดงในตารางที่ ข.8

ตารางที่ ข.8 ความแตกต่างของแต่ละระดับของ DOCU

ระดับ	เงื่อนไข
ต่ำมาก	ไม่มีการจัดทำเอกสารการพัฒนาซอฟต์แวร์
ต่ำ	เอกสารจัดทำไม่ครอบคลุมในหลายๆครั้งของวงจรชีวิต
ปานกลาง	เอกสารจัดทำได้ครบตามข้อกำหนดขั้นต่ำของวงจรชีวิต
สูง	เอกสารจัดได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่ใช้ภายในองค์กรแต่ไม่ได้เป็นข้อกำหนดไว้ในวงจรชีวิต เพื่อทำให้เกิดความสมบูรณ์มากขึ้น
สูงมาก	เอกสารจัดทำได้ครบตามข้อกำหนดของวงจรชีวิต และมีเอกสารเพิ่มเติมที่มีความละเอียดเพื่อให้เกิดความสมบูรณ์มากที่สุด

1.5 ความต้องการที่จะนำผลิตภัณฑ์ซอฟต์แวร์กลับมาใช้ใหม่ (Required Reusability [RUSE])

หมายถึง ตัวขับค่าใช้จ่ายมาจากความพยายามสร้างซอฟต์แวร์ให้มีลักษณะที่เป็นองค์ประกอบ (Component) เพื่อสามารถนำกลับไปใช้ (Reuse) กับซอฟต์แวร์อื่นๆ ในอนาคต ซึ่งการที่จะนำบางส่วนของโปรแกรมกลับมาใช้ในลักษณะใดๆ ของซอฟต์แวต์ที่จะพัฒนานั้นก็ขึ้นอยู่กับ การออกแบบซอฟต์แวร์ให้มีลักษณะเป็นฟังก์ชันหรือเป็นโมดูลที่เล็กที่สุดที่มีฟังก์ชันเดียว ดังแสดงในตารางที่ ข.9

ตารางที่ ข.9 ความแตกต่างของแต่ละระดับของ RUSE

ระดับ	เงื่อนไข
ต่ำ	ไม่มีการออกแบบสำหรับการนำโปรแกรมกลับมาใช้ใหม่ในอนาคต
ปานกลาง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโครงการ (Project) เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำส่วนของโปรแกรมของ Microsoft word6 มาใช้กับ Microsoft word7 เป็นต้น
สูง	มีการออกแบบเพื่อให้สามารถนำบางส่วนของโปรแกรมที่ไม่ได้อยู่ในโปรแกรม (Program) เดียวกับซอฟต์แวร์ ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคต เช่น นำบางส่วนของโปรแกรมของ Microsoft excel มาใช้กับ Microsoft word เป็นต้น
สูงมาก	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมที่ไม่ได้อยู่ในระบบ (Product line)เดียวกับซอฟต์แวร์ที่จะพัฒนากลับมาใช้ใหม่ได้ในอนาคตเช่น นำส่วนของโปรแกรมของระบบซื้อ-ขาย มาใช้กับระบบธนาคาร เป็นต้น
สูงที่สุด	มีการออกแบบเพื่อให้สามารถนำโปรแกรมหรือบางส่วนของโปรแกรมไปใช้ได้ในหลายๆระบบ เช่น นำส่วนของโปรแกรมของระบบบัญชี ระบบการซื้อ-ขาย และระบบงานบุคคล มาใช้กับระบบธนาคาร เป็นต้น

2. ปัจจัยกลุ่มแพลทฟอร์ม (Platform Factors)

2.1 การจำกัดเวลาที่ใช้ในการประมวลผล (Execution Time Constraint[Time])

หมายถึง ตัวขับค่าใช้จ่ายที่มาจากเวลาที่เข้ากระทำการ (Execution Time) ขึ้นอยู่กับซอฟต์แวร์ระบบ (Software System) การกำหนดระดับของตัวขับค่าใช้จ่ายประเภทนี้คิดเป็นเปอร์เซนต์ ของเวลาที่คาดว่าซอฟต์แวร์จะเข้ากระทำการ จากเวลาทั้งหมดที่ระบบมีให้กับซอฟต์แวร์ ดังแสดงในตารางที่ ข.10

ตารางที่ ข.10 ความแตกต่างของแต่ละระดับของ TIME

ระดับ	เงื่อนไข
ปานกลาง	เวลาที่ซอฟต์แวร์เข้ากระทำการน้อยกว่าหรือเท่ากับ 50% ของเวลาที่ซีพียู มีให้
สูงมาก	เวลาที่ซอฟต์แวร์เข้ากระทำมากกว่า 50 % ของเวลาที่ซีพียูมี ให้แต่น้อยกว่าหรือเท่ากับ 70 % ของเวลาที่ซีพียูมีให้
สูงมาก	เวลาที่ซอฟต์แวร์เข้ากระทำมากกว่า 70 % ของเวลาที่ซีพียูมี ให้แต่น้อยกว่าหรือเท่ากับ 85 % ของเวลาที่ซีพียูมีให้
สูงที่สุด	เวลาที่ซอฟต์แวร์เข้ากระทำการมากกว่า 85% ของเวลาที่ซีพียูมีให้น้อยกว่าหรือเท่ากับ 95% ของเวลาที่ซีพียูมีให้

2.2 การจำกัดหน่วยความจำหลัก (Main Storage Constraint[STOR])

หมายถึง ตัวขับค่าใช้จ่ายที่มาจากการพิจารณาว่าซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากน้อยเพียงใด ซึ่งถ้าซอฟต์แวร์ใช้หน่วยความจำมากซอฟต์แวร์ก็จะต้องมีการจัดการหน่วยความจำที่ดี การกำหนดระดับจะพิจารณาโดยคิดเป็นเปอร์เซ็นต์ของขนาดหน่วยความจำหลักที่คาดว่าจะใช้จากขนาดหน่วยความจำหลักที่ระบบให้ ดังแสดงในตารางที่ ข.11

2.3 การเปลี่ยนแพลทฟอร์ม (Platform Volatility[PVOL])

หมายถึง ซอฟต์แวร์ที่จะพัฒนาต้องถูกออกแบบหรือพัฒนาให้มีความคล่องตัวสูงในการเปลี่ยนแปลงหรือปรับปรุงให้เข้ากับแพลทฟอร์มใหม่ ซึ่งถ้าซอฟร์แวร์ที่จะพัฒนาต้องทำการเปลี่ยนบ่อยๆ ก็ทำให้เกิดความลำบากในการพัฒนา หรือใช้เวลามากในการพัฒนาเพื่อให้ซอฟต์แวร์สามารถทำงานบนหลายๆ แพลทฟอร์มได้ เป็นต้น แพลทฟอร์มในที่นี้จะมีความหมายรวมทั้งฮาร์ดแวร์และซอฟต์แวร์ (OS,DBMS etc) ตัวอย่างเช่น ถ้าซอฟต์แวร์ที่จะพัฒนาเป็นระบบปฏิบัติการ แพลทฟอร์มคืออุปกรณ์คอมพิวเตอร์ ถ้าซอฟต์แวร์ที่จะพัฒนาเป็นระบบจัดการฐานข้อมูล (database management system) แพลทฟอร์มคือ ระบบปฏิบัติการ ระบบจัดการฐานข้อมูล และฮาร์ดแวร์ ถ้าซอฟต์แวร์ที่จะพัฒนาคือตัวค้นหาข้อความบนเครือข่าย (network text browser) แพลทฟอร์มคือ ฮาร์แวร์ ระบบปฏิบัติการ และคลังเก็บข้อมูลแบบกระจาย (distribute information repositories) นอกจากนี้แพลทฟอร์มยังรวมไปถึงคตัวแปลงภาษา (compiler) ด้วย ดังในตารางที่ ข.12

ตารางที่ ข.11 ความแตกต่างของแต่ละระดับของ STOR

ระดับ	เงื่อนไข
ปานกลาง	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักน้อยกว่าหรือเท่ากับ 50% ของหน่วยความจำหลักที่สามารถใช้ได้
สูง	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากกว่า 50% ของหน่วยความจำหลักที่สามารถใช้ได้ แต่น้อยกว่าหรือเท่ากับ 70% ของหน่วยความจำหลักที่สามารถใช้ได้
สูงมาก	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากกว่า 70% ของหน่วยความจำหลักที่สามารถใช้ได้ แต่น้อยกว่าหรือเท่ากับ 85% ของหน่วยความจำหลักที่สามารถใช้ได้
สูงที่สุด	ซอฟต์แวร์จะใช้พื้นที่ในหน่วยความจำหลักมากกว่า 85% ของหน่วยความจำหลักที่สามารถใช้ได้ แต่น้อยกว่าหรือเท่ากับ 95% ของหน่วยความจำหลักที่สามารถใช้ได้

ตารางที่ ข.12 ความแตกต่างของแต่ละระดับของ PVOL

ระดับ	เงื่อนไข
ต่ำ	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 12 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 1 เดือน
ปานกลาง	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 6 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 2 สัปดาห์
สูง	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 เดือน หรือทำการปรับปรุงบางส่วนทุกๆ 1 สัปดาห์
สูงมาก	จะทำการเปลี่ยนแพลทฟอร์มทุกๆ 2 สัปดาห์ หรือทำการปรับปรุงบางส่วนทุกๆ 2 วัน

3. ปัจจัยกลุ่มบุคลากร (Personal Factor)

3.1 ความสามารถในการวิเคราะห์ (Analyst Capability[ACAP])

หมายถึง ตัวขับค่าใช้จ่ายที่พิจารณาจากความสามารถของบุคลากรในการวิเคราะห์ระบบงานตามที่ร้องขอ (Request) และการออกแบบซอฟต์แวร์อย่างรายละเอียด (Detailed Design) เช่น การออกแบบเกี่ยวกับข้อมูล (Data Design) การออกแบบเกี่ยวกับโครงสร้าง (Architectural Design) การออกแบบเกี่ยวกับกระบวนการ (Procedural Design) การออกแบบเกี่ยวกับการเชื่อมต่อ (Interface Design) เป็นต้น ปัจจัยหลักที่ใช้ในการแบ่งระดับความสามารถคือ ความสามารถ (ability)ในวิเคราะห์ออกแบบ ประสิทธิภาพ (efficiency) และความละเอียด (thoroughness) ในการวิเคราะห์ออกแบบ และความสามารถในการสื่อสารกับผู้อื่นๆ ในการกำหนดระดับจะไม่นำประสบการณ์ในการวิเคราะห์มาพิจารณาด้วยเนื่องจากได้แยกออกเป็นอีกหนึ่งตัวขับค่าใช้จ่าย การกำหนดระดับของสามารถในการวิเคราะห์จะพิจารณาความสามารถโดยเฉลี่ยของนักวิเคราะห์ระบบในทีมว่ามีความสามารถอยู่ในตำแหน่งเปอร์เซ็นต์ไทส์ที่เท่าใด ดังแสดงในตารางที่ ข.13

ตารางที่ ข.13 ความแตกต่างของแต่ละระดับของ ACAP

ระดับ	เงื่อนไข
ต่ำมาก	เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25
ต่ำ	เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45
ปานกลาง	เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65
สูง	เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85
สูงมาก	เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

3.2 ความสามารถของโปรแกรมเมอร์ (Programmer Capability[PCAP])

หมายถึง ตัวขับค่าใช้จ่ายมาจากความสามารถทีมนักพัฒนา ถึงแม้ว่ามีการให้ความสำคัญกับการวิเคราะห์ระบบอย่างมาก แต่เครื่องมือในปัจจุบันก็ได้พัฒนาจนมีความซับซ้อนมากขึ้น ดังนั้นการใช้งานเครื่องมือเหล่านี้ก็ขึ้นกับความสามารถของโปรแกรมเมอร์ การจะทำกำหนดระดับให้นั้นควรจะพิจารณาบนพื้นฐานของความสามารถโดยรวมของโปรแกรมเมอร์ในทีมที่พัฒนามากกว่าที่จะพิจารณาโปรแกรมเมอร์เพียงคนเดียว ปัจจัยร่วมที่มีส่วนช่วยในการกำหนดระดับคือ ความสามารถในการพัฒนาซอฟต์แวร์ ประสิทธิภาพของซอฟต์แวร์ ความสมบูรณ์ของซอฟต์แวร์ที่พัฒนา และความสามารถในการติดต่อสื่อสารกับผู้อื่น ตัวขับค่าใช้จ่ายประเภทนี้จะไม่นำประสบการณ์ในการเขียนโปรแกรมมาพิจารณาด้วย เนื่องจากได้แยกออกเป็นอีกหนึ่งตัวขับ

ค่าใช้จ่าย การกำหนดระดับจะพิจารณาจากความสามารถเฉลี่ยในการเขียนโปรแกรมของทีมโปรแกรมเมอร์ว่าอยู่ในตำแหน่งของเปอร์เซ็นต์ไทล์ที่เท่าใด ดังแสดงในตารางที่ ข.14

ตารางที่ ข.14 ความแตกต่างของแต่ละระดับของ PCAP

ระดับ	เงื่อนไข
ต่ำมาก	เปอร์เซ็นต์ไทล์ที่ 0 ถึง 25
ต่ำ	เปอร์เซ็นต์ไทล์ที่ 26 ถึง 45
ปานกลาง	เปอร์เซ็นต์ไทล์ที่ 46 ถึง 65
สูง	เปอร์เซ็นต์ไทล์ที่ 66 ถึง 85
สูงมาก	เปอร์เซ็นต์ไทล์ที่ 86 ถึง 100

3.3 ประสบการณ์ใช้แอพพลิเคชัน (Applications Experience[APEX])

หมายถึง ตัวขับค่าใช้จ่ายที่ขึ้นอยู่กับประสบการณ์ในการพัฒนาซอฟต์แวร์ (แอพพลิเคชัน) ชนิดเดียวกับซอฟต์แวร์ที่จะพัฒนา ซึ่งในตัวขับค่าใช้จ่ายนี้จะพิจารณาจากระยะเวลาเฉลี่ยที่ทีมพัฒนาเคยใช้พัฒนาซอฟต์แวร์ประเภทเดียวกับซอฟต์แวร์ที่จะพัฒนา ดังแสดงในตารางที่ ข.15

ตารางที่ ข.15 ความแตกต่างของแต่ละระดับของ APEX

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 7 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปี แต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

3.4 สบการณ์แพลทฟอร์ม (Platform Experience[PLEX])

หมายถึง ตัวขับค่าใช้จ่ายที่พิจารณาจากการทีมที่พัฒนามีประสบการณ์ในการใช้แพลทฟอร์มได้เป็นอย่างดี ซึ่งแพลทฟอร์มในที่นี้รวมทั้งฮาร์ดแวร์และซอฟต์แวร์ เช่น มีการรวมรูปภาพกับตัวเชื่อมประสาน (user interface) มีการใช้ฐานข้อมูล มีการใช้ระบบเครือข่าย (networking) เป็นต้น ซึ่งในตัวขับค่าใช้จ่ายนี้จะพิจารณาจากระยะเวลาที่ทีมพัฒนาเคยใช้แพลทฟอร์มประเภทเดียวกับ

แพลฟอร์มที่จะใช้ในซอฟต์แวร์ที่จะพัฒนา ดังแสดงในตารางที่ ข.16

ตารางที่ ข.16 ความแตกต่างของแต่ละระดับของ PLEX

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปี แต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

3.5 ประสบการณ์โปรแกรมภาษาและเครื่องมือ (Language and Tool Experience [LTEX])

หมายถึง ตัวขับค่าใช้จ่ายที่เป็นการวัดประสบการณ์ในการใช้ภาษาโปรแกรม (Programming language) และใช้เครื่องมือในการพัฒนาซอฟต์แวร์ การพัฒนาซอฟต์แวร์ยังรวมถึงการใช้งานเครื่องมือ พัฒนาซอฟ์แวร์ในการวิเคราะห์และออกแบบ การจัดการโครงแบบ (Configuration management) การจัดทำเอกสารการพัฒนาซอฟต์แวร์ การจัดการไลบรารี่ (library management) ของโปรแกรมภาษา การกำหนดรูปแบบการเขียนโปรแกรม (program style) ที่เป็นมาตรฐานเดียวกัน และการตรวจสอบความสอดคล้อง (consistency checking) ของโปรแกรม เป็นต้น นอกจากนี้การมีประสบการณ์ในการเขียนโปรแกรมในภาษาใดภาษาหนึ่งจะมีผลต่อระยะเวลาที่ใช้ในการพัฒนาด้วย ดังแสดงในตารางที่ ข.17

ตารางที่ ข.17 ความแตกต่างของแต่ละระดับของ LTEX

ระดับ	เงื่อนไข
ต่ำมาก	น้อยกว่าหรือเท่ากับ 2 เดือน
ต่ำ	มากกว่า 2 เดือนแต่น้อยกว่าหรือเท่ากับ 6 เดือน
ปานกลาง	มากกว่า 6 เดือนแต่น้อยกว่าหรือเท่ากับ 1 ปี
สูง	มากกว่า 1 ปี แต่น้อยกว่าหรือเท่ากับ 3 ปี
สูงมาก	มากกว่า 3 ปี

3.6 ความต่อเนื่องของบุคลากร (Personal Continuity[PCON])

หมายถึง ความต่อเนื่องของบุคลากรหมายถึง การหมุนเวียนของบุคลากรที่อยู่ในทีมพัฒนาซอฟต์แวร์ตั้งแต่เริ่มพัฒนาจนกระทั่งเสร็จสมบูรณ์ โดยถ้ามีการหมุนเวียนเพียงเล็กน้อย ซอฟต์แวร์ก็น่าจะมีความต่อเนื่องในการพัฒนา บุคลากรในที่นี้หมายถึง บุคลากรในทุกตำแหน่งที่ร่วมอยู่ในทีมพัฒนา เช่น ผู้จัดการโครงการ ผู้ช่วยผู้จัดการโครงการ นักวิเคราะห์ระบบ โปรแกรมเมอร์ พนักงานทั่วไปเลขานุการ เป็นต้น ดังแสดงในตารางที่ ข.18

ตารางที่ ข.18 ความแตกต่างของแต่ละระดับของ PCON

ระดับ	เงื่อนไข
ต่ำมาก	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 48% ต่อปี
ต่ำ	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 24% ต่อปี แต่น้อยกว่า 48 % ต่อไป
ปานกลาง	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 12% ต่อปี แต่น้อยกว่า 24 % ต่อไป
สูง	บุคลากรมีการเปลี่ยนแปลงมากกว่าหรือเท่ากับ 6% ต่อปี แต่น้อยกว่า 12 % ต่อไป
สูงมาก	บุคลากรมีการเปลี่ยนแปลงน้อยกว่า 6% ต่อปี

4. ปัจจัยกลุ่มโครงการ (Project Factor)

4.1 การใช้เครื่องมือซอฟต์แวร์ (Use of Software tools[TOOL])

หมายถึง ความสามารถหรือประสิทธิภาพของเครื่องมือซอฟต์แวร์ได้พัฒนาจนเป็นที่ยอมรับว่ามีผลต่อการพัฒนาโครงการ เนื่องจากเครื่องมือในสมัยแรกๆ สามารถทำได้เพียงการบรรณาธิการ (edit) และเขียน (Code) โปรแกรม แต่ต่อมาเครื่องมือได้มีการพัฒนาจนสามารถนำมาใช้ในทุกขั้นของวงจรชีวิตของการพัฒนา (Software Development Life Cycle:SDLC) ได้ เช่น ขั้นตอนการวิเคราะห์ (Analysis) ขั้นตอนการออกแบบ (Design) และขั้นตอนการทำให้เกิดผล (Implementation) ดังแสดงในตารางที่ ข.19

4.2 การพัฒนาในหลายสถานที่ (Multisite Development[SITE])

หมายถึง ตัวขับค่าใช้จ่ายที่ถูกเพิ่มเข้ามาในแบบจำลองโคโคโม2 เนื่องจากการวิจัยพบว่าสถานที่ที่ใช้พัฒนาและอุปกรณ์ที่ใช้ในการสื่อสารระหว่างบุคลากรในทีมพัฒนา มีผลต่อการพัฒนาซอฟต์แวร์เช่น สถานที่ที่ห่างไกลกันหรืออุปกรณ์การสื่อสารที่ไม่ทันสมัยจะทำให้ไม่สามารถแก้ไขปัญหา ได้ทันท่วงที เนื่องจากความล่าช้าในการติดต่อ ดังแสดงในตารางที่ ข.20 และตารางที่ ข.21

ตารางที่ ข.19 ความแตกต่างของแต่ละระดับของ TOOL

ระดับ	เงื่อนไข
ต่ำมาก	เครื่องมือสามารถเขียน (Code) บรรณาธิการ (Edit) และตรวจสอบความผิดพลาดในการเขียนโปรแกรมได้ (Debug)
ต่ำ	เครื่องมือที่ใช้เป็นแบบเครื่องมือเคสที่ทำฟ้อนเอนด์เคส (Front-end Computer Aided software Engineering : Front end CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงแรกๆ เช่น ช่วยในขั้นตอนของการเก็บรวบรวมความต้องการต่างๆ เกี่ยวกับซอฟต์แวร์ที่จะพัฒนา (Requirements phase) ช่วยในขั้นตอนของการวางแผนการพัฒนาซอฟต์แวร์ (Planning phase) หรือ ช่วยในขั้นตอนการออกแบบ (Design phase) หรือแบบเครื่องมือเคสที่ทำแบล็กเอนด์เคส (Backend CASE) คือ เครื่องมือเคสที่ช่วยพัฒนาซอฟต์แวร์ตามวงจรชีวิตในช่วงหลังเช่น ช่วนในขั้นตอนของการทำให้เกิดผล (Implementation phase) ช่วยในขั้นตอนของการรวม (Integration phase) หรือช่วยในขั้นตอนของการบำรุงรักษา (Maintenance phase)
ปานกลาง	เครื่องมือที่ใช้พัฒนาซอฟต์แวร์สามารถใช้ได้ในทุกขั้นตอนของวงจรชีวิต ในขั้นพื้นฐานได้ นอกจากนี้ยังสามารถทำการรวมบางขั้นตอนของวงจรชีวิตได้
สูง	เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นตอนของวงจรชีวิต และยังสามารถทำการรวมทุกขั้นตอนของวงจรชีวิตได้
สูงมาก	เครื่องมือใช้ในการพัฒนาซอฟต์แวร์สามารถใช้ได้ทุกขั้นตามรูปแบบของวงจรชีวิตนอกจากนี้ยังสามารถทำการรวมกันโดยระเบียบวิธีการ (Method) ขบวนการ (Process) หรือการนำกลับมาใช้ (Reuse)

4.3 กำหนดตารางการพัฒนาที่ต้องการ (Required Development Schedule[SCED])

หมายถึง ตัวขับค่าใช้จ่ายที่มาจากการกำหนดระดับจะขึ้นอยู่กับข้อจำกัดของทีมพัฒนา ดังนั้นจึงจะพิจารณาเป็นทีมที่พัฒนาโดยจะเทียบเป็นเปอร์เซ็นต์ของการขยายเวลาการทำงานออกไป หรือ การเร่งเวลาการทำงานให้เร็วขึ้นเมื่อเทียบกับเวลาการทำงานปกติ ซึ่งการเร่งตารางการทำงานมีแนวโน้มว่าจะใช้ความพยายามสูงในเฟตท้ายๆ ของการพัฒนา เนื่องจากอาจมีปัญหาหรือข้อสงสัยหลายอย่างที่ไม่ได้ถูกพิจารณาหรือแก้ไขในเฟสแรกๆ ดังนั้นจึงต้องมาทำในเฟสท้ายๆ และการขยายเวลาการทำงานออกไปมีแนวโน้มว่าจะใช้ความพยายามในเฟสแรกๆ มาก เนื่องจากใช้ในการวางแผนการพัฒนาซอฟต์แวร์ (Planning) การกำหนดคุณลักษณะ (specifications) ของซอฟต์แวร์ที่จะพัฒนา และการวางแผนการตรวจสอบความถูกต้อง (Validation) ของซอฟต์แวร์ ดังแสดงในตารางที่ ข.22

ตารางที่ ข.20 ความแตกต่างของแต่ละระดับของ SITE เกี่ยวกับสถานที่

ระดับ	เงื่อนไขของสถานที่
ต่ำมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีทั้งในประเทศและต่างประเทศ
ต่ำ	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด และมีบริษัทที่พัฒนาหลายบริษัท
ปานกลาง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในหลายจังหวัด หรือ มีบริษัทที่พัฒนาหลายบริษัท หรือ ตั้งอยู่ในจังหวัดที่ไม่สำคัญเพียงจังหวัดเดียว เช่น หนองบัวลำพูล ปัตตานี
สูง	สถานที่ที่พัฒนาซอฟต์แวร์ตั้งอยู่ในกรุงเทพฯ หรือ จังหวัดสำคัญ (กรุงเทพฯ เชียงใหม่ นครราชสีมา ภูเก็ต สงขลา)
สูงมาก	สถานที่ที่พัฒนาซอฟต์แวร์มีหลายตึก หรือ เป็นหมู่ตึก
สูงที่สุด	สถานที่ที่พัฒนาซอฟแวร์จะอยู่ในบริเวณเดียวกัน เช่น อยู่ในตึกเดียวกัน อยู่ในชั้นเดียวกัน เป็นต้น

ตารางที่ ข.21 ความแตกต่างของแต่ละระดับของ SITE เกี่ยวกับอุปกรณ์การสื่อสาร

ระดับ	เงื่อนไขของการสนับสนุนด้านการสื่อสาร
ต่ำมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ โทรศัพท์รวม หรือจดหมาย
ต่ำ	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือโทรศัพท์ส่วนตัว หรือแฟ็กส์ส่วนตัว
ปานกลาง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณแคบ เช่น ใช้สัญญาณคลื่นวิทยุ เครือข่ายเฉพาะ (private network) เครือข่ายท้องถิ่น (LAN) เป็นต้น
สูง	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การใช้ช่องสัญญาณกว้าง เช่น ใช้สัญญาณดาวเทียม เครือข่ายสาธารณะ(WAN)
สูงมาก	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ การประชุมทางไกลผ่านวิดีทัศน์
สูงที่สุด	เทคโนโลยีสูงสุดที่ใช้ในการสื่อสารระหว่างผู้พัฒนา คือ มัลติมีเดียที่สามารถโต้ตอบได้

ตารางที่ ข.22 ความแตกต่างของแต่ละระดับของ SCED

ระดับ	เงื่อนไข
ต่ำมาก	เวลาในการพัฒนาน้อยกว่าหรือเท่ากับ 75% ของเวลาปกติ
ต่ำ	เวลาในการพัฒนามากกว่า 75% ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 85%
ปานกลาง	เวลาในการพัฒนามากกว่า 85% ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 100%
สูง	เวลาในการพัฒนามากกว่า 100% ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 130%
สูงมาก	เวลาในการพัฒนามากกว่า 130% ของเวลาปกติแต่น้อยกว่าหรือเท่ากับ 160%

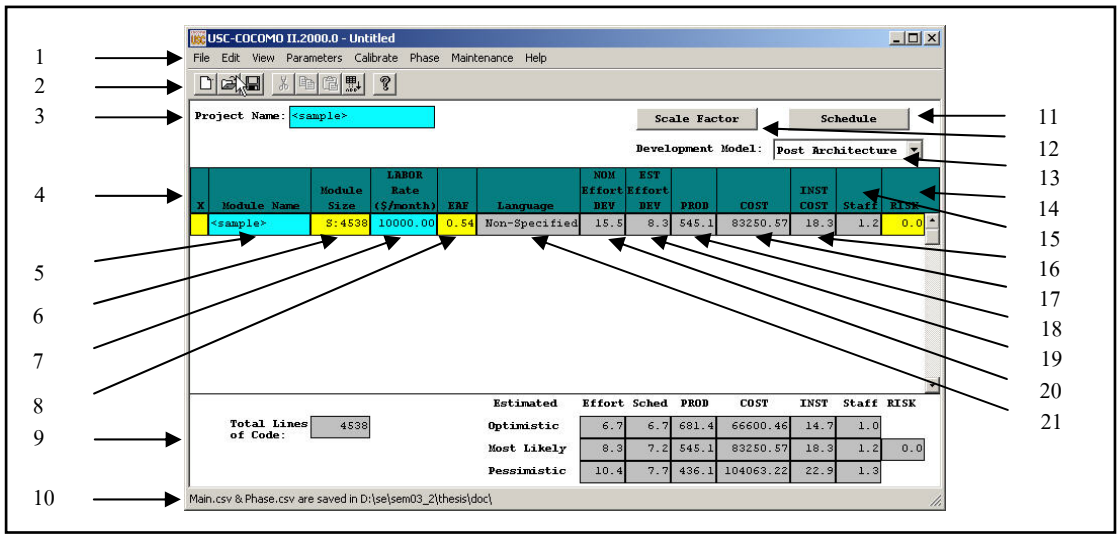
ภาคผนวก ค

คู่มือการใช้งานโปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000

โปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000

1. ส่วนประกอบของโปรแกรม (Navigating COCOMO)

โปรแกรมยูเอสซี โคโคโม2 เวอร์ชัน 2000 เป็นโปรแกรมที่สามารถคำนวณค่าต่างๆที่จำเป็นในการวางแผนในการพัฒนาโครงการเช่น จำนวนความพยายาม ระยะการพัฒนา จำนวนนักพัฒนา และอื่นๆซึ่งจะอธิบายลำดับต่อไป โดยจะใช้สูตรการคำนวณจากแบบจำลองโคโคโม2 และเป็นโปรแกรมประเภทที่ติดตั้งสำหรับผู้ใช้คนเดียว มีติดต่อกับผู้ใช้ในลักษณะเป็นส่วนต่อประสานกราฟิกส์(GUI)

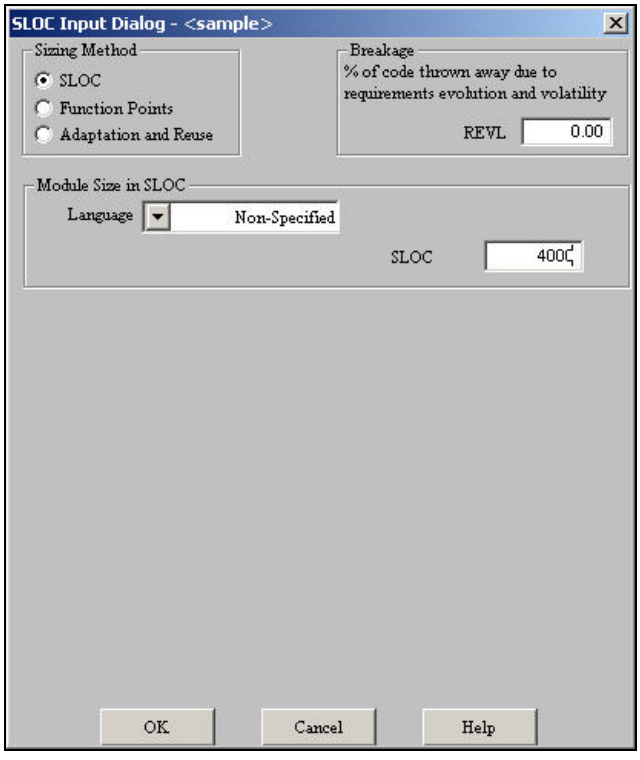


รูปที่ ค.1 หน้าจอภาพโปรแกรมยูเอสซี โคโคโม2

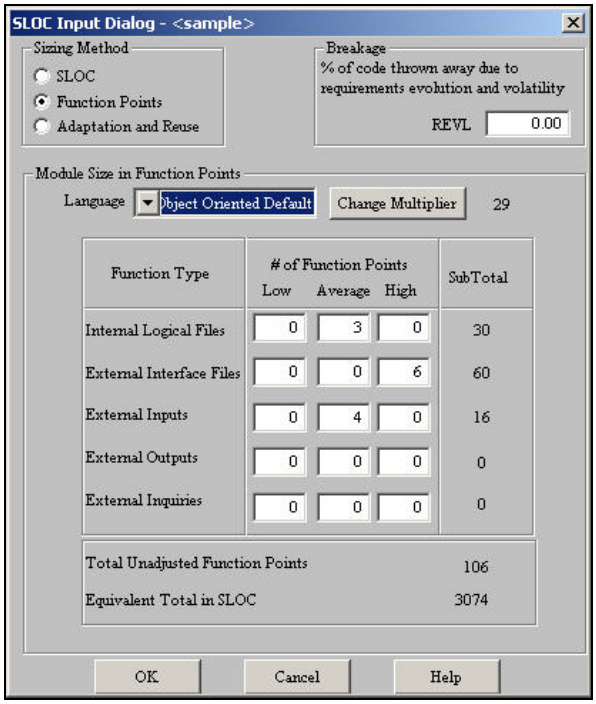
จากรูปที่ ค.1 คือส่วนประกอบของโปรแกรมยูเอสซี โคโคโม2 ผู้ใช้สามารถที่จะใส่ข้อมูลลงไปที่กล่องรับข้อมูลที่เลือกจากเมนูบาร์หรือในคอลัมภ์รับข้อมูล ส่วนประกอบต่างจะอธิบายเป็นข้อๆมาลำดับหมายเลขดังต่อไปนี้

1) แถบเมนูหลัก (Main Menu bar) เป็นเมนูหลักของโปรแกรม ยูเอสซี โคโคโม2 แถบเมนูหลักนี้ จะมีแถบให้เลือกคือ File, View, Edit, Paraments, Calibrate, Phase Distribution และ Help แถบเมนูหลักเหล่านี้จะกล่าวต่อไป หรือสามารถจะเรียนรู้จากแถบเมนู Help

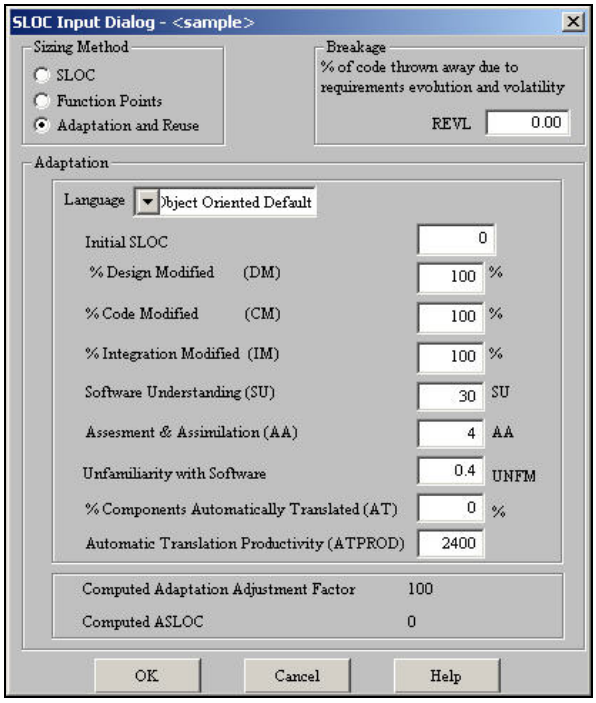
2) แถบเครื่องมือ (Tool bar) เป็นปุ่มแถบเครื่องมือ ทำงานเช่นเดียวกับวินโดว์สแอพพลิเคชันอื่นๆ ประกอบด้วย New Project, Open Project, Save Project, Delete Module, Copy&Paste, Insert Clipboard content , Insert a module, และ About functions
3) ชื่อโครงการ (Project Name) เป็นกล่องข้อความสำหรับใส่ชื่อโครงการ
4) ลำดับของโมดูล (X) เป็นลำดับของโมดูล ซึ่งในหนึ่งโครงการนั้นเราสามารถเพิ่มได้หลายโมดูล
5) คอลัมภ์ชื่อโมดูล (Module Name Column) เป็นคอลัมภ์ใช้สำหรับใส่ชื่อโมดูล
6) คอลัมภ์ขนาดของโมดูล (Module Size [SLOC] Column) เป็นคอลัมภ์สำหรับใส่ขนาดของโมดูลซึ่งสามารถใส่ขนาดของโมดูลได้ 3 รูปแบบประกอบด้วยแบบบรรทัดคำสั่ง (SLOC) แบบฟังก์ชันพอยต์ (Function point) และ แบบจำนวนบรรทัดคำสั่งที่ปรับปรุงและการนำกลับมาใช้ (Adaptation and reuse) แสดงในรูปที่ ค.2 ค.3 และ ค.4



รูปที่ ค.2 กล่องรับขนาดของโมดูลที่เป็นจำนวนบรรทัดคำสั่ง



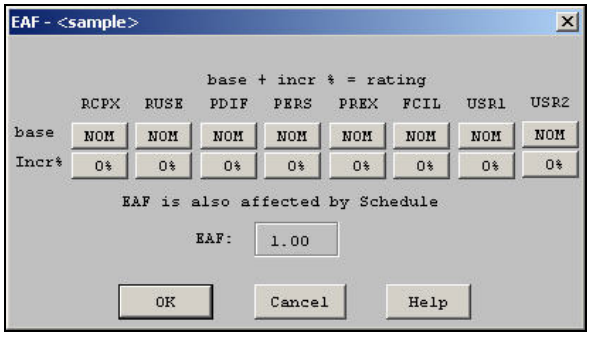
รูปที่ ค.3 กล่องรับขนาดของโมดูลที่เป็นฟังก์ชันพอยต์



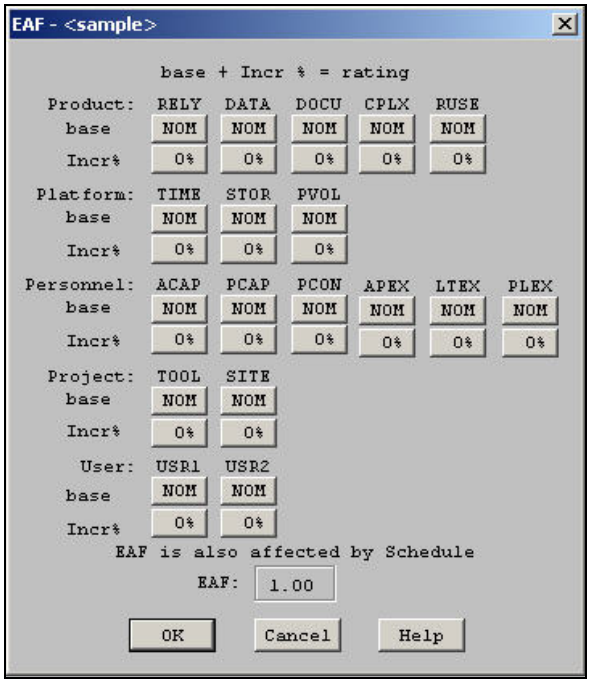
รูปที่ ค.4 กล่องรับขนาดของโมดูลที่เป็นจำนวนบรรทัดคำสั่งที่ปรับปรุงและการนำกลับมาใช้

7) คอลัมภ์ต้นทุนค่าใช้จ่าย (Labor Rate Column) เป็นคอลัมภ์สำหรับใส่ค่าต้นทุนค่าใช้จ่ายต่อนักพัฒนาต่อเดือน

8) คอลัมภ์ค่าตัวคูณความพยายาม (Effort adjustment Factor [EAF] Column) เป็นคอลัมภ์สำหรับการใส่ค่าตัวคูณความพยายาม หรือ Effort Multipliers ที่ประกอบด้วย RELY, DATA, CPLX, RUSE, DOCU, TIME, STOR, PVOL, ACAP, PCAP, APEX, PLXP, LTEX, PCON, TOOL และ SITE แสดงตามรูปที่ ค.5 และ ค.6

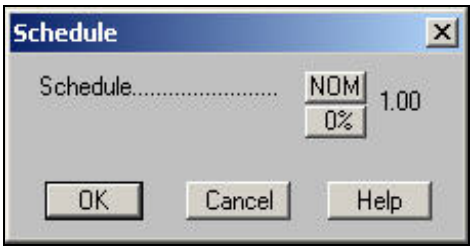


รูปที่ ค.5 กล่องรับค่าตัวคูณความพยายามสำหรับแบบจำลองเออรีดิไซด์



รูปที่ ค.6 กล่องรับค่าตัวคูณความพยายามสำหรับแบบจำลองโพสอาร์คิเท็กเจอร์

9) พื้นที่ผลลัพธ์การคำนวณ (Totals Area) เป็นพื้นที่แสดงผลลัพธ์จากการคำนวณต่างๆ ประกอบด้วย จำนวนบรรทัดคำสั่ง จำนวนความพยายาม ตารางเวลา จำนวนบรรทัดคำสั่งที่ต้องพัฒนาต่อนักพัฒนาต่อหนึ่งเดือน ค่าใช้จ่าย ต้นทุนในการพัฒนาต่อหนึ่งบรรทัดคำสั่ง จำนวนนักพัฒนา และค่าความเสี่ยง
10) สถานะการทำงาน (Status bar) เป็นข้อความที่แสดงสถานะการทำงานของโปรแกรม
11) ปุ่มตารางงาน (Schedule Button) เป็นปุ่มเปิดกล่องรับข้อมูลของตาราเวลา (Schedule) เมื่อคลิกปุ่มจะกล่องรับข้อมูลของตาราเวลาปรากฏขึ้มเพื่อรับค่าตัวคูณความพยายาม ที่เป็นค่า SCED เพื่อที่จะเพิ่มหรือลดตารางเวลาตามการวิเคราะห์ของทีมนักพัฒนา แสดงในรูปที่ ค.7
12) ปุ่มค่าสเกลของปัจจัย (Scale factor button) เป็นปุ่มค่าสเกลของปัจจัย แสดงในรูปที่ ค.8



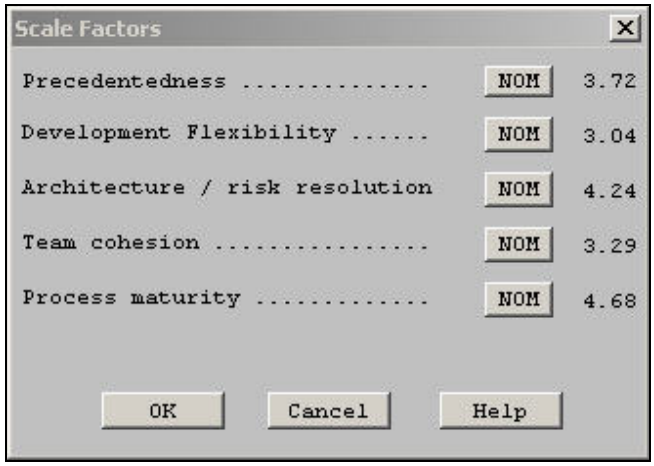
รูปที่ ค.7 กล่องรับข้อมูลของตารางเวลา

13) ปุ่มการเลือกแบบจำลอง (Model Selection button) เป็นปุ่มการเลือกแบบจำลอง โดยจะเลือกได้ทั้งแบบเออรีดิไซด์ และ โพสอาร์คิเท็กเจอร์
14) คอลัมภ์ความเสี่ยงของโครงการ (Risk Column) เป็นระดับความเสี่ยงของโครงการ โดยเมื่อมีการคลิกคอลัมภ์นี้โปแกรมจะแสดงกล่องแสดงความเสี่ยงแบบต่างๆแสดงตามรูปที่ ค.9
15) คอลัมภ์จำนวนนักพัฒนา (Staff Column) เป็นจำนวนนักพัฒนาโครงที่ใช้
16) คอลัมภ์ต้นทุนการพัฒนาต่อหนึ่งบรรทัดคำสั่ง (Instruction Cost Column) เป็นต้นทุนในการพัฒนาต่อหนึ่งบรรทัดคำสั่ง
17) คอลัมภ์ต้นทุน (Cost Column) เป็นต้นทุนในการพัฒนาโมดูลนั้นๆ
18) คอลัมภ์จำนวนบรรทัดคำสั่งที่ต้องพัฒนาต่อนักพัฒนาต่อเดือน (Productivity [PROD] Column) เป็นจำนวนบรรทัดคำสั่งที่ต้องพัฒนาต่อนักพัฒนาต่อเดือน

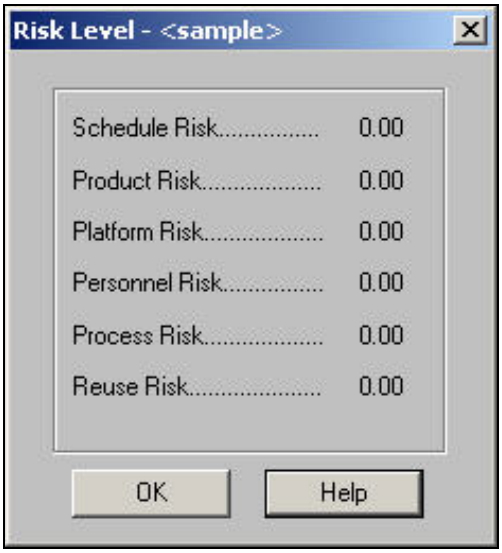
19) คอลัมภ์จำนวนความพยายาม (Estimated Person-month [EST PM] Column) เป็นคอลัมภ์ที่แสดงค่าประมาณจำนวนนักพัฒนาต่อเดือน

20) คอลัมภ์จำนวนความพยายาม (Nominal Person-month [NOM PM] Column) เป็นคอลัมภ์แสดงค่าประมาณจำนวนนักพัฒนาต่อเดือน ที่คำนวนค่าตัวคูณความพยายามในระดับปกติ

21) โปรแกรมภาษา (Languages) เป็นโปรแกรมภาษาที่ทำการประมาณโดยโปรแกรม



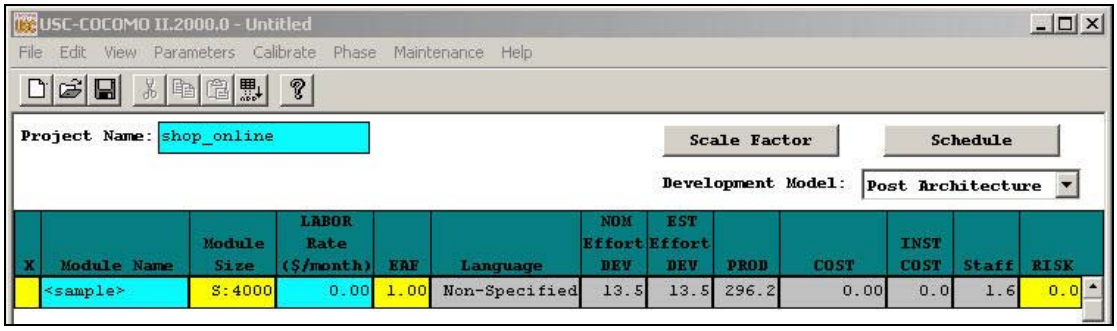
รูปที่ ค.8 กล่องค่าสเกลของปัจจัย



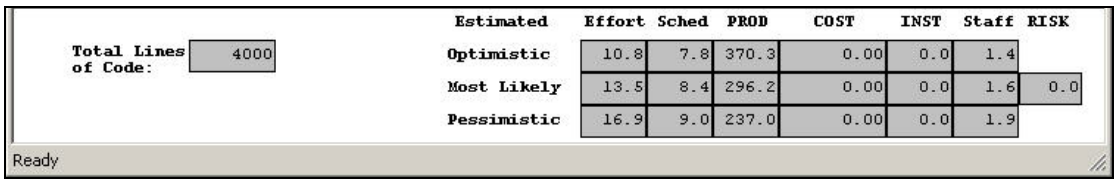
รูปที่ ค.9 กล่องแสดงความเสี่ยง

2. การใช้งานโปรแกรม

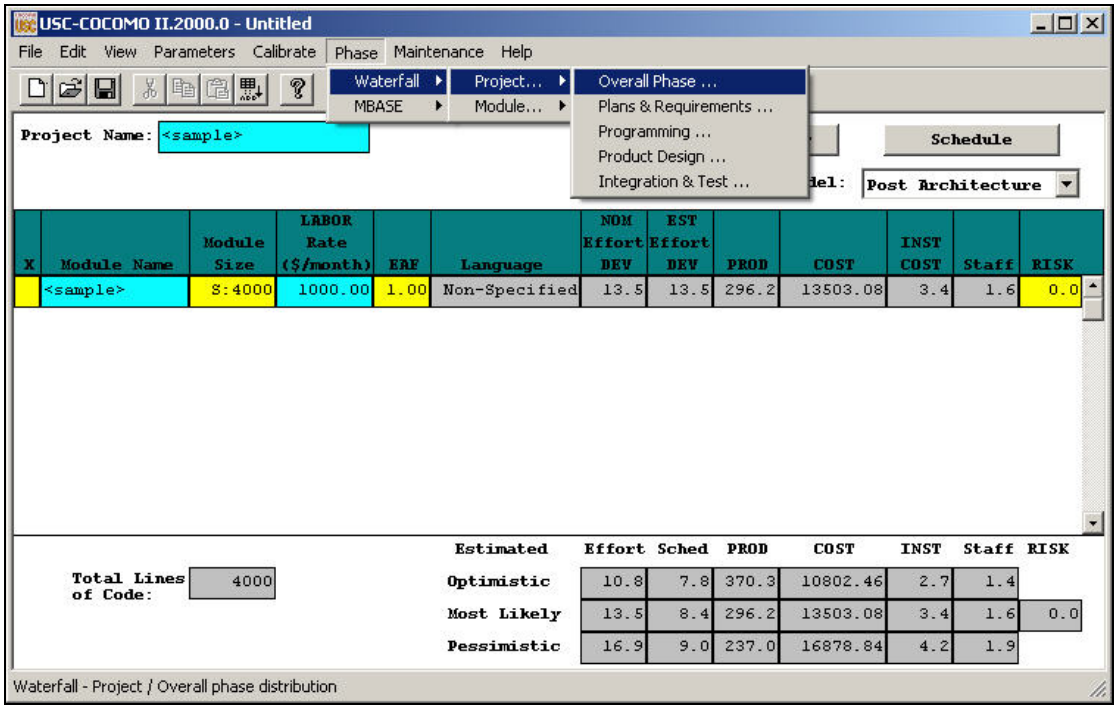
เริ่มด้วยการเปิดโปรแกรมโคโคโม 2 แล้วใส่ชื่อโครงการในช่องชื่อโคราง หลังจากนั้นก็โหลดโมดูลด้วยปุ่ม Add New Module ใส่ชื่อโมดูล ใส่ขนาดโมดูล ใส่ค่าใช้จ่ายต่อนักพัฒนาต่อเดือน (Labor rate) ใส่ค่าตัวคูณความพยายาม (Effort Multipliers) โปรแกรมจะคำนวนค่าต่างๆให้แสดงในส่วนล่างของโปรแกรม โดยโปรแกรมจะประมาณค่าโดยจะแสดงออกมาใน 3 ระดับคือ ค่าผลลัพท์ที่ดีที่สุด ค่าในช่วงปานกลางและ ค่าที่เลวร้ายที่สุด เมื่อมีการประมาณค่าเสร็จ แล้วสามารถดูขั้นตอนการพัฒนาโปรแกรมแบบน้ำตก (waterfal) ซึ่งขั้นตอนการพัฒนาโปรแกรมแบบน้ำตกนี้แบ่งขั้นตอนการพัฒนาเป็น 4 ขั้นตอนคือ การวางแผนและการกำหนดความต้องการ (Plans & Requirements) การออกแบบผลิตภัณฑ์ (Product Design) การเขียนโปรแกรม (Programming) และการรวมโปรแกรมและการทดสอบ (Integration & Test) โดยสามารถจะดูขั้นตอนการพัฒนาทั้งหมด (Overall Phase) หรือสามารถดูเป็นรายละเอียดในแต่ละขั้นตอนการพัฒนา โดยเลือกจากแถบเมนู Phase แสดงตามรูปที่ ค.10 ถึง รูปที่ ค.17



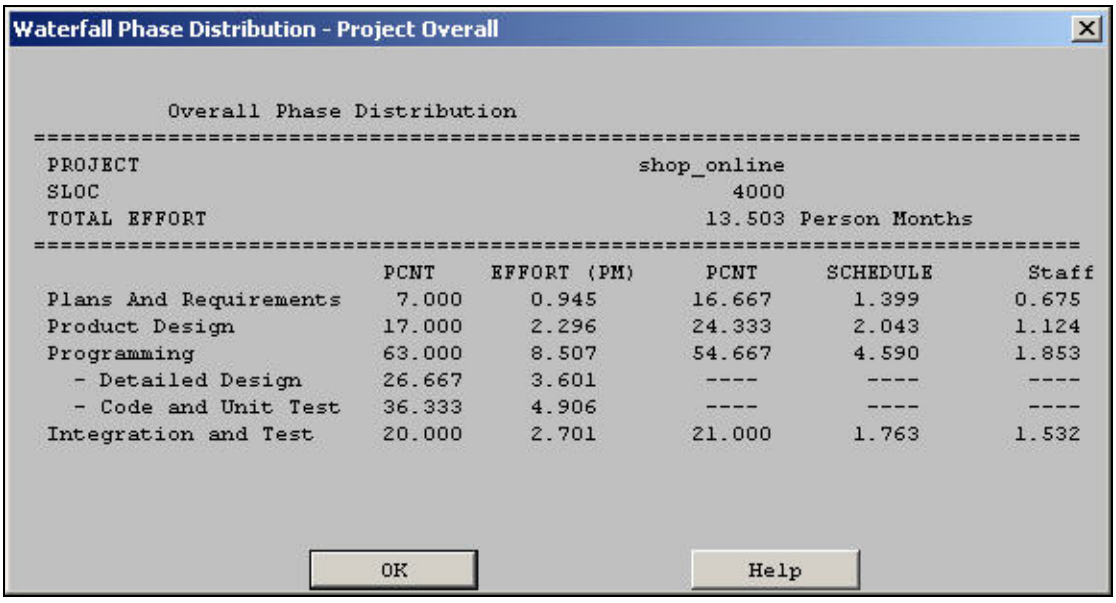
รูปที่ ค.10 การสร้างโมดูลและการใส่ค่าจำนวนบรรทัดคำสังและค่าใช้จ่าย



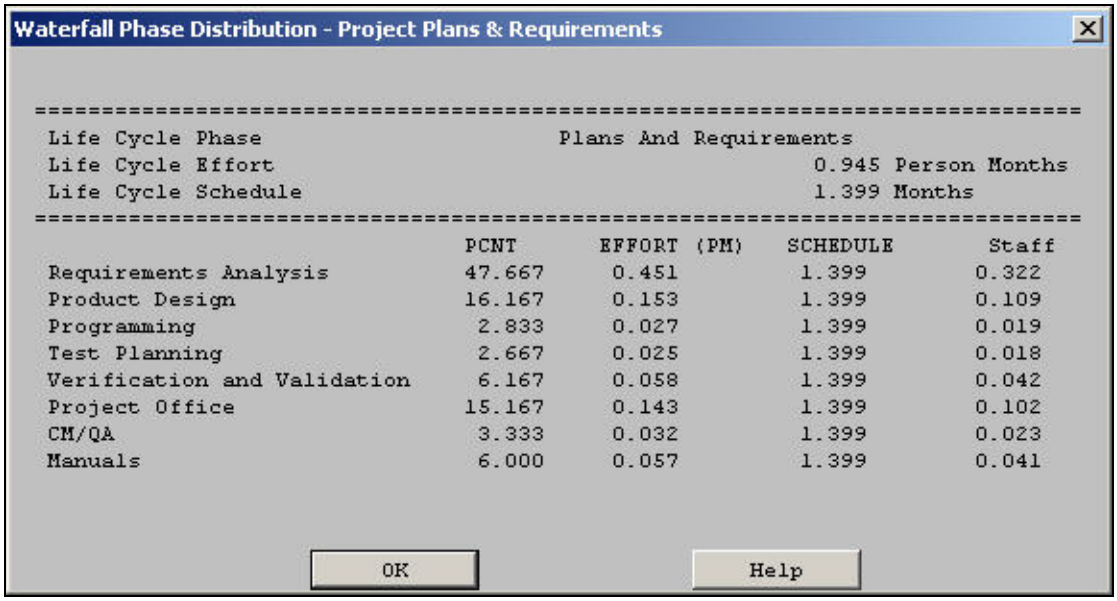
รูปที่ ค.11 พื้นที่ผลลัพธ์การคำนวณ



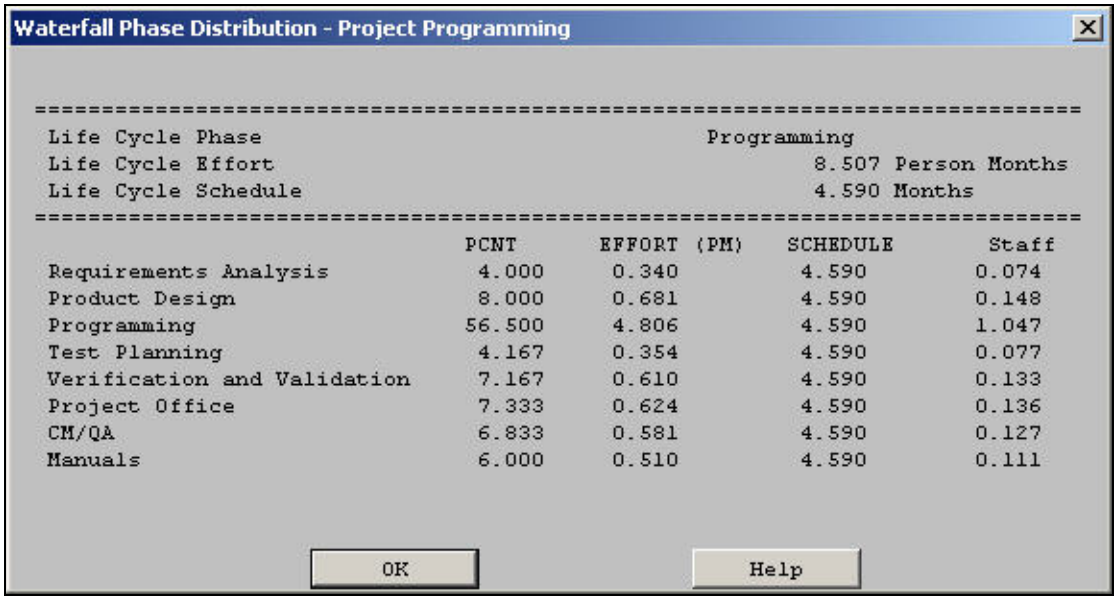
รูปที่ ค.12 เมนูแสดงขั้นตอนพัฒนาโปรแกรมแบบน้ำตก



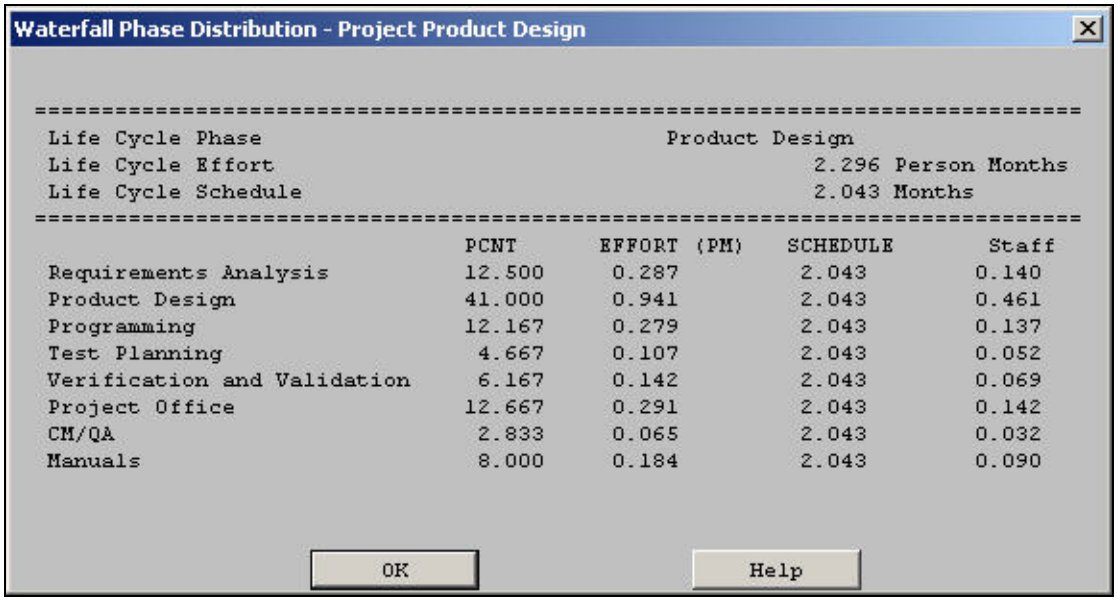
รูปที่ ค.13 แสดงขั้นตอนพัฒนาโปรแกรมแบบน้ำตกในทุกช่วง



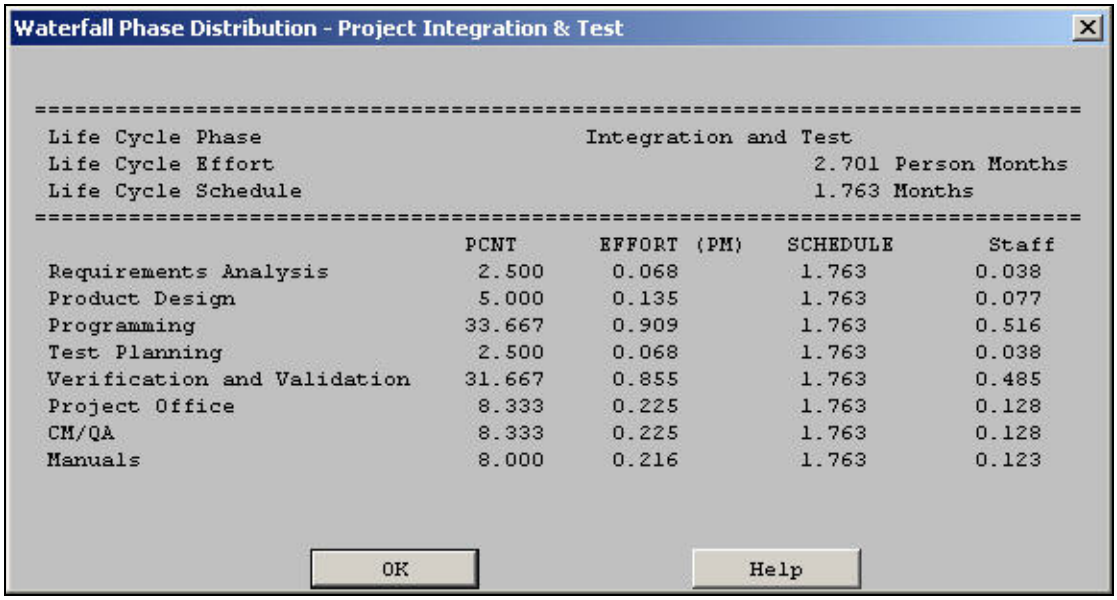
รูปที่ ค.14 แสดงขั้นตอนพัฒนาในช่วงการวางแผนและการกำหนดความต้องการ



รูปที่ ค.15 แสดงขั้นตอนพัฒนาในช่วงการออกแบบผลิตภัณฑ์



รูปที่ ค.16 แสดงขั้นตอนพัฒนาในช่วงการเขียนโปรแกรม



รูปที่ ค.17 แสดงขั้นตอนพัฒนาในช่วงการรวมโปรแกรมและการทดสอบ

ภาคผนวกเป็นการอธิบายการใช้งานโปรแกรมโดยสังเขป ศึกษาข้อมูลเพิ่มเติมได้ที่ http://sunset.use.edu/COCOMOII/cocomo.html

ภาคผนวก ง

เอกสารความต้องการของระบบแจ้งการเดินการออนไลน์

1. วัตถุประสงค์

วัตถุประสงค์ของเอกสารนี้จะเป็นการอธิบายความต้องการของระบบแจ้งการเดินทางออนไลน์ ซึ่งได้จากการรวบรวมข้อมูลจากเอกสาร แบบสอบถาม การสัมภาษณ์ และ การสังเกตระบบแจ้งการเดินทางของอาจารย์พิเศษที่เดินทางมาทำการสอน ณ มหวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร โดยเอกสารความต้องการของระบบแจ้งการเดินทางออนไลน์ฉบับนี้ จะกำหนดกรอบของระบบที่จะพัฒนาเป็นซอฟ์ทแวร์ และที่สำคัญที่สุดเอกสารความต้องการระบบฉบับนี้ไม่ได้บอกความเป็นจริงของระบบทั้งหมดซึ่งจะต้องรวบรวมข้อมูลเพิ่มเติมเมื่อถึงขั้นตอนการออกแบบและการเขียนโปรแกรม

1.1 ขอบข่ายระบบ

ปัจจุบันระบบแจ้งการเดินทางออนไลน์ผ่านเว็บได้มีการใช้งานในสำนักงานหลายแห่งแล้วเพื่อทำการติดระหว่างสำนักงานที่มีสถานที่ตั้งอยู่ห่างกันแทนการติดต่อทางโทรศัพท์ เพื่อลดค่าใช้จ่ายและความถูกต้องของข้อมูล ซึ่งระบบแจ้งการเดินทางออนไลน์จะประกอบด้วยเครื่องแม่ข่ายคอมพิวเตอร์ คอมพิวเตอร์ส่วนบุคคล และระบบเครือข่าย ระบบแจ้งการเดินทางออนไลน์จะทำการรับแจ้งการเดินทางของอาจารย์พิเศษจากเจ้าหน้าที่ที่ดูแลการเดินทางของอาจารย์พิเศษรายงานต่อเจ้าหน้าของวิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร และเจ้าหน้าที่ส่วนกลางในวิทยาเขตบางเขต เพื่อทราบกำหนดการเดินทางของอาจาร์พิเศษ

1.2 ข้อกำหนด

1) ระบบแจ้งการเดินทางออนไลน์ คือระบบที่นำเครื่องแม่ข่ายคอมพิวเตอร์และโปรแกรมแจ้งการเดินทางรับแจ้งการเดินทางของอาจารย์พิเศษ จัดเก็บข้อมูล และแสดงรายงานการเดินทาง
2) โปรแกรมแจ้งการเดินทาง คือโปรแกรมเว็บที่รับข้อมูลการเดินทาง จักเก็บข้อมูลและแสดงรายการงานเดินทาง โดยการใช้งานโปรแกรมจะใช้งานผ่านโปรแกรมเว็บเบราว์เซอร์บนระบบเครือข่ายอินเตอร์เน็ต
3) เครื่องแม่ข่ายคอมพิวเตอร์ คือเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการที่สามารถรองรับการใช้งานโปรแกรมเว็บได้
4) เครื่องคอมพิวเตอร์ คือเครื่องคอมพิวเตอร์ส่วนบุคคลที่มีการเชื่อมต่อระบบอินเตอร์เน็ตโดยจะเชื่อมต่อจากสถานที่ใดก็ได้

2. คุณสมบัติของระบบ

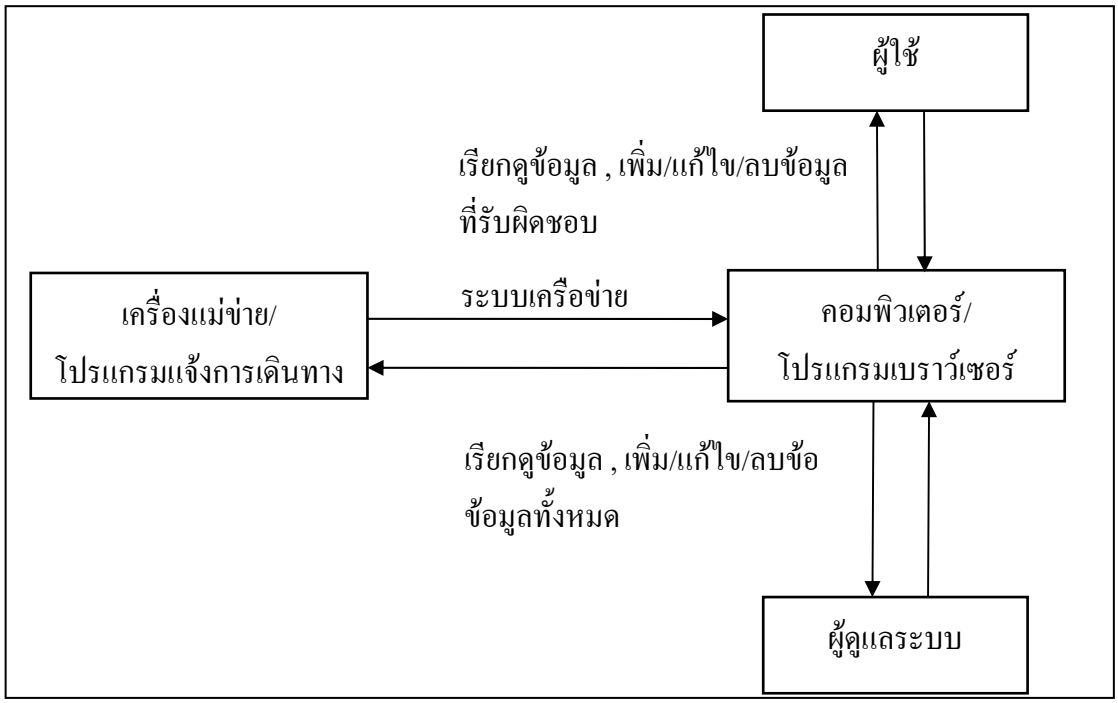
2.1 รูปแบบของระบบ

1) ระบบแจ้งการเดินทางออนไลน์ คือระบบที่นำเครื่องแม่ข่ายคอมพิวเตอร์และโปรแกรมแจ้งการเดินทาง ทำหน้าที่รับแจ้งการเดินทางของอาจารย์พิเศษผ่านเว็บ จัดเก็บข้อมูลลงในฐานข้อมูล และแสดงรายงานการเดินทางผ่านเว็บไปยังคอมพิวเตอร์ที่เชื่อมต่อระบบอินเตอร์เน็ต โดยมีส่วนประกอบดังนี้

2) โปรแกรมแจ้งการเดินทาง เป็นโปรแกรมเว็บที่รับข้อมูลการเดินทาง จัดเก็บลงฐานข้อมูล จัดเก็บข้อมูลพื้นฐานที่จำเป็นต่อการแจ้งการเดินทาง แสดงรายงานการเดินทางตามความต้องการของผู้ใช้ และ ลบหรือแก้ไขข้อมูลจากผู้ใช้หรือผู้ดูแล

2.2 โครงสร้างของระบบ

ระบบแจ้งการเดินทางออนไลน์ สามารถเขียนเป็นแผนผังโครงสร้างได้ดังรูปที่ ง.1



รูปที่ ง.1 แผนผังโครงสร้างระบบแจ้งการเดินทางออนไลน์ (State Diagram)

จากโครงสร้างของระบบแจ้งการเดินทางออนไลน์รูปที่ 1 โปรแกรมแจ้งการเดินทางและข้อมูลพื้นฐานจะถูกติดตั้งในเครื่องแม่ข่ายเชื่อต่อโดยระบบเครือข่ายไปยังเครื่องคอมพิวเตอร์ในมหาวิทยาลัยหรือภายนอกที่ติดตั้งโปรแกรมเบราว์เซอร์ โดยผู้ใช้และผู้ดูแลระบบจะใช้งานโปรแกรมแจ้งการเดินทางผ่านเว็บเบราว์เซอร์

2.3 คุณสมบัติผู้ใช้งาน

แจ้งการเดินทางออนไลน์จะแบ่งอยู่ใช้งานออกเป็น 2 ระดับด้วยกันคือ ผู้ดูแลระบบและผู้ใช้ทั่วไป

1) ผู้ดูแลระบบเป็นผู้ที่ควบคุมดูแลโปรแกรมทั้งหมดของระบบ เพิ่ม แก้ไข และลบข้อมูลเดินทาง และข้อมูลพื้นฐาน
2) ผู้ใช้ทั่วไปเป็นเจ้าหน้าที่ที่รับผิดชอบดูแลการเดินทางของอาจารย์พิเศษ สามารถเพิ่ม แก้ไข และลบ ข้อมูลเดินทาง และข้อมูลพื้นฐานที่เป็นข้อมูลของอาจารย์พิเศษ

2.4 คุณสมบัติของอุปกรณ์ในระบบ

1) เครื่องแม่ข่ายคอมพิวเตอร์มีระบบปฏิบัติการที่รองรับการใช้งานโปรแกรมภาษาพีเอชพีและฐานข้อมูลมายเอสคิวแอล มีการเชื่อมต่อระบบเครือข่ายของมหาวิทยาลัย
2) เครื่องคอมพิวเตอร์ เป็นเครื่องคอมพิวเตอร์ส่วนบุคคลที่มีการเชื่อมต่อระบบอินเตอร์เน็ตและติดตั้งโปรแกรมเบราว์เซอร์

3. ความต้องการของระบบเจ้าการเดินออนไลน์

3.1 ความต้องการที่เป็นฟังก์ชัน

ก01 โปรแกรมแจ้งการเดินทาง จะต้องสามารถเพิ่ม/แก้ไข/ลบข้อมูลการเดินทางและข้อมูลพื้นฐาน สำหรับผู้ใช้และผู้ดูแลระบบ

ก01-01 การเพิ่มข้อมูลสำหรับผู้ใช้สามารถเพิ่มข้อมูลการเดินทางและข้อมูลพื้นฐานที่เป็นข้อมูลของอาจารย์พิเศษ ส่วนผู้ดูแลระบบสามารถเพิ่มข้อมูลได้ทั้งหมด

ก01-02 การแก้ไขข้อมูลสำหรับผู้ใช้สามารถแก้ไขข้อมูลการเดินทางและข้อมูลพื้นฐานที่เป็นข้อมูลของอาจารย์พิเศษในหน่วยงานที่ดูแล ส่วนผู้ดูแลระบบสามารถแก้ไขข้อมูลได้ทั้งหมด

ก01-03 การลบข้อมูลสำหรับผู้ใช้สามารถลบข้อมูลการเดินทางและข้อมูลพื้นฐานที่เป็นข้อมูลของอาจารย์พิเศษในหน่วยงานที่ดูแล ส่วนผู้ดูแลระบบสามารถลบข้อมูลได้ทั้งหมด

ก01-04 การเพิ่มและแก้ไขข้อมูลการเดินทางสามารถเลือกรายชื่ออาจารย์พิเศษและรายการการเดินทางได้จากลิสต์รายการ

ข01 โปรแกรมแจ้งการเดินทางสามารถแสดงข้อมูลการเดินทาง ได้ทั้งแบบย่อแบบสมบูรณ์ที่สามารถลิงค์ไปยังการแก้ไขและลบข้อมูลได้ และ 20 รายการล่าสุด

3.2 ความต้องการติดต่อกับผู้ใช้งาน

จะต้องเป็นโปรแกรมเว็บที่ใช้งานง่ายไม่เน้นการออกแบบกราฟิกส์ สามารถค้นหาการเดินทางตามช่วงเวลาได้

3.3 ความต้องของระบบฐานข้อมูล

ง01 ระบบแจ้งการเดินทางออนไลน์จะบันทึกข้อมูลลงในฐานข้อมูลมายเอสคิวแอลบนเครื่องแม่ข่าย รายละเอียดของฐานข้อมูลแสดงตามภาคผนวก

4. ความต้องการของระบบที่ไม่เป็นฟังก์ชัน

4.1 ความปลอดภัยของระบบ

1) ระบบแจ้งการเดินทางออนไลน์จะต้องมีการกำหนดระบบรหัสผ่าน
2) ระบบแจ้งการเดินทางออนไลน์จะต้องกำหนดลำดับสิทธิของผู้ใช้ในการเข้าใช้ระบบ

4.2 ความต้องการของเครื่องคอมพิวเตอร์

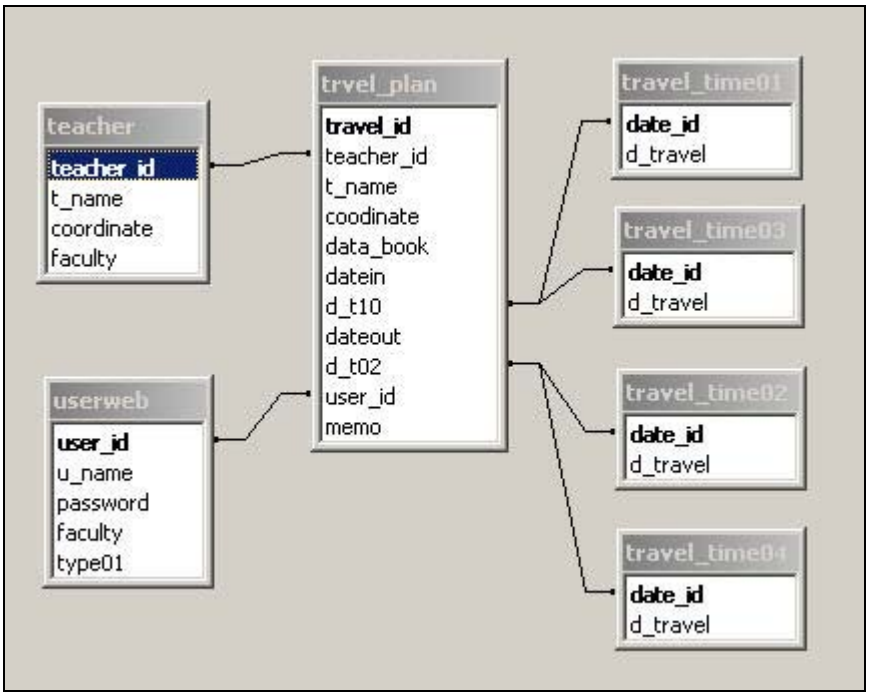
1) เครื่องแม่ข่ายคอมพิวเตอร์เป็นเครื่องแม่ข่ายของทางมหาวิทยาลัย
2) เครื่องคอมพิวเตอร์เป็นเครื่องคอมพิวเตอร์สำนักงานทั่วไปไม่ระบุคุณสมบัติของเครื่อง

4.3. การติดตามและการบำรุงรักษาระบบ

1) มีการอบรมการใช้งานโปรแกรม
2) มีการเข้าบำรุงรักษาทุกเดือนเป็นระยะ 1 ปี และสามารถติดต่อให้คำปรึกษาทางโทรศัพท์ได้

5. ภาคผนวก

5.1 แผนผังฐานข้อมูลโปรแกรมแจ้งการเดินทาง



รูปที่ ง.2 แผนผังฐานข้อมูลโปรแกรมแจ้งการเดินทาง

5.2 ฐานข้อมูลโปรแกรมแจ้งการเดินทาง

ฐานข้อมูลประกอบด้วย 7 ตาราง

1) ตาราง travel_plant เป็นข้อมูลการเดินทางอาจารย์พิเศษ
2) ตาราง teacher01 เป็นข้อมูลพื้นฐานอาจารย์พิเศษ
3) ตาราง travel_time01 เป็นข้อมูลพื้นฐานการเดินทาง01
4) ตาราง travel_time02 เป็นข้อมูลพื้นฐานการเดินทาง02
5) ตาราง travel_time03 เป็นข้อมูลพื้นฐานการเดินทาง03
6) ตาราง travel_time04 เป็นข้อมูลพื้นฐานการเดินทาง03
7) ตาราง userweb เป็นข้อมูลพื้นฐานผู้ใช้

ภาคผนวก จ

ตารางฟังก์ชันพอยต์ของระบบแจ้งการเดินการออนไลน์

จากเอกสารความต้องการของโปรแกรมแจ้งการเดินทาง ในภาคผนวก ง สามารถที่จะนับออกมาเป็นฟังก์ชันพอยต์ได้ตามตารางที่ จ.1

ตารางที่ จ.1 ตารางจำนวนฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมแจ้งการเดินทาง

คำอธิบาย (อ้างอิงเอกสารความต้องการ)	Type	DETs	RETs/FTRs	Complexity	Weight
เข้าสู่ระบบ (4.1)	EI	3	1	L	3
มัลติมีเดียไฟล์ (jpg) (3.2)	MMF	1	0	L	1
ลิสต์อาจารย์ (ก01-04)	EQ	4	1	L	3
ลิสต์การเดินทาง01 (ก01-04)	EI	1	1	L	3
ลิสต์การเดินทาง02 (ก01-04)	EI	1	1	L	3
ลิสต์การเดินทาง03 (ก01-04)	EI	1	1	L	3
ลิสต์การเดินทาง04 (ก01-04)	EI	1	1	L	3
เพิ่มข้อมูลการเดินทาง (ก01-01)	EI	11	1	L	3
รายการแก้ไขข้อมูลการเดินทาง (ก01-02)	EQ	10	1	L	3
แก้ไขข้อมูลการเดินทาง (ก01-02)	EI	11	1	L	3
รายการลบข้อมูลการเดินทาง (ก01-03)	EQ	10	1	L	3
ลบข้อมูลการเดินทาง (ก01-03)	EI	11	1	L	3
แสดงข้อมูลการเดินทาง แบบย่อ(ข01)	EQ	10	1	L	3
แสดงข้อมูลการเดินทาง แบบสมบูรณ์ (ข01)	EQ	10	1	L	3
แสดงข้อมูลการเดินทาง 20 รายการล่าสุด (ข01)	EQ	5	1	L	3
รายการก่อนเพิ่มข้อพื้นฐานของข้อมูลอาจารย์ (ก01-01)	EQ	4	1	L	3
เพิ่มข้อพื้นฐานของข้อมูลอาจารย์ (ก01-01)	EI	5	1	L	3
รายการก่อนแก้ไขข้อพื้นฐานของข้อมูลอาจารย์ (ก01-02)	EQ	4	1	L	3
แก้ไขข้อพื้นฐานของข้อมูลอาจารย์ (ก01-02)	EI	4	1	L	3
รายการก่อนลบข้อพื้นฐานของข้อมูลอาจารย์ (ก01-03)	EQ	4	1	L	3
ลบข้อพื้นฐานของข้อมูลอาจารย์ (ก01-03)	EI	1	1	L	3
รายการก่อนเพิ่มข้อพื้นฐานของข้อมูลการเดินทาง01 (ก01-01)	EQ	2	1	L	3
เพิ่มข้อพื้นฐานของข้อมูลการเดินทาง01 (ก01-01)	EI	3	1	L	3
รายการก่อนแก้ไขข้อพื้นฐานของข้อมูลการเดินทาง01 (ก01-02)	EQ	2	1	L	3
แก้ไขข้อพื้นฐานของข้อมูลการเดินทาง01 (ก01-02)	EI	3	1	L	3
รายการก่อนลบข้อพื้นฐานของข้อมูลการเดินทาง01 (ก01-03)	EQ	2	1	L	3
ลบข้อพื้นฐานของข้อมูลการเดินทาง01 (ก01-03)	EI	1	1	L	3
รายการก่อนเพิ่มข้อพื้นฐานของข้อมูลการเดินทาง02 (ก01-01)	EQ	2	1	L	3
เพิ่มข้อพื้นฐานของข้อมูลการเดินทาง02 (ก01-01)	EI	3	1	L	3
รายการก่อนแก้ไขข้อพื้นฐานของข้อมูลการเดินทาง02 (ก01-02)	EQ	2	1	L	3
แก้ไขข้อพื้นฐานของข้อมูลการเดินทาง02 (ก01-02)	EI	3	1	L	3
รายการก่อนลบข้อพื้นฐานของข้อมูลการเดินทาง02 (ก01-03)	EQ	2	1	L	3

ตารางที่ จ.1 ตารางจำนวนฟังก์ชันพอยต์ร่วมกับส่วนประกอบเว็บของโปรแกรมแจ้งการเดินทาง (ต่อ)

คำอธิบาย (อ้างอิงเอกสารความต้องการ)	Type	DETs	RETs/FTRs	Complexity	Weight
ลบข้อพื้นฐานของข้อมูลการเดินทาง02 (ก01-03)	EI	1	1	L	3
รายการก่อนเพิ่มข้อพื้นฐานของข้อมูลการเดินทาง03 (ก01-01)	EQ	2	1	L	3
เพิ่มข้อพื้นฐานของข้อมูลการเดินทาง03 (ก01-01)	EI	3	1	L	3
รายการก่อนแก้ไขข้อพื้นฐานของข้อมูลการเดินทาง03 (ก01-02)	EQ	2	1	L	3
แก้ไขข้อพื้นฐานของข้อมูลการเดินทาง03 (ก01-02)	EI	3	1	L	3
รายการก่อนลบข้อพื้นฐานของข้อมูลการเดินทาง03 (ก01-03)	EQ	2	1	L	3
ลบข้อพื้นฐานของข้อมูลการเดินทาง03 (ก01-03)	EI	1	1	L	3
รายการก่อนเพิ่มข้อพื้นฐานของข้อมูลการเดินทาง04 (ก01-01)	EQ	2	1	L	3
เพิ่มข้อพื้นฐานของข้อมูลการเดินทาง04 (ก01-01)	EI	3	1	L	3
รายการก่อนแก้ไขข้อพื้นฐานของข้อมูลการเดินทาง04 (ก01-02)	EQ	2	1	L	3
แก้ไขข้อพื้นฐานของข้อมูลการเดินทาง04 (ก01-02)	EI	3	1	L	3
รายการก่อนลบข้อพื้นฐานของข้อมูลการเดินทาง04 (ก01-03)	EQ	2	1	L	3
ลบข้อพื้นฐานของข้อมูลการเดินทาง04 (ก01-03)	EI	1	1	L	3
รายการก่อนเพิ่มข้อพื้นฐานของข้อมูลผู้ใช้ (ก01-01)	EQ	5	1	L	3
เพิ่มข้อพื้นฐานของข้อมูลผู้ใช้ (ก01-01)	EI	6	1	L	3
รายการก่อนแก้ไขข้อพื้นฐานของข้อมูลผู้ใช้ (ก01-02)	EQ	5	1	L	3
แก้ไขข้อพื้นฐานของข้อมูลผู้ใช้ (ก01-02)	EI	6	1	L	3
รายการก่อนลบข้อพื้นฐานของข้อมูลผู้ใช้ (ก01-03)	EQ	5	1	L	3
ลบข้อพื้นฐานของข้อมูลผู้ใช้ (ก01-03)	EI	1	1	L	3
ตาราง teacher01 (ง01)	ILF	4	1	L	7
ตาราง travel_plant (ง01)	ILF	11	1	L	7
ตาราง travel_time01 (ง01)	ILF	2	1	L	7
ตาราง travel_time02 (ง01)	ILF	2	1	L	7
ตาราง travel_time03 (ง01)	ILF	2	1	L	7
ตาราง travel_time04 (ง01)	ILF	2	1	L	7
ตาราง userweb (ง01)	ILF	5	1	L	7

ประวัติผู้เขียน

นายจักรนรินทร์ คงเจริญ รหัสประจำตัวนักศึกษา M4740063 มีภูมิลำเนาอยู่จังหวัดสกนคร เกิดเมื่อวันที่ 11 ตุลาคม 2519 สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณบิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่นเมื่อปี พ.ศ. 2543 และในปีเดียวกันนั้นได้เข้าทำงานที่สำนักวิทยบริการ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร ในตำแหน่งนักวิชาการคอมพิวเตอร์ รับผิดชอบเป็นผู้ดูแลเว็บและพัฒนาโปรแกรม จากนั้นย้ายมาสังกัดคณะวิทยาศาสตร์และวิศวกรรมศาสตร์ในตำแหน่งเดียวกัน และทำหน้าที่เป็นผู้ช่วยสอนไปพร้อมกันด้วย รับผิดชอบสอนในวิชาปฏิบัติการ ผู้ดูแลเว็บ และเป็นวิทยากรอบรมโปรแกรมคอมพิวเตอร์ต่างๆ หลังจากนั้นได้รับทุนพัฒนาบุคคลากรของวิทยาเขต เข้าศึกษาในระดับปริญญาโท ณ สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2547