# การเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการ พัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินใจของผู้เชี่ยวชาญ (Expert Judgment) และ วิธีตัวแบบขั้นตอนวิธี (Algorithmic Model) ภายใต้สภาวะแวดล้อมในประเทศไทย

นายชานน จรัสสุทธิกุล

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาการพัฒนาซอฟต์แวร์ด้านธุรกิจ ภาควิชาสถิติ
คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# THE ACCURACY COMPARISON OF SOFTWARE COST ESTIMATION BETWEEN EXPERT JUDGMENT AND ALGORITHMIC MODEL IN THAILAND.

Mr. Chanon Jaratsuttikul

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Business Software Development
Department of Statistics
Faculty of Commerce and Accountancy
Chulalongkorn University
Academic Year 2018

Copyright of Chulalongkorn University

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการ พัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินใจของผู้เชี่ยวชาญ (Expert Judgment) และ วิธีตัวแบบขั้นตอนวิธี (Algorithmic Model) ภายใต้สภาวะแวดล้อมในประเทศไทย |
| โดย | นายชานน จรัสสุทธิกุล |
| สาขาวิชา | การพัฒนาซอฟต์แวร์ด้านธุรกิจ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | รองศาสตราจารย์ ดร.ถาวร อานุภาพไตรรงค์ |

คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะพาณิชยศาสตร์และการบัญชี
(รองศาสตราจารย์ ดร.พสุ เดชะรินทร์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.อัษฎาพร ทรัพย์สมบูรณ์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ถาวร อานุภาพไตรรงค์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วัฒนา วิริยสิทธาวัฒน์)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.วรลักษณ์ วงศ์โดยหวัง ศิริเจริญ)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ชานน จรัสสุทธิกุล : การเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการ พัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินใจของผู้เชี่ยวชาญ (Expert Judgment) และวิธีตัวแบบขั้นตอนวิธี (Algorithmic Model) ภายใต้สภาวะแวดล้อมในประเทศไทย. ( THE ACCURACY COMPARISON OF SOFTWARE COST ESTIMATION BETWEEN EXPERT JUDGMENT AND ALGORITHMIC MODEL IN THAILAND.) อ.ที่ปรึกษาหลัก : รศ. ดร.ถาวร อานุภาพไตรรงค์

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่มีความคลาดเคลื่อนสูง อาจส่งผลให้ต้นทุนการจัดการโครงการ และเวลาที่ใช้ในการทำโครงการซอฟต์แวร์เพิ่มขึ้น ผู้วิจัยจึงได้ทำการทดลองเพื่อศึกษาและเปรียบเทียบความแม่นยำการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีประเมินที่ได้รับความนิยม 2 วิธี คือ วิธีตัวแบบขั้นตอนวิธี และวิธีการตัดสินของผู้เชี่ยวชาญ สำหรับวิธีตัวแบบขั้นตอนวิธี ในปัจจุบันมีตัวแบบให้เลือกใช้อยู่เป็นจำนวนมาก ผู้วิจัยได้เลือกใช้เฉพาะตัวแบบโคโคโม่ ฟังก์ชันพอยต์ และยูสเคสพอยต์ เนื่องจากเป็นตัวแบบที่ได้รับความนิยมสูงในการใช้งาน และสำหรับวิธีการตัดสินของผู้เชี่ยวชาญ ผู้วิจัยได้เชิญผู้ประเมินที่เป็นผู้ที่มีประสบการณ์ในการพัฒนาโครงการซอฟต์แวร์ไม่น้อยกว่า 10 ปี จำนวน 5 คน มาเป็นผู้ประเมิน โดยการทดลองในงานวิจัยนี้จะใช้โครงการซอฟต์แวร์จำนวน 10 โครงการ ซึ่งเป็นโครงการซอฟต์แวร์ที่ได้รับการพัฒนาเสร็จแล้ว แม้ผลการทดลองจะแสดงให้เห็นว่าการประเมินด้วยตัวแบบยูสเคสพอยต์มีความแม่นยำมากกว่าวิธีการตัดสินของผู้เชี่ยวชาญ แต่วิธีการตัดสินของผู้เชี่ยวชาญก็แม่นยำมากกว่าตัวแบบฟังก์ชันพอยต์ และ โคโคโม่ ดังนั้นจึงไม่สามารถสรุปได้ว่าวิธีใดมีความแม่นยำมากกว่ากัน ระหว่างวิธีตัวแบบขั้นตอนวิธี และวิธีการตัดสินของผู้เชี่ยวชาญ นอกจากนี้ผู้วิจัยได้ทำการทดลองเพื่อทดสอบว่าหากผู้เชี่ยวชาญได้เห็นผลลัพธ์จากตัวแบบขั้นตอนวิธี และนำค่าดังกล่าวมาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ผู้เชี่ยวชาญจะเปลี่ยนแปลงค่าประเมินหรือไม่และค่าประเมินที่เปลี่ยนแปลงนั้นจะช่วยให้การประเมินแม่นยำขึ้นหรือไม่ ผลการทดลองสรุปได้ว่าผู้เชี่ยวชาญมีการเปลี่ยนแปลงค่าประเมินประมาณครึ่งหนึ่งของจำนวนการทดลองทั้งหมดและการเห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีไม่ได้ช่วยให้การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญแม่นยำขึ้นอย่างมีนัยสำคัญ

สาขาวิชา การพัฒนาซอฟต์แวร์ด้านธุรกิจ ลายมือชื่อนิสิต .....

ปีการศึกษา 2561 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

# # 5981509726 : MAJOR BUSINESS SOFTWARE DEVELOPMENT
KEYWORD: software cost estimation, algorithmic model, expert judgment, cocomo, function point, use case point

Chanon Jaratsuttikul : THE ACCURACY COMPARISON OF SOFTWARE COST ESTIMATION BETWEEN EXPERT JUDGMENT AND ALGORITHMIC MODEL IN THAILAND.. Advisor: Assoc. Prof. Tharwon Arnuphaptrairong, Ph.D.

Software cost estimation with very high errors may affect and cause the increasing in cost of software management and time spent on a software project. Therefore, the author conducted an experiment in order to study and compare the accuracy of two popular software estimation methods –- algorithmic model and expert judgment method. Since there are many algorithmic models available in the literature, only three algorithmic models were included in this experiment -- COCOMO, Function point, and Use case point, because they are widely used models. In this experiment, 5 experienced experts who have experience in software development more than 10 years were asked to estimate 10 completed software projects. It is found that Use case point method is the most accurate but Function point and COCOMO estimates are less accurate than expert judgment estimates. It is therefore no clear conclusion whether one method is better than the other. In addition, the author also extended the experiment to investigate the effects of displaying the estimates from the three models to the experts. The results from comparing the accuracy between expert judgment estimates before and after seeing the results from an algorithmic models show that experts changed their results for half of the experiments. However the accuracy of expert judgment estimates were not enhanced.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| | | |
|---|---|---|
| Field of Study: | Business Software Development | Student's Signature ..... |
| Academic Year: | 2018 | Advisor's Signature ..... |

# กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงได้ ต้องขอขอบพระคุณ อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.ถาวร อานุภาพไตรรงค์ ที่ช่วยให้คำปรึกษาและตรวจข้อผิดพลาดที่มีปริมาณมหาศาลได้ ขอขอบพระคุณรองศาสตราจารย์ ดร.อัษฎาพร ทรัพย์สมบูรณ์ ที่ช่วยให้คำแนะนำในการปิดช่องโหว่ของงานวิจัยนี้ ขอขอบพระคุณผู้ช่วยศาสตราจารย์ วัฒนา วิริยสิทธาวัฒน์ ที่ช่วยตรวจสอบยูสเคสไดอะแกรม และคอยช่วยเหลือในตอนสอบจบ ขอขอบพระคุณกรรมการภายนอก

ขอบคุณนางสาวสกลพร โนรี นายชานน วิโรจนานุวัฒน์และพี่นิสิต ที่อุตส่าห์สละเวลาอันมีค่ามาช่วยเหลือผู้วิจัย ขอขอบคุณผู้เชี่ยวชาญที่เข้าร่วมการทดลองนี้ทั้ง 5 ท่านและสุดท้ายผู้วิจัยขอบพระคุณคุณแม่คุณพ่อที่ให้กำลังใจ สนับสนุนมาโดยตลอด

ชานน จรัสสุทธิกุล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# สารบัญ

**หน้า**

.....ค

บทคัดย่อภาษาไทย.....ค

.....ง

บทคัดย่อภาษาอังกฤษ.....ง

กิตติกรรมประกาศ.....จ

สารบัญ.....ฉ

สารบัญตาราง.....ฌ

หน้า.....ฌ

ตารางรูปภาพ.....ถ

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญของปัญหา.....1

1.2 วัตถุประสงค์ของการวิจัย.....7

1.3 ขอบเขตของการวิจัย.....8

1.4 ขั้นตอนและวิธีดำเนินการวิจัย.....8

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....8

1.6 คำจำกัดความที่ใช้ในการวิจัย.....8

บทที่ 2 ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง.....10

2.1 วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์.....10

2.2 วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่นำมาใช้ในการทดลอง.....12

2.3 งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความแม่นยำของค่าความพยามในการพัฒนาซอฟต์แวร์
.....44

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.4 สรุปการทบทวนวรรณกรรม ..... 51
บทที่ 3 ระเบียบวิธีวิจัย ..... 54
3.1 คำถามวิจัย ..... 54
3.2 การตั้งสมมติฐาน ..... 55
3.3 การออกแบบการทดลอง ..... 58
3.4 วิเคราะห์ผลการทดลอง ..... 69
บทที่ 4 ผลการวิเคราะห์ข้อมูล ..... 79
4.1 รายละเอียดโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 80
4.2 รายละเอียดผู้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จำนวน 5 คน ..... 82
4.3 การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธีทั้งสามวิธี คือ ฟังก์ชันพอยต์ (Function Point) โคโคโม่ (COCOMO) ยูสเคสพอยต์ (Use Case Point) ..... 84
4.3.1 ฟังก์ชันพอยต์ ..... 85
4.3.2 โคโคโม่ ..... 104
4.3.3 ยูสเคสพอยต์ ..... 111
4.4 ความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากตัวแบบวิธีทั้งสาม ..... 124
4.5 การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ..... 126
4.6 ความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากวิธีการตัดสินของผู้เชี่ยวชาญ ..... 127
4.7 การทดสอบสมมติฐาน ..... 128
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ ..... 154
5.1 สรุปผลการวิจัย ..... 154
5.2 การนำงานวิจัยไปใช้ ..... 158
5.3 ข้อจำกัดของงานวิจัยและข้อเสนอแนะ ..... 158
บรรณานุกรม ..... 160
ประวัติผู้เขียน ..... 161

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บรรณานุกรม.....162

ภาคผนวก ก.....166

ภาคผนวก ข.....167

ภาคผนวก ค.....168

ภาคผนวก ง.....169

ภาคผนวก จ.....170

ภาคผนวก ซ.....171



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# สารบัญตาราง

**หน้า**

ตารางที่ 1 สรุปผลงานวิจัยที่เกี่ยวข้องกับการทดลองการเปรียบเทียบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี ..... 3

ตารางที่ 2 ค่าตัวแปร c และ k สำหรับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่ ในแต่ละชนิดของระบบ (System Type) (Jong, 1996)..... 13

ตารางที่ 3 ค่าตัวแปร d สำหรับการประเมินค่าความพยายามด้วยวิธีโคโคโม่ ในแต่ละชนิดของโครงการซอฟต์แวร์ (Jong, 1996)..... 13

ตารางที่ 4 ตัวอย่างค่าพิจารณาปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ (Cost Driver) ที่ใช้ในการนำมาใช้ในการประเมินค่าความพยายามด้วยวิธีโคโคโม่ (Jong, 1996)..... 15

ตารางที่ 5 ค่าของตัวแปร a และ b ในการประเมินค่าความพยายามด้วยวิธีโคโคโม่ ในแต่ละชนิดของโครงการซอฟต์แวร์ (Boehm, 1981)..... 16

ตารางที่ 6 ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ของช่วงตัวแบบการออกแบบช่วงต้น และช่วงตัวแบบหลังการมีสถาปัตยกรรม (Westfall, 2009) ..... 21

ตารางที่ 7 คำอธิบายของปัจจัยที่ควรคำนึงเพิ่มเติม (Kishore, 2001)..... 22

ตารางที่ 8 ค่าที่เหมาะสมสำหรับการกำหนดค่าให้กับปัจจัยที่ควรคำนึงเพิ่มเติม (อานุภาพไตรรงค์, 2558)..... 22

ตารางที่ 9 การคิดค่าความซับซ้อนของ EI (Solution., 2004) ..... 27

ตารางที่ 10 การคิดค่าความซับซ้อนของ EO และ EQ (Solution., 2004)..... 27

ตารางที่ 11 การคิดค่าความซับซ้อนของ ILF และ EIF (Solution., 2004)..... 27

ตารางที่ 12 ค่าตัวคูณปัจจัย (Factor Multiplier) ของค่าความซับซ้อนองค์ประกอบทั้ง 5 คือ ILF EIF EI EO และ EQ ในการประเมินค่าความพยายามด้วยวิธีฟังก์ชันพอยต์ Function Point (Menguy, 2001)..... 28

ตารางที่ 13 เกณฑ์การพิจารณาด้านการสื่อสารข้อมูล ..... 29

ตารางที่ 14 เกณฑ์การพิจารณาด้านการประมวลผลข้อมูลแบบกระจาย ..... 30

ตารางที่ 15 เกณฑ์การพิจารณาด้านสมรรถนะ..... 30

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 16 เกณฑ์การพิจารณาด้านความต้องการใช้งานหนักด้านฮาร์ดแวร์ ..... 31
ตารางที่ 17 เกณฑ์การพิจารณาด้านอัตรารายการเปลี่ยนแปลง ..... 31
ตารางที่ 18 เกณฑ์การพิจารณาด้านการบันทึกข้อมูลออนไลน์ ..... 32
ตารางที่ 19 เกณฑ์การพิจารณาด้านประสิทธิภาพเพื่อผู้ใช้ขั้นปลาย ..... 33
ตารางที่ 20 เกณฑ์การพิจารณาด้านการปรับให้เป็นปัจจุบันแบบออนไลน์ ..... 33
ตารางที่ 21 เกณฑ์การพิจารณาด้านการประมวลผลที่ซับซ้อน ..... 34
ตารางที่ 22 เกณฑ์การพิจารณาด้านความสามารถนำกลับมาใช้ใหม่ ..... 35
ตารางที่ 23 เกณฑ์การพิจารณาด้านความสะดวกในการติดตั้ง ..... 35
ตารางที่ 24 เกณฑ์การพิจารณาด้านความสะดวกในการปฏิบัติงาน ..... 36
ตารางที่ 25 เกณฑ์การพิจารณาด้านหลายสถานที่ ..... 36
ตารางที่ 26 เกณฑ์การพิจารณาด้านอำนวยความสะดวกในการเปลี่ยนแปลง ..... 38
ตารางที่ 27 วิธีการคำนวณค่าความพยายามในการพัฒนาซอฟต์แวร์จากฟังก์ชันพอยต์ ..... 38
ตารางที่ 28 ค่าน้ำหนัก (Weighting Factor) ของผู้แอคเตอร์ชนิดต่างๆ ในการประเมินค่าความพยายามด้วยยูสเคสพอยต์ (Use Case Point) (อานุภาพไตรรงค์, 2558) ..... 40
ตารางที่ 29 15 ค่าน้ำหนักปัจจัย (Weighting Factor) ของยูสเคส (Use Case) ชนิดต่างๆ (อานุภาพไตรรงค์, 2558) ..... 41
ตารางที่ 30 ค่าน้ำหนักของปัจจัยทางเทคนิค (Technical Factor) ทั้ง 13 ประการ ในการปรับค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยต์ (อานุภาพไตรรงค์, 2558) ..... 42
ตารางที่ 31 ค่าน้ำหนักของปัจจัยทางด้านทรัพยากร (Environment Factor) ทั้ง 8 ประการ ในการปรับค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยต์ (อานุภาพไตรรงค์, 2558) ..... 43
ตารางที่ 32 ผลสรุปวรรณกรรมเกี่ยวกับการทดลองการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ..... 52
ตารางที่ 33 จำนวนโครงการที่ใช้ในการทดลองและจำนวนผู้เชี่ยวชาญที่เข้าร่วมการทดลองในงานวิจัยทั้ง **5** งานวิจัย ..... 61
ตารางที่ 34 รายละเอียดซอฟต์แวร์ที่นำมาใช้ในการทดลองทั้ง 10 โครงการซอฟต์แวร์ ..... 63

ตารางที่ 35 คำอธิบายโครงการซอฟต์แวร์โดยย่อ ..... 64

ตารางที่ 36 แบบฟอร์มบันทึกค่าประเมิน ..... 68

ตารางที่ 37 สรุปผลค่าประเมินของตัวแบบขั้นตอนวิธีทั้ง 3 วิธี ค่าประเมินของผู้เชี่ยวชาญ และค่าความพยายามในการพัฒนาซอฟต์แวร์จริง ..... 70

ตารางที่ 38 ฟอร์มบันทึกค่าความคลาดเคลื่อนสัมพัทธ ของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบขั้นตอนวิธี ..... 72

ตารางที่ 39 ข้อมูลสมมติของผลการคำนวณค่าความคลาดเคลื่อนสัมพัทธ์และค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของค่าประเมินด้วยวิธีตัวแบบวิธี ..... 74

ตารางที่ 40 ข้อมูลสมมติของผลการคำนวณค่าความคลาดเคลื่อนสัมพัทธ์และค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของค่าประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ..... 75

ตารางที่ 41 ตัวอย่างตารางสรุปผลค่าความคลาดเคลื่อนเฉลี่ย ของแต่ละวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ..... 76

ตารางที่ 42 สรุปการวิเคราะห์ความแปรปรวน ..... 76

ตารางที่ 43 ข้อมูลโครงการซอฟต์แวร์ทั้ง 10 โครงการ ประกอบไปด้วย จำนวนบรรทัด, ประเภทโครงการซอฟต์แวร์ จำนวนความต้องการซอฟต์แวร์ด้านหน้าที่และค่าความพยายามที่ใช้ในการพัฒนาซอฟต์แวร์จริง ..... 81

ตารางที่ 44 การหาค่าองค์ประกอบทั้ง 5 ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 85

ตารางที่ 45 ค่าความซับซ้อนขององค์ประกอบทั้ง 5, ค่า DET, FTR/RET ของ 10 โครงการซอฟต์แวร์ (H คือ High, A คือ Average และ L คือ Low) ..... 85

ตารางที่ 46 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ HHTP ..... 86

ตารางที่ 47 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ PTT ..... 87

ตารางที่ 48 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ KAPP ..... 87

ตารางที่ 49 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ EATD ..... 87

ตารางที่ 50 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ ST ..... 88

ตารางที่ 51 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ CINOUT ..... 88

ตารางที่ 52 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ MINSH ..... 89

ตารางที่ 53 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ LMS ..... 89

ตารางที่ 54 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ LEAP ..... 90

ตารางที่ 55 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ TM ..... 90

ตารางที่ 56 สรุปค่าฟังก์ชันพอยต์ไม่ปรับค่าของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 90

ตารางที่ 57 ค่าความซับซ้อนเชิงเทคนิค การสื่อสารข้อมูล (Data Communication) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 92

ตารางที่ 58 ค่าความซับซ้อนเชิงเทคนิค การประมวลผลข้อมูลแบบกระจาย (Distributed Data Processing) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 92

ตารางที่ 59 ค่าความซับซ้อนเชิงเทคนิค สมรรถนะ (Performance) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 93

ตารางที่ 60 ค่าความซับซ้อนเชิงเทคนิค ความต้องการใช้งานหนักด้านฮาร์ดแวร์ (Havily Used Configuration) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 93

ตารางที่ 61 ค่าความซับซ้อนเชิงเทคนิค อัตรารายการเปลี่ยนแปลง (Transaction Rate) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 94

ตารางที่ 62 ค่าความซับซ้อนเชิงเทคนิค การบันทึกข้อมูลออนไลน์ (On-line Data Entry) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 94

ตารางที่ 63 ค่าความซับซ้อนเชิงเทคนิค ประสิทธิภาพเพื่อผู้ใช้ขั้นปลาย (End-User Efficiency) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 95

ตารางที่ 64 ค่าความซับซ้อนเชิงเทคนิค การปรับให้เป็นปัจจุบันแบบออนไลน์ (On-line Update) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 95

ตารางที่ 65 ค่าความซับซ้อนเชิงเทคนิค การประมวลผลที่ซับซ้อน (Complexity Processing) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 96

ตารางที่ 66 ค่าความซับซ้อนเชิงเทคนิค ความสามารถนำกลับมาใช้ใหม่ (Reusability) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 96

ตารางที่ 67 ค่าความซับซ้อนเชิงเทคนิค ความสะดวกในการติดตั้ง (Installation Ease) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 97

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 68 ค่าความซับซ้อนเชิงเทคนิค ความสะดวกในการปฏิบัติงาน (Operational Ease) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 97

ตารางที่ 69 ค่าความซับซ้อนเชิงเทคนิค หลายสถานที่ (Multiple Sites) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 98

ตารางที่ 70 ค่าความซับซ้อนเชิงเทคนิค คำอธิบายระดับขั้นปัจจัยอำนวยความสะดวกในการเปลี่ยนแปลง (Facilitate Change) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 98

ตารางที่ 71 ค่าฟังชันพอยต์ที่ยังไม่ปรับค่าและฟังก์ชันพอยต์ที่ปรับค่าแล้วของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 102

ตารางที่ 72 วิธีการหาค่าความพยายามในการพัฒนาซอฟต์แวร์จากค่าฟังก์ชันพอยต์ ..... 102

ตารางที่ 73 ค่าความพยายามในการพัฒนาซอฟต์แวร์ที่ได้จากการประเมินด้วยวิธีฟังก์ชันพอยต์ (ผู้วิจัยปรับหน่วยจาก คน-ชั่วโมง เป็น คน-เดือน แล้ว) ..... 103

ตารางที่ 74 ค่าฟังก์ชันพอยต์, ภาษา, ค่า Kilo Source Line of Code ต่อ 1 ฟังก์ชันพอยต์ และขนาดของโครงการซอฟต์แวร์ทั้ง 10 โครงการ (หน่วย Kilo Source Line of Code) ..... 104

ตารางที่ 75 หมวดหมู่และปัจจัยในแต่ละหมวดหมู่ของการประเมินค่าความพยายามด้วยวิธีตัวแบบโคโคโม่ ..... 105

ตารางที่ 76 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ HHTP ..... 105

ตารางที่ 77 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ PTT ..... 106

ตารางที่ 78 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ KAPP ..... 106

ตารางที่ 79 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ EATD ..... 107

ตารางที่ 80 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ ST107

ตารางที่ 81 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ CINOUT ..... 108

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 82 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ MINSH ..... 108
ตารางที่ 83 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ LMS ..... 109
ตารางที่ 84 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ LEAP ..... 109
ตารางที่ 85 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ TM ..... 110
ตารางที่ 86 ค่าประเมินจากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ..... 110
ตารางที่ 87 หนักค่าน้ำแอคเตอร์ของโครงการซอฟต์แวร์ HHTP ..... 112
ตารางที่ 88 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ PTT ..... 112
ตารางที่ 89 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ KAPP ..... 112
ตารางที่ 90 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ EATD ..... 113
ตารางที่ 91 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ ST ..... 113
ตารางที่ 92 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ CINOUT ..... 113
ตารางที่ 93 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ MINSH ..... 114
ตารางที่ 94 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ LMS ..... 114
ตารางที่ 95 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ LEAP ..... 114
ตารางที่ 96 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ TM ..... 115
ตารางที่ 97 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ HHTP ..... 115
ตารางที่ 98 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ PTT ..... 116
ตารางที่ 99 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ KAPP ..... 116

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 100 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ EATD ..... 116

ตารางที่ 101 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ ST ..... 117

ตารางที่ 102 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ CINOUT ..... 117

ตารางที่ 103 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ MINSH ..... 117

ตารางที่ 104 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ LMS ..... 118

ตารางที่ 105 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ LEAP ..... 118

ตารางที่ 106 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ TM ..... 118

ตารางที่ 107 ผลลัพธ์ค่าประเมินของยูสเคสพอยต์ที่ยังไม่ได้ปรับค่า ..... 119

ตารางที่ 108 ผลลัพธ์ของค่าประเมินค่าความซับซ้อนทางเทคนิค ทั้ง 13 ปัจจัย ของโครงการซอฟต์แวร์ในการทดลอง ทั้ง 10 โครงการ ..... 120

ตารางที่ 109 ผลลัพธ์ของค่าประเมินค่าความซับซ้อนทางเทคนิค ภายหลังจากการคำนวณของทั้ง 13 ปัจจัย ของโครงการซอฟต์แวร์ในการทดลอง ทั้ง 10 โครงการ ..... 120

ตารางที่ 110 ผลลัพธ์ของค่าประเมินค่าความซับซ้อนทางสิ่งแวดล้อมทั้ง 8 ปัจจัย ของโครงการซอฟต์แวร์ ทั้ง 10 โครงการ ..... 121

ตารางที่ 111 ผลลัพธ์ของการคำนวณค่าความซับซ้อนทางสิ่งแวดล้อมทั้ง 8 ปัจจัย ของโครงการซอฟต์แวร์ ทั้ง 10 โครงการ ..... 121

ตารางที่ 112 ผลลัพธ์จากการประเมินค่ายูสเคสพอยต์ที่ปรับค่าแล้วของโครงการซอฟต์แวร์ ทั้ง 10 โครงการ ..... 122

ตารางที่ 113 ค่าประเมินจากการประเมินด้วยยูสเคสพอยต์ของโครงการซอฟต์แวร์ในการทดลองทั้ง 10 โครงการ ..... 123

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 114 สรุปค่าความพยายามในการพัฒนาซอฟต์แวร์จากการประเมินด้วยวิธีฟังก์ชันพอยต์ ยูสเคสพอยต์ และ โคโคโม่ (หน่วย คน-เดือน) ..... 124

ตารางที่ 115 ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของตัวแบบขั้นตอนวิธี ฟังก์ชันพอยต์ โคโคโม่ และ ยูสเคสพอยต์ ของโครงการซอฟต์แวร์ที่ใช้ในการทดลองทั้ง 10 โครงการ ..... 125

ตารางที่ 116 ค่าประเมินจากผู้เชี่ยวชาญทั้ง 5 ในการประเมินโครงการซอฟต์แวร์ทั้ง 10 โครงการ 126

ตารางที่ 117 ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของผู้ประเมินทั้ง 5 คน (การประเมินในครั้งที่ 1) ..... 127

ตารางที่ 118 สรุปผลค่าความคลาดเคลื่อนเฉลี่ยของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี ฟังก์ชันพอยต์ โคโคโม่ ยูสเคสพอยต์ และวิธีการตัดสินของผู้เชี่ยวชาญ ..... 128

ตารางที่ 119 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบโคโคโม่ ก่อนการทดสอบสมมติฐาน ..... 129

ตารางที่ 120 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบฟังก์ชันพอยต์ ก่อนการทดสอบสมมติฐาน ..... 129

ตารางที่ 121 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบยูสเคสพอยต์ ก่อนการทดสอบสมมติฐาน ..... 130

ตารางที่ 122 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ก่อนการทดสอบสมมติฐาน ..... 130

ตารางที่ 123 ผลการทดสอบสมมติฐานด้วยวิธี Kruskal-Wallis ..... 131

ตารางที่ 124 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบยูสเคสพอยต์ ..... 133

ตารางที่ 125 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบโคโคโม่และวิธีการตัดสินของผู้เชี่ยวชาญ ..... 134

ตารางที่ 126 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบฟังก์ชันพอยต์และวิธีการตัดสินของผู้เชี่ยวชาญ ..... 136

ตารางที่ 127 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัวแบบโคโคโม่และตัวแบบยูสเคสพอยต์ ..... 137

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 128 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัวแบบฟังก์ชันพอยต์และตัวแบบยูสเคสพอยต์ ..... 138

ตารางที่ 129 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัวแบบโคโคโม่และตัวแบบฟังก์ชันพอยต์ ..... 139

ตารางที่ 130 ผลการทดสอบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์และค่า p-value ของวิธีการประเมิน 6 คู่ ..... 139

ตารางที่ 131 การทดสอบ z-test ของโอกาสเปลี่ยนแปลงค่าประเมินของผู้เชี่ยวชาญทั้ง 5 คน ..... 142

ตารางที่ 132 ค่าความคลาดเคลื่อนเฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินและหลังการเปลี่ยนแปลงค่าประเมินของผู้ประเมินจำนวน 5 คน ..... 143

ตารางที่ 133 การทดสอบความเป็นการกระจายแบบปกติ ..... 144

ตารางที่ 134 การทดสอบ paired t-test ของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินและหลังเปลี่ยนแปลงค่าประเมิน ..... 145

ตารางที่ 135 การทดสอบความเป็นการกระจายแบบปกติ ..... 146

ตารางที่ 136 การทดสอบ paired t-test ของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินและหลังเปลี่ยนแปลงค่าประเมิน ..... 147

ตารางที่ 137 ข้อมูลจำนวนปีประสบการณ์ของผู้ประเมิน ตำแหน่งงานล่าสุด ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยและอันดับความแม่นยำ ..... 148

ตารางที่ 138 ข้อมูลจำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม และค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย ..... 149

ตารางที่ 139 ข้อมูลตำแหน่งงานของผู้ประเมิน ค่าความคาดเคลื่อนสัมพัทธ์เฉลี่ย และอันดับความแม่นยำ ..... 151

ตารางที่ 140 ข้อมูลภาษาการเขียนโปรแกรม ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย และอันดับความแม่นยำของผู้ประเมินทั้ง 5 คน ..... 152

ตารางที่ 141 สรุปผลค่าความคลาดเคลื่อนเฉลี่ยของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี ฟังก์ชันพอยต์ โคโคโม่ ยูสเคสพอยต์ และวิธีการตัดสินของผู้เชี่ยวชาญ ..... 156

ตารางที่ 142 ความแตกต่างของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของวิธีการประเมิน 6 คู่ ..... 156

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 143 ค่า Function Point Language Table..... 166

ตารางที่ 144 การวิเคราะห์ความแปรปรวนของข้อมูลสมมติในสมมติฐานที่ 1 ..... 170

ตารางที่ 145 การทดสอบสมมติฐานของข้อมูลสมมติด้วยการทดสอบแบบจับคู่ ..... **ผิดพลาด! ไม่ได้กำหนดบุ๊กมาร์ก**



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# ตารางรูปภาพ

หน้า

ภาพที่ 1 ความสัมพันธ์ขององค์ประกอบทั้ง 5 ในการประเมินค่าความพยายามด้วยฟังก์ชันพอยต์ (ถาวร อานุภาพไตรรงค์, 2558)..... 26

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# บทที่ 1 บทนำ

## 1.1 ที่มาและความสำคัญของปัญหา

เจ้าของโครงการซอฟต์แวร์ต้องการทราบค่าความพยายามในการพัฒนาซอฟต์แวร์และต้นทุนของซอฟต์แวร์ เช่นเดียวกับ ผู้จัดการโครงการ ที่ต้องการประเมินค่าความพยายามเพื่อวางแผนและควบคุมงานทางด้านการพัฒนาซอฟต์แวร์ แต่อย่างไรก็ตามการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ส่วนใหญ่จะไม่แม่นยำ มีงานวิจัยของ Molokken-Ostvold และ Jorgensen (2003) พบว่าโครงการซอฟต์แวร์ที่มีการพัฒนาขึ้นจริงส่วนใหญ่ใช้ค่าความพยายามในการพัฒนามากกว่าที่ประเมินไว้ 30-40% (M. Jorgensen, 2007)

งานวิจัยของ Standish Group พบว่า 31.1% ของโครงการซอฟต์แวร์ ถูกยกเลิกก่อนจะเสร็จโครงการ และ 52.7% มีต้นทุนในการพัฒนาโครงการสูงกว่าที่ประมาณการไว้ถึง 189% (Group, 2014)

การประเมินที่แม่นยำช่วยให้การพัฒนาซอฟต์แวร์เป็นไปได้ตามที่วางแผนไว้ ลดปัญหาต้นทุนและปัญหาระยะเวลาในการพัฒนาซอฟต์แวร์ที่เกินกำหนด ด้วยเหตุนี้การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จึงมีความสำคัญต่อการพัฒนาโครงการซอฟต์แวร์ (Atkinson, 1994)

ที่ผ่านมาได้มีการนำเสนอวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์หลากหลายวิธี ผู้วิจัยได้ทำการรวบรวมวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ทั้งหมด 9 วิธี โดยมาจากงานวิจัยของ (Shekhar, 2016) และ (Behrens, 1983) ดังนี้

1) วิธีตัวแบบขั้นตอนวิธี (Algorithmic model)
2) วิธีอุปมา (Analogy)
3) วิธีการตัดสินของผู้เชี่ยวชาญ (Expert Judgment)
4) วิธีพาร์กินสัน (Parkinson)
5) วิธีราคาที่ชนะ (Price to win)
6) วิธีจากล่างขึ้นบน (Bottom-up Strategies)
7) วิธีจากบนลงล่าง (Top-down Strategies)
8) วิธีการเรียนรู้ของเครื่อง (Machine Learning)
9) วิธีตรรกศาสตร์คลุมเครือ (Fuzzy Logic)

งานวิจัยของ (Molokken M., 2003) ได้ทำการสำรวจการใช้วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ พบว่า วิธีการตัดสินของผู้เชี่ยวชาญ พบการใช้งานที่แพร่หลายมากที่สุด ถึง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

85% นอกจากนี้ยังพบการใช้งานของวิธีตัวแบบขั้นตอนวิธี ว่าเป็นที่นิยมถึง 13% และเป็นวิธีอื่นๆ อีก 12%

ในปี พ.ศ. 2557 โครงการมาตรฐานราคากลางและเกณฑ์การประเมินราคาซอฟต์แวร์ ของกระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร ได้บรรยายถึงปัญหาที่เกิดขึ้นในธุรกิจการพัฒนาซอฟต์แวร์ในประเทศไทย เช่น ผู้พัฒนาซอฟต์แวร์ไม่สามารถดำเนินการได้ทันระยะเวลาที่ต้องการ โครงการซอฟต์แวร์ค่าใช้จ่ายเกินกว่าที่กำหนดไว้ ผู้ดูแลโครงการไม่ทราบถึงจำนวนผู้พัฒนาซอฟต์แวร์และค่าใช้จ่ายที่เกิดขึ้น (กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร, 2557)

งานวิจัยของ (Arnuphaptrairong, 2018) ได้ทำการสำรวจการใช้วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ในประเทศไทย พบว่า วิธีการตัดสินของผู้เชี่ยวชาญ ถูกนำมาใช้งานอย่างแพร่หลายมากที่สุด

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ และวิธีตัวแบบขั้นตอนวิธีได้มีการถกเถียงกันระหว่าง Magne Jorgensen และ Barry Boehm แต่ก็ยังไม่สามารถสรุปได้ว่าวิธีการประเมินวิธีใดดีกว่ากัน (M. Jorgensen, & Boehm, B W., 2009)

จากการทบทวนวรรณกรรมที่เกี่ยวข้องกับการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ พบว่ามีงานวิจัยที่เกี่ยวข้อง ดังนี้

งานวิจัยของ (M. Jorgensen, 2002) กล่าวถึงการศึกษางานการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญรวมไปถึงการเปรียบเทียบประสิทธิภาพของวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัดสินของผู้เชี่ยวชาญ เทียบกับวิธีตัวแบบขั้นตอนวิธี งานวิจัยนี้ได้ทำการศึกษางานวิจัยอื่นๆ ทั้งหมด 15 งานวิจัย ซึ่งมีทั้งงานวิจัยที่เป็นการทดลอง งานวิจัยที่เป็นการสังเกตการณ์ และงานวิจัยที่เป็นการใช้แบบสอบถาม ผู้วิจัยจะเลือกศึกษาเฉพาะงานวิจัยที่เกี่ยวข้องกับการทดลองเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี เนื่องจากงานวิจัยที่เป็นการทดลองจะสามารถศึกษาแนวทางการทดลองและทราบผลการทดลองว่าวิธีการใดมีความแม่นยำมากว่ากัน งานวิจัยที่เกี่ยวข้องกับการทดลองการเปรียบเทียบความแม่นยำของวิธีทั้งสองมีทั้งหมด 5 งานวิจัย แสดงในตารางที่ 1

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 1 สรุปผลงานวิจัยที่เกี่ยวข้องกับการทดลองการเปรียบเทียบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี

| **งานวิจัย** | **ตัวแบบขั้นตอนวิธี** | **ผลลัพธ์** |
|---|---|---|
| 1.(Kuster, 1990) | บีวายแอล (BYL), เอสติแมกซ์ (Estimacs) | การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี มีความแม่นยำมากกว่า การประมาณด้วยวิธีการตัดสินของผู้เชี่ยวชาญ |
| 2.(Vicinanza, 1991) | โคโคโม่ (COCOMO), ฟังก์ชันพอยต์ (Function Point) | การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| 3.(Mukhopadhyay, 1992) | โคโคโม่ (COCOMO),ฟังก์ชันพอยต์ (Function Point) | การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| 4.(Atkinson, 1994) | ฟังก์ชันพอยต์ (Function Point) | การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| 5.(Pengelly, 1995) | ประสิทธิภาพขนาดบรรทัดในการเขียนโค้ด (ELOC), วิธีตัวแบบโดตี้ (Doty Model), โคโคโม่ (COCOMO), ฟังก์ชันพอยต์ (Function Point), ตัวแบบพัทนัมสลิม (Putnum SLIM Model), การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญโดยใช้วิธีจากล่างขึ้นบน | การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญโดยใช้วิธีจากล่างขึ้นบน มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โดยงานวิจัยทั้ง 5 มีรายละเอียดดังนี้

งานวิจัยของ Kusters et al. (1990) ได้มีการทดลองเปรียบเทียบความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของวิธีการประเมิน 4 วิธี ดังนี้

1) วิธีการตัดสินของผู้เชี่ยวชาญ

2) วิธีตัวแบบบีวายแอล (BYL)

3) วิธีตัวแบบเอสติแมก (Estimacs)

4) วิธีผสมระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี

ในการวิจัยจะให้ข้อมูลโครงการซอฟต์แวร์ชื่อ Bonus System ขององค์กรที่ดำเนินธุรกิจเกี่ยวกับการขาย แก่ผู้ประเมินทั้ง 14 คน มีรายละเอียดที่ประกอบไปด้วยสภาพแวดล้อมขององค์กร รายละเอียดฟังก์ชันการทำงาน และเป้าหมายของโครงการ ข้อมูลเหล่านี้เป็นข้อมูลที่ทางทีมพัฒนาได้รับก่อนพัฒนาโครงการซอฟต์แวร์จริง ภายหลังการประเมิน

งานวิจัยนี้สามารถสรุปการทดลองได้ว่า การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี มีความแม่นยำมากกว่า การประมาณด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

งานวิจัยของ Vicinanza et al. (1991) กล่าวถึงการทดสอบความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ เปรียบเทียบกับวิธีตัวแบบขั้นตอนวิธี 2 แบบ คือ โคโคโม่ และฟังก์ชันพอยต์ กับโครงการซอฟต์แวร์ 10 โครงการ

ผู้วิจัยในการวิจัยนี้ ได้เชิญผู้ประเมินในตำแหน่งผู้จัดการซอฟต์แวร์ (Software Manager) ผู้ที่มีประสบการณ์ 5 คนมาเข้าร่วมการทดสอบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ และบันทึกวิธีการประเมินของผู้ประเมินว่า ผู้ประเมิน ใช้วิธีใดในการประเมิน ซึ่งโครงการซอฟต์แวร์ที่นำมาให้ประเมิน มีขนาดตั้งแต่ 39,000 ถึง 450,000 บรรทัด (lines of code) และมีค่าความพยายามในการพัฒนาซอฟต์แวร์ตั้งแต่ 23 - 1,107 คน-เดือน

งานวิจัยนี้สรุปว่า การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่า โคโคโม่ และ ฟังก์ชันพอยต์ และจากการทดลองพบว่า 4 ใน 5 ของผู้ประเมิน ใช้การตัดสินของผู้เชี่ยวชาญ โดยใช้หลักการของ แผนการขั้นตอนวิธี (Algorithmic Strategies) เป็นแผนการที่ใช้ ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์มาคิด ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ส่วนผู้ประเมิน อีกท่านใช้หลักการของแผนการอุปมา (Analogy Strategies) เป็นแผนการที่ไม่ได้ใช้ ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ แต่ใช้การเปรียบเทียบค่าฟังก์ชันพอยต์ ที่คล้ายคลึงกันในโครงการซอฟต์แวร์ที่เคยทำมา ซึ่ง ผู้ประเมินที่ใช้แผนการอุปมา นั้นสามารถประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ได้แม่นยำมากที่สุด (แต่ไม่สามารถสรุปได้ว่าการใช้วิธีการตัดสินของผู้เชี่ยวชาญ ด้วยแผนการอุปมา นั้นมี

ความแม่นยำมากกว่าแผนการขั้นตอนวิธี เนื่องจากผู้ประเมินที่ใช้วิธีแผนการขั้นตอนวิธี มีเพียงคนเดียว)

งานวิจัยของ Mukhopadhyay et al. (1992) พบว่าการสร้างแบบจำลองของการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ ชื่อว่า เอสเตอร์ (Estor) เป็นตัวแบบการให้เหตุผลตามกรณี (Case-Based Reasoning Model หรือ CBR) มาจากการระดมความคิดของผู้เชี่ยวชาญ

เพื่อทดสอบความแม่นยำและความแน่นอนของผลลัพธ์ที่ได้จาก เอสเตอร์ นักวิจัยโครงงานชิ้นนี้จึงนำเอสเตอร์มาทำการทดลองเปรียบเทียบกับวิธีการตัดสินของผู้เชี่ยวชาญ และ วิธีตัวแบบขั้นตอนวิธี อีก 2 ชนิดคือ โคโคโม่ และ ฟังก์ชันพอยต์ โดยใช้โครงการซอฟต์แวร์ที่สำเร็จแล้ว 15 โครงการ

ผลการทดลองสามารถเรียงลำดับความแม่นยำได้ดังนี้วิธีการตัดสินของผู้เชี่ยวชาญ เอสเตอร์ โคโคโม่ และ ฟังก์ชันพอยต์ ตามลำดับ

งานวิจัยของ Pengelly (1995) กล่าวถึงการทดลองการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของโครงการคุณภาพของระบบสารสนเทศการบริหารของการบริการ (the Quality of Service management Information system) ซึ่งเป็นการทดลองเปรียบเทียบการประเมินระหว่างวิธี

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1) ประสิทธิภาพจำนวนบรรทัดในการเขียนโค้ด (Effective lines of code หรือ ELOC)
2) วิธีการตัดสินของผู้เชี่ยวชาญ (Expert Judgment) โดยใช้ วิธีจากล่างขึ้นบน (Bottom-up Estimation)
3) ตัวแบบโดตี้ (Doty Model)
4) โคโคโม่ (COCOMO)
5) ฟังก์ชันพอยต์ (Function Point Analysis)
6) ตัวแบบพัทนัมสลิม (Putnum SLIM Model)

การทดลองจะดำเนินการตามวิธีการประเมินของแต่ละวิธีเพื่อให้ได้ค่าประมาณจากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ หลังจากนั้นจะวิเคราะห์ผลลัพธ์ที่ได้จากการประเมินของแต่ละวิธี

งานวิจัยนี้สรุปผลการทดลองได้ว่า การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าการประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี ตัวแบบอื่นๆ

นอกจากนี้ ผู้วิจัยได้ทำการศึกษางานวิจัยของ Jorgensen เพิ่มเติมพบว่า

งานวิจัยของ Jorgensen (2007) กล่าวถึงการทบทวนวรรณกรรมของงานที่เกี่ยวข้องเกี่ยวกับการใช้ วิธีการตัดสินของผู้เชี่ยวชาญ และ ตัวแบบขั้นตอนวิธี และ การรวมวิธีทั้งสอง ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ จากการศึกษาวรรณกรรมที่เกี่ยวข้องทั้ง 16 วรรณกรรม พบวรรณกรรมที่อ้างถึง วิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่า ตัวแบบขั้นตอนวิธี จำนวน 10 วรรณกรรม (คิดเป็น 62.5%) ส่วนอีก 6 วรรณกรรมเป็นวรรณกรรมที่ วิธีตัวแบบขั้นตอนวิธีมีความแม่นยำมากกว่าวิธีการตัดสินของผู้เชี่ยวชาญ แต่ยังไม่สามารถสรุปผลได้อย่างแน่ชัดเนื่องจากพบว่า 12 จาก 16 งานวิจัยที่เกี่ยวข้อง มีการให้ปริมาณข้อมูลที่ใช้ประกอบการประเมินของวิธีทั้งสองไม่เท่ากัน (รายละเอียดวรรณกรรมทั้ง 16 วรรณกรรม ปรากฏอยู่ใน https://ijf.forecasters.org/data.htm แต่ปัจจุบันไม่พบหลักฐานข้อมูลในเว็บไซต์ดังกล่าว) (M. Jorgensen, 2007)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

งานวิจัยของ ขวัญชนก ธรณธรรมกุล และปิติพัฒน์ เจิมบุญธนะพงศ์ (2558) กล่าวถึงการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่าง โคโคโม่ ร่วมกับ Web Model และ วิธีการตัดสินของผู้เชี่ยวชาญ งานวิจัยนี้สรุปผลการทดลองว่า การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี โคโคโม่ ร่วมกับ Web Model มีความแม่นยำกว่าวิธีการประเมินค่าความพยายามด้วยวิธีการตัดสินของผู้เชี่ยวชาญ (เจิมบุญธนะพงศ์, 2558)

จากการศึกษางานวิจัยที่กล่าวมาข้างต้น พบว่า

1. งานวิจัยที่ผู้วิจัยศึกษามา ขาดความทันสมัยของข้อมูล

   ยังไม่พบงานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบ

ขั้นตอนวิธี ที่เป็นงานวิจัยในยุคปัจจุบัน อีกทั้งงานวิจัยที่กล่าวมาข้างต้นภาษาการเขียนโปรแกรมที่ใช้ในการทดลองยังเป็นภาษาที่เก่าอยู่ อาทิเช่น ภาษา COBOL เป็นต้น

2. มีงานวิจัยจำนวนหนึ่งขาดความน่าเชื่อถือของข้อมูล

งานวิจัยของ Kusters (1990) และ Pengelly (1995) ใช้โครงการซอฟต์แวร์ในการทดลอง เพียง 1 โครงการ ซึ่งทำให้ผลลัพธ์จากการทดลองขาดความน่าเชื่อถือ ส่วนงานวิจัยของ ขวัญชนก ธรณธรรมกุล และปิติพัฒน์ เจิมบุญธนะพงศ์ (2558) ได้ทำการทดลองโดยใช้โครงการซอฟต์แวร์ เนื้อหาบทเรียนออนไลน์ เพียงโครงการเดียว

3. การทดลองของงานวิจัย ไม่ได้ทดลองในสภาวะแวดล้อมในประเทศไทย

มีงานวิจัยของ (Adrian, 2013) กล่าวถึงการวิเคราะห์ความสามารถในการพัฒนาซอฟต์แวร์ในประเทศต่างๆ พบว่า สภาพแวดล้อมและวัฒนธรรมองค์กรในการพัฒนาซอฟต์แวร์ แตกต่างกันในแต่ละประเทศ ทำให้ความสามารถในการพัฒนาซอฟต์แวร์ของแต่ละประเทศไม่เท่ากัน งานวิจัยที่ผู้วิจัยได้ทบทวนวรรณกรรมส่วนใหญ่ไม่ได้ทำการทดลองในสภาวะแวดล้อมในประเทศไทย จึงอาจทำให้ผลลัพธ์จากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์อันเนื่องมาจากสภาวะแวดล้อมคนละประเทศกันมีความแตกต่างกัน การทดลองในสภาวะแวดล้อมในประเทศไทย หมายถึง การดำเนินการพัฒนาซอฟต์แวร์ภายในประเทศไทย ทีมผู้พัฒนาเป็นคนไทย ภายใต้บริษัทซอฟต์แวร์ที่จดทะเบียนในประเทศไทย

ผู้วิจัยจึงไม่สามารถสรุปได้ว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีตัวแบบขั้นตอนวิธี และวิธีการตัดสินของผู้เชี่ยวชาญ วิธีใดแม่นยำกว่ากัน ภายใต้สภาวะแวดล้อมการพัฒนาซอฟต์แวร์ในประเทศไทย รวมไปถึงปัจจัยของผู้ประเมิน ว่าผู้ที่ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ควรมีลักษณะเช่นใด การประเมินถึงจะมีความแม่นยำ และค่าความแม่นยำของผู้ประเมินเมื่อทราบถึงค่าประเมินของการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**1.2 วัตถุประสงค์ของการวิจัย**

1) เพื่อเปรียบเทียบความแม่นยำของการประเมินความพยายามในการพัฒนาซอฟต์แวร์ภายในสภาวะแวดล้อมที่ผู้เชี่ยวชาญเป็นคนไทย ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี
2) เพื่อศึกษาความแม่นยำของการประเมิน เมื่อนำผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีมาประกอบการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

**1.3 ขอบเขตของการวิจัย**

1) งานวิจัยนี้จะทำการทดลองโดยใช้โครงการซอฟต์แวร์ที่ได้มีการดำเนินโครงการเสร็จแล้ว
2) การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีที่เกิดขึ้นในงานวิจัยนี้มี 3 ตัวแบบขั้นตอนวิธีด้วยกันคือ โคโคโม่ ฟังก์ชันพอยต์ และ ยูสเคสพอยต์
3) งานวิจัยนี้จะวิจัยเฉพาะความแม่นยำในช่วงเริ่มต้นของโครงการซอฟต์แวร์หรือการประมูลโครงการซอฟต์แวร์

**1.4 ขั้นตอนและวิธีดำเนินการวิจัย**

1) ผู้วิจัยหาโครงการซอฟต์แวร์จำนวน 10 โครงการและติดต่อผู้เชี่ยวชาญ 5 คน
2) ผู้วิจัยทำการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีจำนวน 10 โครงการและบันทึกผล
3) ผู้เชี่ยวชาญกรอกแบบสอบถามที่ 1 ก่อนเริ่มการทดลอง
4) ผู้เชี่ยวชาญประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญทีละโครงการ บันทึกผล และกรอกแบบสอบถามที่ 2 หลังการประเมินทีละโครงการ
5) ผู้วิจัยแสดงข้อมูลประเมินที่ได้จากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีและให้ผู้เชี่ยวชาญประเมินใหม่อีกครั้งหนึ่ง พร้อมทั้งบันทึกผลการประเมินทีละโครงการ

ผู้วิจัยและผู้เชี่ยวชาญปฏิบัติตามขั้นตอนที่ 4 และ 5 จนกว่าจะครบทั้ง 10 โครงการ

**1.5 ประโยชน์ที่คาดว่าจะได้รับ**

1) ช่วยให้ทราบว่าวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีตัวแบบขั้นตอนวิธีหรือการตัดสินของผู้เชี่ยวชาญวิธีใดแม่นยำมากกว่ากันในสภาวะแวดล้อมของคนไทย
2) ช่วยให้องค์กรหรือหน่วยงานสามารถเลือกวิธีการประเมินและ/หรือเลือกผู้เชี่ยวชาญในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ได้อย่างเหมาะสม

**1.6 คำจำกัดความที่ใช้ในการวิจัย**

**ผู้ประเมินหรือผู้เชี่ยวชาญ (Expert)** คือ บุคคลที่มีความชำนาญ หรือ มีเกณฑ์ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ โดยเกณฑ์การประเมินได้มาจากความรู้ ความเข้าใจในเรื่องเหล่านั้น (Institutes., 2004)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

**หน่วยค่าความพยายามในการพัฒนาซอฟต์แวร์** คือ จำนวนคน-เดือน (Man-month) ที่ใช้ในการพัฒนาซอฟต์แวร์ โดย 1 คน-เดือนมีค่าเท่ากับ 152 คนชั่วโมง (Man-Hour) (Ourada, 1991)

**ค่าความคลาดเคลื่อนสัมพัทธ์ (Magnitude Relative Error หรือ MRE)** คือ ค่าสัมบูรณ์ของความแตกต่างระหว่างค่าประมาณและค่าที่เกิดขึ้นจริง หารด้วยค่าที่เกิดขึ้นจริง (Fenton, 1997) ดังสมการ

$$\text{MRE} = \left| \frac{MMest - MMact}{MMact} \right|$$

โดย

MMact คือ ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง

MMest คือ ค่าประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จากผู้เชี่ยวชาญ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 2 ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

ในบทนี้จะทบทวนวรรณกรรมที่เกี่ยวข้อง ซึ่งประกอบไปด้วย

1. วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์
2. วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่นำมาใช้ในการทดลอง
3. งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความแม่นยำของค่าความพยายามในการพัฒนาซอฟต์แวร์
4. สรุปการทบทวนวรรณกรรม

## 2.1 วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์

หัวข้อนี้จะกล่าวถึงการหาค่าความพยายามในการพัฒนาซอฟต์แวร์ ด้วยวิธีต่างๆ (Shekhar, 2016) (Behrens, 1983) ดังนี้

1) วิธีตัวแบบขั้นตอนวิธี (Algorithmic model)
2) วิธีอุปมา (Analogy)
3) วิธีการตัดสินของผู้เชี่ยวชาญ (Expert Judgment)
4) วิธีพาร์กินสัน (Parkinson)
5) วิธีราคาที่ชนะ (Price to win)
6) วิธีจากล่างขึ้นบน (Bottom-up Strategies)
7) วิธีจากบนลงล่าง (Top-down Strategies)
8) วิธีการเรียนรู้เรื่องเครื่อง (Machine Learning)
9) วิธีตรรกศาสตร์คลุมเครือ (Fuzzy Logic)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1) การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี (Algorithmic Model)

การนำเอาคุณลักษณะ (Characteristics) บางประการของระบบหรือสิ่งแวดล้อมระบบที่จะพัฒนา มาเป็นตัวแปร ในการประมาณหรือพยากรณ์ค่าความพยายามและต้นทุนในการพัฒนาซอฟต์แวร์ (อานุภาพไตรรงค์, 2558)

ตัวอย่างของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีนี้ที่ได้รับความนิยม เช่น โคโคโม่ (COCOMO) คิดค้นโดย Boehm ฟังก์ชันพอยต์ (Function Point) คิดค้นโดย Albrecht และ ยูสเคสพอยต์ (Use Case Point) คิดค้นโดย Karner (Shekhar, 2016) (Kemerer, 1987)

2) การประเมินค่าความพยายามด้วยวิธีอุปมา (Analogy)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีอุปมา (Analogy) เป็นการใช้ข้อมูลของโครงการซอฟต์แวร์ในอดีตที่มีค่าของพารามิเตอร์ใกล้เคียงกับซอฟต์แวร์ที่จะใช้ในการประเมินตัวอย่างตัวแปร เช่น ขนาดซอฟต์แวร์ เวลา เป็นต้น (Boehm, 1981)

3) การประเมินค่าความพยายามด้วยวิธีการตัดสินของผู้เชี่ยวชาญ (Expert Judgment)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ (Expert Judgment) การประเมินด้วยวิธีนี้เป็นการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยใช้การตัดสินใจ/ความคิดเห็น ของผู้เชี่ยวชาญ (Expert) ซึ่งการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี Expert Judgment เป็นวิธีที่ได้รับความนิยมและนำไปประยุกต์ใช้จริงในปัจจุบัน (M. Jorgensen, 2005)

4) การประเมินค่าความพยายามด้วยวิธีพาร์กินสัน (Parkinson)

เป็นการประมาณค่าความพยายามโดยการพิจารณาจากแรงงานของพนักงานที่มีอยู่ (Staff Effort Available) มาเป็นฐานในการประมาณ (อานุภาพไตรรงค์, 2558)

5) การประเมินค่าความพยายามด้วยวิธีราคาที่ชนะ (Price to win)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีราคาที่ชนะ (Price to win) เป็นการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ โดยพิจารณาจากงบประมาณของลูกค้าหรือราคาของซอฟต์แวร์ที่เกิดจากการประมูลราคาซอฟต์แวร์ (Boehm, 1981)

6) การประเมินค่าความพยายามด้วยวิธีจากล่างขึ้นบน (Bottom-up Strategies)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีจากล่างขึ้นบน (Bottom-up-Strategies) เป็นการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยเริ่มประเมินจากองค์ประกอบย่อย หรือระบบย่อย ของซอฟต์แวร์ แล้วจึงนำผลลัพธ์จากการประเมินมารวมกันเป็นระบบใหญ่ เพื่อให้ได้ค่าความพยายามในการพัฒนาซอฟต์แวร์ทั้งโครงการ การประเมินด้วยวิธีนี้จะทำให้ผู้ประเมินทำการประเมินจากความเข้าใจในการทำงานของซอฟต์แวร์ตั้งแต่องค์ประกอบเล็กๆ และเมื่อเกิดความคลาดเคลื่อนในการประเมินค่าความพยายาม ผู้ประเมินสามารถตรวจสอบได้ว่าเกิดความคลาดเคลื่อนที่ระบบย่อยใด (Boehm, 1981)

7) การประเมินค่าความพยายามด้วยวิธีจากบนลงล่าง (Top-down Strategies)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีจากบนลงล่าง Top-down Strategies เป็นการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยพิจารณาจากภาพรวมของซอฟต์แวร์ก่อน เมื่อได้ค่าความพยายามในการพัฒนาซอฟต์แวร์แบบภาพรวมแล้ว จึงทำการแบ่งซอฟต์แวร์ออกเป็นระบบที่เล็กลง ซึ่งข้อดีของการประเมินค่าความพยายามด้วยวิธีนี้ คือจะทำให้ผู้ประเมินไม่ลืมที่จะคำนึงถึงการประกอบรวมส่วนต่างๆของระบบ (Integration) การกำหนดค่าในการ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การจัดการ (Configuration Management) และ อื่นๆ ที่เกี่ยวข้องและส่งผลต่อความแม่นยำในการประเมิน (Boehm, 1981)

8) การประเมินค่าความพยายามด้วยวิธีการเรียนรู้เรื่องเครื่อง (Machine Learning)

การประเมินค่าความพยายามด้วยวิธีการเรียนรู้เรื่องเครื่อง (Machine Learning) เป็นการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยการนำข้อมูลลักษณะของโครงการซอฟต์แวร์ เช่น ขนาดของซอฟต์แวร์ ภาษาที่ใช้ในการพัฒนา ชนิดของโครงการซอฟต์แวร์ ขนาดของฐานข้อมูล ค่าความซับซ้อนของโครงการซอฟต์แวร์ มาสร้างแบบจำลองของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ด้วยอัลกอริทึมทาง Machine Learning เช่น Neural Network, Support Vector Machine เป็นต้น (Tayyab, 2017)

9) การประเมินค่าความพยายามด้วยวิธีตรรกศาสตร์คลุมเครือ (Fuzzy Logic)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตรรกศาสตร์คลุมเครือ (Fuzzy Logic) เป็นการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยใช้ การกำหนดเงื่อนไข (IF-ELSE) มาแปลงเป็นสมการทางคณิตศาสตร์ ซึ่งสามารถประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ได้แม้ข้อมูลที่ได้รับจะไม่สมบูรณ์ ตรรกศาสตร์คลุมเครือจะสามารถสร้างผลลัพธ์ที่มีค่าอยู่ระหว่างค่า 0 และ 1 ได้ ซึ่งจะไม่เหมือนกับการกำหนดเงื่อนไขตรงๆที่มีเพียงแค่ค่า 0 หรือ 1 (Shekhar, 2016)

**2.2 วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่นำมาใช้ในการทดลอง**

นักวิจัยได้เลือกวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ด้วยวิธีตัวแบบขั้นตอนวิธีและ วิธีการตัดสินของผู้เชี่ยวชาญ เนื่องจากได้รับความนิยมนำมาใช้ในการประเมินมากที่สุด (M. Jorgensen, & Boehm, B W., 2009) ในส่วนของวิธีตัวแบบขั้นตอนวิธี นักวิจัยเลือกใช้ตัวแบบในการพัฒนาซอฟต์แวร์ที่ใช้ในการทดลอง 3 วิธี ดังนี้

1) โคโคโม่ (COCOMO) เป็นตัวแบบขั้นตอนวิธีที่นิยมใช้กันอย่างแพร่หลาย (Shekhar, 2016)
2) ฟังก์ชันพอยต์ (Function Point) นิยมใช้ในองค์กรส่วนใหญ่ (Arnuphaptrairong, 2018)
3) ยูสเคสพอยต์ (Use Case Point) เป็นตัวแบบขั้นตอนวิธีที่นิยมใช้ในการประเมินซอฟต์แวร์เชิงวัตถุ (Fan, 2009)

1) โคโคโม่ (Constructive Cost Model หรือ COCOMO)

โคโคโม่ เป็นแบบจำลองที่ใช้อย่างแพร่หลาย คิดค้นโดย Barry Boehm แบ่งออกเป็น 3 เวอร์ชัน คือ

ก) โคโคโม่ 81 (COCOMO Basic Model หรือ COCOMO 81)

โคโคโม่ 81 เป็นวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เวอร์ชันแรกของโคโคโม่ เพื่อใช้ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ มีวิธีการคำนวณดังนี้

$$E = c \times size^{k}$$

E คือค่าความพยายามในการพัฒนาซอฟต์แวร์ มีหน่วยเป็นคน-เดือน (Man-Month)

Size คือขนาดของซอฟต์แวร์มีหน่วยเป็น KSLOC (Kilo Source Line of Code) c และ k เป็นค่าคงที่ โดยพิจารณาจากชนิดของระบบ ดังตารางที่ 2

ตารางที่ 2 ค่าตัวแปร c และ k สำหรับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่ ในแต่ละชนิดของระบบ (System Type) (Jong, 1996)

| ชนิดของระบบ (System Type) | c | k |
|---|---|---|
| ออแกนิค (Organic) | 2.4 | 1.05 |
| เซมิ-ดีแทช (Semi-detached) | 3.0 | 1.12 |
| ฝังตัว (Embeded) | 3.6 | 1.20 |

ออแกนิค คือ โครงการซอฟต์แวร์ที่สร้างด้วยทีมขนาดเล็ก ซอฟต์แวร์ขนาดเล็ก สมาชิกในทีมมีประสบการณ์ในการพัฒนาซอฟต์แวร์ลักษณะนี้มาก่อนแล้ว

เซมิ-ดีแทช คือ โครงการซอฟต์แวร์ที่ต้องใช้สภาพแวดล้อมในการพัฒนามีลักษณะพิเศษ เช่น ต้องใช้จำนวนคนจำนวนหนึ่ง ต้องใช้ผู้พัฒนาที่มีประสบการณ์จำนวนหนึ่ง ต้องใช้ผู้พัฒนาที่มีความรู้ด้านการพัฒนาจำนวนหนึ่ง สมาชิกในทีมพัฒนาไม่เคยมีประสบการณ์ในการพัฒนาซอฟต์แวร์ประเภทนี้มาก่อน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ฝังตัว คือ โครงการซอฟต์แวร์ที่มีความซับซ้อนสูงมาก ต้องการทีมพัฒนาจำนวนมาก และผู้พัฒนาต้องมีประสบการณ์มากพอที่จะสามารถพัฒนาซอฟต์แวร์ที่ซับซ้อนนี้ได้

เวลาที่ใช้ในการพัฒนาระบบ (Development Time) สามารถคำนวณได้จาก

$$DT = 2.5(E)^{d}$$

โดยที่ DT มีหน่วยเป็นเดือน และ d มีค่าดังตารางที่ 3

ตารางที่ 3 ค่าตัวแปร d สำหรับการประเมินค่าความพยายามด้วยวิธีโคโคโม่ ในแต่ละชนิดของโครงการซอฟต์แวร์ (Jong, 1996)

| โครงการซอฟต์แวร์ (Software Project) | d |
|---|---|
| ระบบออแกนิค (Organic) | 0.38 |
| ระบบเซมิ-ดีแทช (Semi-detached) | 0.35 |
| ระบบฝังตัว (Embedded) | 0.32 |

ข) เวอร์ชั่นกลางของโคโคโม่ 81 (Intermediate Version of COCOMO 81)

เวอร์ชั่นกลางของโคโคโม่ 81 (Intermediate Version of COCOMO 81) จะแตกต่างจากโคโคโม่ 81 (COCOMO 81) ตรงที่ เวอร์ชันกลางของโคโคโม่ 81 นั้นจะพิจารณาปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ (Cost Driver) มีทั้งหมด 15 ปัจจัยด้วยกัน ดังนี้

**หมวดหมู่ ปัจจัยทางด้านผลิตภัณฑ์ (Product Attributes)**

ก) ความต้องการซอฟต์แวร์ให้มีความน่าเชื่อถือ (Required Software Reliability หรือ RELY)

ข) ขนาดของฐานข้อมูลแอปพลิเคชัน (Size of Application Database หรือ DATA)

ค) ความซับซ้อนของโครงการซอฟต์แวร์ (Complexity of the Product หรือ CPLX)

**หมวดหมู่ ปัจจัยทางด้านฮาร์ดแวร์ (Hardware attributes)**

ก) ข้อจำกัดด้านประสิทธิภาพของระยะเวลาประมวลผล (Run-time Performance Constraints หรือ TIME)

ข) ข้อจำกัดด้านหน่วยความจำ (Memory Constraints หรือ STOR)

ค) ความผันผวนของทรัพยากรที่เครื่องคอมพิวเตอร์เสมือนใช้งาน (Volatility of the Virtual Machine Environment หรือ VIRT)

ง) ความต้องการด้านเวลาในการประมวลผลความต้องการของผู้ใช้งาน (Required turnabout time หรือ TURN)

**หมวดหมู่ ปัจจัยด้านบุคคล (Personal attributes)**

จ) ความสามารถในการคิดวิเคราะห์ (Analyst Capability หรือ ACAP)

ฉ) ความสามารถด้านวิศวกรรมซอฟต์แวร์ (Software Engineering Capability หรือ PCAP)

ช) ประสบการณ์ทางด้านแอปพลิเคชัน (Application Experience หรือ AEXP)

ซ) ประสบการณ์ทางด้านเครื่องคอมพิวเตอร์ (Virtual Machine Experience หรือ VEXP)

ฌ) ประสบการณ์ทางด้านภาษาโปรแกรมมิ่ง (Programming Language Experience หรือ LEXP)

**หมวดหมู่ ปัจจัยด้านโครงการซอฟต์แวร์ (Project Attributes)**

ญ) การใช้งานเครื่องมือซอฟต์แวร์ (Use of Software Tools หรือ TOOL)

ฎ) วิธีการทางด้านวิศวกรรมซอฟต์แวร์ (Application of Software Engineering Methods หรือ MODP)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ฏ) ความต้องการทางด้านระยะเวลาการพัฒนา (Required Development Schedule หรือ SCED)

การกำหนดตัวเลขของปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ แสดงในตารางที่ 4

ตารางที่ 4 ตัวอย่างค่าพิจารณาปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ (Cost Driver) ที่ใช้ในการนำมาใช้ในการประเมินค่าความพยายามด้วยวิธีโคโคโม่ (Jong, 1996)

| ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ (Cost Driver) | ต่ำมาก (Very Low) | ต่ำ (Low) | ปกติ (Nominal) | สูง (High) | สูงมาก (Very High) | สูงพิเศษ (Extra High) |
|---|---|---|---|---|---|---|
| RELY | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 | |
| DATA | | 0.94 | 1.00 | 1.08 | 1.16 | |
| CPLX | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| TIME | | | 1.00 | 1.11 | 1.30 | 1.66 |
| STOR | | | 1.00 | 1.06 | 1.21 | 1.56 |
| VIRT | | 0.87 | 1.00 | 1.15 | 1.30 | |
| TURN | | 0.87 | 1.00 | 1.07 | 1.15 | |
| ACAP | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 | |
| PCAP | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 | |
| AEXP | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 | |
| VEXP | 1.21 | 1.10 | 1.00 | 0.90 | | |
| LEXP | 1.14 | 1.07 | 1.00 | 0.95 | | |
| TOOL | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 | |
| MODP | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 | |
| SCED | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 | |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ผู้วิจัยได้นำข้อมูลในตารางมาจาก (Jong, 1996) ซึ่งตามแหล่งอ้างอิงไม่ได้ระบุเหตุผลที่มีช่องว่างในตาราง ผู้ประเมินจะให้คะแนนปัจจัยแต่ละปัจจัยเป็นระดับตั้งแต่ ต่ำมาก (Very Low) ถึงสูงพิเศษ (Extra High) ผลคูณของปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์จะเรียกว่า

ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ที่ปรับค่าแล้ว (Effort Adjustment Factor หรือ EAF) ส่วนใหญ่จะมีค่าอยู่ระหว่าง 0.9-1.4

วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จะแตกต่างจากโคโคโม่ 81 (COCOMO 81) โดยคำนวณปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ เพิ่มเติมดังนี้

$$E = a(Size)^b \times EAF$$

E คือ ค่าความพยายามในการพัฒนาซอฟต์แวร์มีหน่วยเป็นคน-เดือน (Man-month)
Size คือ ขนาดของซอฟต์แวร์มีหน่วยเป็นจำนวนบรรทัดของโค้ด (LOC หรือ Line of Code)

EAF คือ ผลลัพธ์จากการคูณของปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ทั้งหมด

ค่าของ a และ b มีค่าดังตารางที่ 5

ตารางที่ 5 ค่าของตัวแปร a และ b ในการประเมินค่าความพยายามด้วยวิธีโคโคโม่ ในแต่ละชนิดของโครงการซอฟต์แวร์ (Boehm, 1981)

| โครงการซอฟต์แวร์ | a | b |
|---|---|---|
| ระบบอินทรีย์ (Organic) | 3.2 | 1.05 |
| ระบบเซมิ-ดีแทช (Semi-detached) | 3.0 | 1.12 |
| ระบบฝังตัว (Embedded) | 2.8 | 1.20 |

ค) ) COCOMO II (2000)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

COCOMO II นั้นจะแบ่งออกเป็นสามขั้นตอนในการพัฒนาระบบ ตามช่วงเวลาต่างๆของการพัฒนาซอฟต์แวร์

- ตัวแบบการประกอบร่างโปรแกรมประยุกต์ (Application Composition Model) – สำหรับช่วงแรกของการพัฒนา เช่น การทำต้นแบบผลิตภัณฑ์ซอฟต์แวร์ (Prototype)
- ตัวแบบการออกแบบช่วงต้น (Early Design Model) – ใช้สำหรับประเมินช่วงถัดไปของการพัฒนาซอฟต์แวร์ เป็นช่วงสำหรับออกแบบโครงสร้างซอฟต์แวร์ หรือหาแนวทางในการออกแบบสถาปัตยกรรมซอฟต์แวร์
- ตัวแบบหลังการมีสถาปัตยกรรม (Post-Architecture Model) – เมื่อโครงการซอฟต์แวร์พร้อมสำหรับการพัฒนาซอฟต์แวร์ มีสถาปัตยกรรมซอฟต์แวร์ที่ชัดเจน ซึ่งการใช้แบบจำลองของตัวแบบหลังการมีสถาปัตยกรรม (Post-Architecture) นี้จะมี ปัจจัยที่ส่งผลต่อค่าความ

พยายามในการพัฒนาซอฟต์แวร์ (Cost Driver) ค่อนข้างเยอะทำให้ผู้ประเมินสามารถประเมินได้แม่นยำมากขึ้น

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของ COCOMO II คำนวณได้จาก

$$PM = (2.94\ (Size)^{E}\ ) \times EM$$

เมื่อ PM คือค่าความพยายามในการพัฒนาซอฟต์แวร์หน่วยเป็น Man-month

Size คือ ขนาดของซอฟต์แวร์มีหน่วยเป็น KSLOC

E คือ ค่าปัจจัยขยาย (Scale Factor) อธิบายในภายหลัง

EM คือ ผลคูณของปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort Multiplier)

ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ ที่ใช้ในการคำนวณหาค่า EM ของ COCOMO II มีทั้งหมด 22 ปัจจัย แบ่งตามหมวดหมู่ดังนี้ (Westfall, 2009)

**หมวดหมู่ ปัจจัยทางด้านผลิตภัณฑ์ (Product Attributes)**

ก) ความซับซ้อนและความน่าเชื่อถือของซอฟต์แวร์ (Product reliability and complexity หรือ RCPX) เป็นการพิจารณาความสามารถของซอฟต์แวร์ว่าสามารถทำงานได้ต่อเนื่องและถูกต้องในช่วงเวลาที่กำหนด โดยหากความซับซ้อนของผลิตภัณฑ์อยู่ธรรมดามาก ธรรมดา ซับซ้อนบ้าง ปานกลาง ซับซ้อน ซับซ้อนมาก ซับซ้อนที่สุด จะมีค่า RCPX อยู่ในระดับ ต่ำที่สุด ต่ำมาก ต่ำ ปานกลาง สูง สูงมากและสูงที่สุด ตามลำดับ

ข) ความต้องการความน่าเชื่อถือของซอฟต์แวร์ (Required Software Reliability หรือ RELY) เป็นการพิจารณาระดับผลกระทบของซอฟต์แวร์เมื่อล้มเหลว ถ้าหากผลกระทบจากการที่ซอฟต์แวร์ล้มเหลวไม่สามารถทำงานได้ มีเพียงเล็กน้อย จะอยู่ในระดับ ต่ำมาก แต่ในทางตรงกันข้าม ถ้าผลกระทบจากการที่ซอฟต์แวร์ล้มเหลวไม่สามารถทำงานได้แล้วอาจก่อให้เกิดอันตรายถึงชีวิต RELY จะอยู่ในระดับสูงมาก

ค) ขนาดของฐานข้อมูลแอปพลิเคชัน (Size of Application Database หรือ DATA) เป็นการพิจารณาระดับผลกระทบของขนาดข้อมูลทดสอบที่จะมีต่อการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ หากข้อมูลทดสอบหารด้วยจำนวนบรรทัดของโปรแกรมน้อยกว่า 10 จะมีผลกระทบในระดับต่ำ หากอัตราส่วนดังกล่าวมากกว่า 10 แต่ไม่ถึง 100 จะมีผลกระทบในระดับปานกลาง หากอัตราส่วนดังกล่าวมีค่ามากกว่า 100 แต่ไม่ถึง 1000 จะมีผลกระทบในระดับสูง และหากอัตราส่วนดังกล่าวมีค่ามากกว่า 1000 จะมีผลกระทบในระดับสูงมาก

ง) ความซับซ้อนของโครงการซอฟต์แวร์ (Complexity of the Product หรือ CPLX) เป็นการพิจารณาความซับซ้อนของโปรแกรมโดย หากโปรแกรมคำนวณง่ายๆ มีการอ่านเขียนข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ธรรมดา จะมีผลกระทบในระดับต่ำมาก หากโปรแกรมคำนวณมีความซับซ้อนเช่น สแควรูท จะมีผลกระทบในระดับต่ำ หากมีการทำงานที่แบ่งแยกเป็นโมดูลจะมีผลกระทบปานกลาง หากมีการเปลี่ยนโครงสร้างของข้อมูลที่ซับซ้อนจะมีผลกระทบสูง และหากโปรแกรมมีการเรียกซ้ำ (Recursive) มีการทำงานของ 3D รวมถึงระบบสมองกลผังตัว จะมีผลกระทบในระดับสูงมาก

จ) เอกสารโครงการซอฟต์แวร์สอดคล้องกับวัฏจักรการพัฒนาซอฟต์แวร์ (Documentation match to life cycle needs หรือ DOCU) เป็นการพิจารณาความต้องการจัดทำเอกสารโดยพิจารณาจาก RUSE เป็นหลัก หาก RUSE อยู่ในระดับปานกลาง DOCU ควรจะอยู่ในระดับปานกลางขึ้นไป และหาก RUSE อยู่ในระดับสูงมากหรือสูงพิเศษ DOCU ควรจะอยู่ในระดับสูงขึ้นไป

ฉ) ความต้องการการนำกลับมาใช้ใหม่ (Required reusability หรือ RUSE) เป็นการพิจารณาความพยายามที่จะต้องเพิ่มขึ้น ถึงการนำส่วนของโปรแกรม (component) ที่สร้างขึ้นกลับมาใช้อีก หากไม่มีการนำกลับมาใช้ของส่วนของโปรแกรมจะอยู่ในระดับต่ำ หากมีการนำกลับมาใช้ทั่วทั้งโปรแกรมจะอยู่ในระดับสูง หากมีการนำกลับมาใช้ทั่วทั้งสายผลิตภัณฑ์จะอยู่ในระดับสูงมาก

**หมวดหมู่ ปัจจัยทางด้านแพลตฟอร์ม (Platform attributes) (หนังสือของ Westfall ใช้คำว่าแพลตฟอร์มแทนคำว่าฮาร์ดแวร์)**

ช) ข้อจำกัดด้านประสิทธิภาพของระยะเวลาประมวลผล (Run-time Performance Constraints หรือ TIME) เป็นการประเมินข้อจำกัดด้านเวลาในกาประมวลผลซอฟต์แวร์ที่สร้างขึ้น หากว่ามีการประมวลผลต่ำกว่า 50% ของเวลาที่เครื่องพร้อมจะอยู่ในระดับปกติ หากว่ามีการประมวลผลมากกว่า 70% จะอยู่ในระดับสูง หากมากกว่า 85% จะอยู่ในระดับสูงมาก และ 95% จะอยู่ในระดับสูงพิเศษ

ซ) ความยากของแพลตฟอร์ม (Platform Difficulty หรือ PDIF) เป็นการพิจารณาความไม่แน่นอนของแพลตฟอร์มหากความไม่แน่นอนของแพลตฟอร์ม เสถียรมาก PDIF จะอยู่ในระดับต่ำมาก และหากความไม่แน่นอนความไม่แน่นอนของแพลตฟอร์มสูง PDIF จะอยู่ในระดับสูงพิเศษ

ฌ) ข้อจำกัดด้านหน่วยความจำ (Memory Constraints หรือ STOR) เป็นการประเมินข้อจำกัดของหน่วยเก็บ ถ้าใช้น้อยกว่า 50% ของหน่วยเก็บหลักจะอยู่ในระดับปานกลาง มากกว่า

70% ของหน่วยเก็บหลักจะอยู่ในระดับสูง มากกว่า 85% จะอยู่ในระดับสูงมาก และมากกว่า 95% จะอยู่ในระดับสูงพิเศษ

ญ) การเปลี่ยนแปลงของแพลตฟอร์ม (Platform Volatility หรือ PVOL) เช่นการเปลี่ยนแปลง OS หรือ DBMS โดยหากเปลี่ยนแปลงเล็กน้อยทุกเดือนจะอยู่ในระดับต่ำ หากมีการเปลี่ยนแปลงใหญ่ทุก 6 เดือนจะอยู่ในระดับปานกลาง หากเปลี่ยนแปลงใหญ่ทุก 6 เดือนและมีการเปลี่ยนแปลงเล็กน้อยทุกสัปดาห์จะอยู่ในระดับสูง หากมีเปลี่ยนแปลงใหญ่ทุก 6 เดือนและมีการเปลี่ยนแปลงเล็กน้อยทุก 2 วันจะอยู่ในระดับสูงมาก

**หมวดหมู่ ปัจจัยด้านบุคคล (Personnel attributes)**

ฎ) บุคลากรที่มีความสามารถ (Personnel Capability หรือ PERS) พิจารณาจากความสามารถของบุคลากรในทีม โดยขึ้นอยู่กับ ACAP และ PCAP หากเปอร์เซ็นต์ไทล์รวมของ ACAP กับ PCAP มีค่า 20% 35% 45% 55% 65% 75% 85% PERS จะมีค่า ต่ำที่สุด ต่ำมาก ต่ำ ปานกลาง สูง สูงมากและสูงพิเศษ ตามลำดับ

ฏ) บุคลากรที่มีประสบการณ์ (Personal Experience หรือ PREX) พิจารณาจากประสบการณ์งานประยุกต์แพลตฟอร์มและประสบการณ์ภาษา เครื่องมือ หากน้อยกว่า 9 เดือน 1 ปี 2 ปี 4 ปี 6 ปี จะมีค่า PREX ต่ำ ปานกลาง สูง สูงมาก และสูงพิเศษ ตามลำดับ

ฐ) ความสามารถในการคิดวิเคราะห์ (Analyst Capability หรือ ACAP) เป็นการพิจารณาความสามารถของนักวิเคราะห์ระบบ โดยการกำหนดระดับของความสามารถโดยเฉลี่ยของบุคลากรในทีมว่าความสามารถอยู่ในเปอร์เซ็นต์ไทล์ที่เท่าใด หากมีเปอร์เซ็นต์ไทล์ที่ 15 จะอยู่ในระดับต่ำมาก เปอร์เซ็นต์ไทล์ที่ 35 จะอยู่ในระดับต่ำ เปอร์เซ็นต์ไทล์ที่ 55 จะอยู่ในระดับปานกลาง เปอร์เซ็นต์ไทล์ที่ 75 จะอยู่ในระดับสูง และเปอร์เซ็นต์ไทล์ที่ 90 จะอยู่ในระดับสูงมาก

ฑ) ความสามารถด้านวิศวกรรมซอฟต์แวร์ (Software Engineering Capability หรือ PCAP) เป็นการพิจารณาความสามารถเฉลี่ยของบุคคลในทีมโดยไม่พิจารณาประสบการณ์ของคนนั้นโดยการพิจารณาเป็นเปอร์เซ็นต์ไทล์ เกณฑ์การพิจารณาเช่นเดียวกับ ACAP

ฒ) บุคลากรมีความต่อเนื่อง (Personal Continuity หรือ PCON) เป็นการประเมินจากอัตราการลาออกของบุคลากร หากอัตราการลาออกของบุคลากรต่ำกว่า 48% ต่อปี จะมีค่าต่ำมาก หาก 24% ต่อปีจะมีค่าต่ำ หาก 12% ต่อปี จะมีค่าปานกลาง หาก 6% ต่อปีจะมีค่าสูง และหากต่ำกว่า 3% ต่อปีจะมีค่าสูงมาก

ณ) ประสบการณ์ทางด้านแอปพลิเคชัน (Application Experience หรือ AEXP) เป็นการประเมินประสบการณ์ของทีมพัฒนา หากประสบการณ์ต่ำกว่า 2 เดือนจะมีค่าต่ำมาก หากประสบการณ์ 6 เดือนจะมีค่าต่ำ หากประสบการณ์ 1 ปี จะมีค่าปานกลาง หากประสบการณ์ 3 ปีจะมีค่าสูง และหากประสบการณ์มากกว่า 6 ปี จะมีค่าสูงมาก

ด) ประสบการณ์ทางด้านแพลตฟอร์ม (Platform Experience หรือ PEXP) เป็นการประเมินประสบการณ์ของทีมพัฒนากับแพลตฟอร์มที่ใช้ โดยมีเกณฑ์เดียวกันกับ AEXP

ต) ประสบการณ์ทางด้านภาษาโปรแกรมมิ่ง (Programming Language Experience หรือ LEXP) เป็นการประเมินประสบการณ์ของทีมพัฒนากับภาษาการเขียนโปรแกรมและเครื่องมือที่ใช้ หลักเกณฑ์เช่นเดียวกันกับ AEXP และ PEXP

**หมวดหมู่ ปัจจัยด้านโครงการซอฟต์แวร์ (Project Attributes)**

ถ) สิ่งอำนวยความสะดวก (Facilities หรือ FCIL) เป็นการพิจารณาสิ่งอำนวยความสะดวกในการพัฒนาหากมีสิ่งอำนวยความสะดวกขั้นพื้นฐานจะอยู่ในระดับต่ำ และหากมีสิ่งอำนวยความสะดวกเข้มข้น จะอยู่ในระดับสูงมาก

ท) การใช้งานเครื่องมือซอฟต์แวร์ (Use of Software Tools หรือ TOOL) เป็นการประเมินความสามารถของเครื่องมือซอฟต์แวร์ หากมีการบูรณาการของเครื่องมือต่ำมากจะอยู่ในระดับต่ำมาก หากมีการบูรณาการบ้างจะอยู่ในระดับต่ำ หากมีการบูรณาการปานกลาง จะอยู่ในระดับปานกลาง จนกระทั่งมีการบูรณาการอย่างดีทั้งกระบวนการและวิธีการ จะอยู่ในระดับสูงพิเศษ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ธ) การพัฒนาต่างสถานที่ (Multisite Development หรือ SITE) เป็นการพิจารณาสถานที่ในการพัฒนาซอฟต์แวร์ หากการทำงานระหว่างประเทศจะอยู่ในระดับต่ำมาก หากทำงานหลายบริษัทและใช้โทรศัพท์สื่อสารจะอยู่ในระดับต่ำ หากทำงานหลายบริษัทและใช้อีเมล์หรือข้อความจะอยู่ในระดับปานกลางหรือสูง หากทำงานในตึกหรืออาคารเดียวกันจะอยู่ในระดับสูงมาก และหากทำงานในสถานที่เดียวกันจะอยู่ในระดับสูงพิเศษ

น) ความต้องการทางด้านระยะเวลาการพัฒนา (Required Development Schedule หรือ SCED) เป็นการพิจารณาข้อจำกัดด้านกำหนดเวลา โดยหากเวลาเร่งมาก 75% ของเวลาปกติจะอยู่ในระดับต่ำมาก 85% 100% 130% 160% ของเวลาปกติจะอยู่ในระดับต่ำ ปานกลาง สูงและสูงมากตามลำดับ

ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ของ COCOMO II ทั้ง 22 ปัจจัย นั้นนำมาใช้ใน 2 ช่วง คือ ช่วงตัวแบบการออกแบบช่วงต้น ทั้งหมด 7 ปัจจัย และช่วงตัวแบบหลังการมีสถาปัตยกรรมทั้งหมด 17 ปัจจัย แสดงในตารางที่ 6

ตารางที่ 6 ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ของช่วงตัวแบบการออกแบบช่วงต้น และช่วงตัวแบบหลังการมีสถาปัตยกรรม (Westfall, 2009)

| | ตัวแบบการออกแบบช่วงต้น | ตัวแบบการออกแบบหลังการมีสถาปัตยกรรม |
|---|---|---|
| ปัจจัยทางด้านผลิตภัณฑ์ | RCPX | RELY, DATA, CPLX, DOCU |
| | RUSE | RUSE |
| ปัจจัยทางด้านแพลตฟอร์ม | PDIF | TIME, STOR, PVOL |
| ปัจจัยด้านบุคคล | PERS | ACAP, PCAP, PCON |
| | PREX | AEXP, PEXP, LTEX |
| ปัจจัยด้านโครงการซอฟต์แวร์ | FCIL | TOOL, SIRE |
| | SCED | SCED |

E คือ ค่าปัจจัยขยาย (Scale Factor) คำนวณได้จากปัจจัยที่ควรคำนึงเพิ่มเติม 5 ปัจจัย มีสูตรการหาค่า E ดังนี้

$$E = 0.91 + 0.01 \sum(SF)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เมื่อ SF คือ ปัจจัยที่ควรคำนึงเพิ่มเติม ดังนั้น $\sum$ (SF) ก็คือผลรวมของปัจจัยที่ควรคำนึงเพิ่มเติมทั้ง 5 ปัจจัย (ใช้ในช่วงตัวแบบการออกแบบช่วงต้น และช่วงตัวแบบหลังการมีสถาปัตยกรรม)

โดย ปัจจัยที่ควรคำนึงเพิ่มเติมทั้ง 5 มีดังต่อไปนี้

ตารางที่ 7 คำอธิบายของปัจจัยที่ควรคำนึงเพิ่มเติม (Kishore, 2001)

| ปัจจัยที่ควรคำนึงเพิ่มเติม | คำอธิบาย |
|---|---|
| ความคล้ายคลึงกับโครงการที่เคยพัฒนามาก่อน (Precedentedness ตัวย่อ PREC) | ความเหมือน/ใกล้เคียงกับโครงการซอฟต์แวร์ที่เคยพัฒนา |
| ความยืดหยุ่นในการพัฒนา (Development Flexibility ตัวย่อ FLEX) | ความสอดคล้องของความต้องการของซอฟต์แวร์กับรายละเอียดการเชื่อมต่อภายนอก (External Interface Specification) |
| สถาปัตยกรรมและการจัดการความเสี่ยง (Architecture/Risk Resolution ตัวย่อ RESL) | ความสำคัญด้านการออกแบบและความเสี่ยง |
| การทำงานร่วมกันเป็นทีม (Team Cohesion ตัวย่อ TEAM) | ความสัมพันธ์จากการทำงานร่วมกัน |
| วุฒิภาวะของกระบวนการทำงาน (Process Maturity ตัวย่อ PMAT) | ประเมินตามอัตราของ SEO CMM |

ซึ่งมีเกณฑ์ในการกำหนดค่าดังตารางที่ 8

ตารางที่ 8 ค่าที่เหมาะสมสำหรับการกำหนดค่าให้กับปัจจัยที่ควรคำนึงเพิ่มเติม (อานุภาพไตรรงค์, 2558)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

| ปัจจัยที่ควรคำนึงเพิ่มเติม | ต่ำมาก (Very Low) | ต่ำ (Low) | ปานกลาง (Normal) | สูง (High) | สูงมาก (Very High) | สูงพิเศษ (Extra High) |
|---|---|---|---|---|---|---|
| PREC | ไม่เคยพัฒนาโครงการลักษณะนี้มาก่อนเลย (Thoroughly unprecedented) | โครงการที่จะพัฒนาแตกต่างจากโครงการที่เคยพัฒนามาก (Largely unprecedented) | โครงการที่จะพัฒนาแตกต่างจากโครงการที่เคยพัฒนาเล็กน้อย (Somewhat unprecedented) | โครงการที่จะพัฒนาคล้ายคลึงกับโครงการที่พัฒนา (Generally familiar) | โครงการที่จะพัฒนาค่อนข้างคล้ายคลึงโครงการที่เคยพัฒนามาก (Largely familiar) | โครงการที่จะพัฒนาคล้ายคลึงโครงการที่เคยพัฒนาอย่างสมบูรณ์ (Thoroughly familiar) |
| FLEX | โครงการที่จะพัฒนาต้องตรงกับความต้องการและให้บริการการเข้าถึงเพื่อใช้งานจากโครงการภายนอกทุกประการ (Rigorous) | โครงการที่จะพัฒนาต้องตรงกับความต้องการและให้บริการการเข้าถึงเพื่อใช้งานจากโครงการภายนอก โดยปรับเปลี่ยนได้เท่าที่จำเป็น (Occasional relaxation) | โครงการที่จะพัฒนาต้องตรงกับความต้องการและให้บริการการเข้าถึงเพื่อใช้งานจากโครงการภายนอกโดยปรับเปลี่ยนได้ตามที่ตกลงกัน (Some relaxation) | โครงการที่จะพัฒนาต้องตรงกับความต้องการและให้บริการการเข้าถึงเพื่อใช้งานจากโครงการภายนอกแต่ปรับเปลี่ยนได้ตามความเหมาะสม (General conformity) | โครงการที่จะพัฒนาต้องตรงกับความต้องการและให้บริการการเข้าถึงเพื่อใช้งานจากโครงการภายนอกปรับเปลี่ยนได้แต่ต้องเป็นไปตามข้อกำหนด (Some conformity) | โครงการที่จะพัฒนาต้องตรงกับความต้องการและให้บริการการเข้าถึงเพื่อใช้งานจากโครงการภายนอกปรับเปลี่ยนได้แต่ต้องเป็นไปตามวัตถุประสงค์โครงการ (General goals) |

| ปัจจัยที่ควรคำนึงเพิ่มเติม | ต่ำมาก (Very Low) | ต่ำ (Low) | ปานกลาง (Normal) | สูง (High) | สูงมาก (Very High) | สูงพิเศษ (Extra High) |
|---|---|---|---|---|---|---|
| RESL | ระดับการบริหารจัดการความเสี่ยงน้อยมาก ประมาณ 20% (Little - 20%) | ระดับการบริหารจัดการความเสี่ยงน้อย ประมาณ 40% (Some - 40%) | ระดับการบริหารจัดการความเสี่ยงปานกลาง ประมาณ 60% (Often - 60%) | ระดับการบริหารจัดการความเสี่ยงสูง ประมาณ 75% (Generally - 75%) | ระดับการบริหารจัดการความเสี่ยงสูงมาก ประมาณ 90% Mostly (90%) | ระดับการบริหารจัดการความเสี่ยงสูงมาก ประมาณ 100% (Full - 100%) |
| TEAM | ทีมงานทำงานร่วมกันด้วยความยากลำบากมาก (Very difficult interactions) | ทีมงานทำงานร่วมกันด้วยความยากลำบากบ้าง (Some difficult interactions) | ทีมงานทำงานร่วมกันตามปกติ (Basically cooperative interactions) | ทีมงานร่วมมือกันทำงานค่อนข้างมาก (Largely cooperative) | ทีมงานร่วมมือกันทำงานสูงมาก (Highly cooperative) | ทีมงานทำงานร่วมมือกันได้อย่างราบรื่น (Seamless interactions) |
| PMAT | ไม่มีการจัดระเบียบหรือมาตรฐานในการทำงาน CMM Level 1 | มีมาตรฐานในการทำงานระดับ CMM Level 1 ระดับบนหรือเทียบเท่า | มีมาตรฐานในการทำงานระดับ CMM Level 2 หรือเทียบเท่า | มีมาตรฐานในการทำงานระดับ CMM Level 3 หรือเทียบเท่า | มีมาตรฐานในการทำงานระดับ CMM Level 3 หรือเทียบเท่า | มีมาตรฐานในการทำงานระดับ CMM Level 5 หรือเทียบเท่า |

Chulalongkorn University จุฬาลงกรณ์มหาวิทยาลัย

2) ฟังก์ชันพอยต์ (Function Point)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์วิธีนี้ เป็นวิธีการประมาณขนาดของซอฟต์แวร์ พัฒนาขึ้นโดย Allan J. Albrecht ซึ่งผลลัพธ์ที่ได้จากการประมาณขนาดของซอฟต์แวร์จะมีหน่วยที่เรียกว่า ฟังก์ชันพอยต์ (Function Point หรือ FP) ใช้ในการประมาณค่าความพยายามในการพัฒนาซอฟต์ต่อไป สำหรับการประเมินขนาดของซอฟต์แวร์ด้วยฟังชันพอยต์นั้น จะพิจารณาองค์ประกอบหลัก 5 องค์ประกอบด้วยกันคือ

ก) ข้อมูลนำเข้าจากภายนอก (External Inputs หรือ EI)

จำนวนข้อมูลนำเข้า โดยพิจารณาจากข้อมูลนำเข้าที่มาจากหน้าจอหรือข้อมูลนำเข้าที่มาจากแอปพลิเคชันอื่นๆ หรือเป็นข้อมูลที่มาจากการประมวลผลข้อมูล ที่มาจากการ เพิ่ม/ลบ/เปลี่ยนแปลง ข้อมูลตรรกะภายใน (Internal Logical Files หรือ ILF)

ข) ข้อมูลส่งออกไปยังภายนอก (External Outputs หรือ EO)

จำนวนข้อมูลส่งออก จากภายในแอปพลิเคชันสู่แอปพลิเคชันอื่นๆ

ค) การสอบถามจากภายนอก (External Inquiry หรือ EQ)

การที่ผู้ใช้ป้อนข้อมูลเพื่อต้องการค้นหารายละเอียดต่างๆ โดยไม่มีการเปลี่ยนแปลงแฟ้มข้อมูลภายใน

ง) ไฟล์ตรรกะภายใน (Internal Logical Files หรือ ILF’s)

ข้อมูลที่เก็บในไฟล์ ผู้ใช้งานสามารถเพิ่ม/ลบ/เปลี่ยนแปลงข้อมูลนั้นได้ เช่น การใช้คำสั่งแก้ไขข้อมูลในฐานข้อมูล เป็นต้น

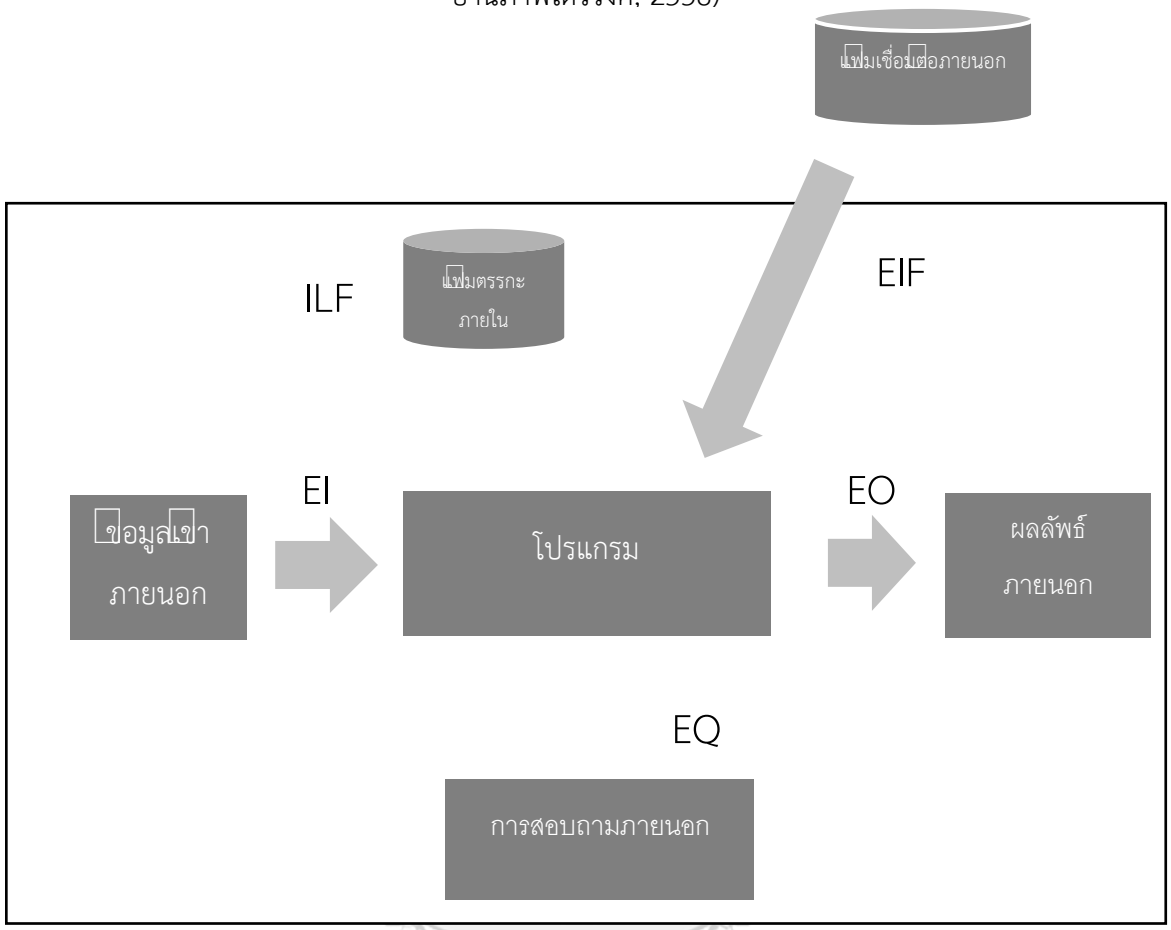
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จ) ไฟล์เชื่อมกับภายนอก (External Interface Files หรือ EIF’s)

แฟ้มข้อมูลที่มาจากข้างนอกระบบ แอปพลิเคชันทำหน้าที่เพียงแค่เรียกใช้งานข้อมูลจากไฟล์เหล่านั้น ผู้ใช้งานระบบหรือแอปพลิเคชันไม่สามารถแก้ไขข้อมูลได้โดยตรง ข้อมูลเหล่านั้นจะสามารถแก้ไขได้จากนอกระบบหรือผู้ใช้งานที่ใช้งานอยู่นอกระบบเท่านั้น

โดยความสัมพันธ์ขององค์ประกอบทั้ง 5 แสดงดังภาพที่ 1

ภาพที่ 1 ความสัมพันธ์ขององค์ประกอบทั้ง 5 ในการประเมินค่าความพยายามด้วยฟังก์ชันพอยต์ (ถาวร อานภาพไตรรงค์, 2558)




ในการประเมินค่าความพยายามด้วยวิธีนี้จะสามารถใช้องค์ประกอบทั้ง 5 มาวิเคราะห์ความซับซ้อนของค่าฟังก์ชันพอยต์ได้ ดังนี้

การประเมินระดับความซับซ้อนขององค์ประกอบทั้ง 5 มีระดับความซับซ้อนอยู่ 3 ระดับ คือ ระดับต่ำหรือไม่ซับซ้อน (Low or Simple) ระดับปานกลาง (Average) ระดับสูงหรือซับซ้อน (High or Complex) โดย EI EO และ EQ จะประเมินระดับความซับซ้อนขึ้นอยู่กับจำนวนไฟล์ที่มีการอัพเดทหรือใช้ในการอ้างถึง (File Type Reference ตัวย่อคือ FTR) และประเภทองค์ประกอบของข้อมูล (Data Element Type หรือตัวย่อ DET) ส่วน ILF และ EIF จะมีการกำหนดระดับความซับซ้อน ตามจำนวนตารางของข้อมูล (Record Element Type หรือ RET) กับประเภทองค์ประกอบของข้อมูล (Data Element Type ตัวย่อคือ DET) การให้ระดับความซับซ้อนในกระบวนการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีฟังก์ชันพอยต์ จะอ้างอิงตามตารางที่ 9 ดังนี้

ตารางที่ 9 การคิดค่าความซับซ้อนของ EI (Solution., 2004)

| จำนวนไฟล์ที่ใช้ในการอ้างถึง (FTR's) | ประเภทองค์ประกอบของข้อมูล (Data Element) | | |
|---|---|---|---|
| | 1-4 | 5-15 | >15 |
| 0-1 | ต่ำ (Low) | ต่ำ (Low) | ปานกลาง (Average) |
| 2 | ต่ำ (Low) | ปานกลาง (Average) | สูง (High) |
| 3 ขึ้นไป | ปานกลาง (Average) | สูง (High) | สูง (High) |

ตารางที่ 10 การคิดค่าความซับซ้อนของ EO และ EQ (Solution., 2004)

| จำนวนไฟล์ที่ใช้ในการอ้างถึง (FTR's) | ประเภทองค์ประกอบของข้อมูล (Data Element) | | |
|---|---|---|---|
| | <6 | 6-19 | >19 |
| 0-1 | ต่ำ (Low) | ต่ำ (Low) | ปานกลาง (Average) |
| 2-3 | ต่ำ (Low) | ปานกลาง (Average) | สูง (High) |
| >3 | ปานกลาง (Average) | สูง (High) | สูง (High) |

ตารางที่ 11 การคิดค่าความซับซ้อนของ ILF และ EIF (Solution., 2004)

| RET's | ประเภทองค์ประกอบของข้อมูล (Data Element) | | |
|---|---|---|---|
| | 1-19 | 20-50 | >50 |
| 1 | ต่ำ (Low) | ต่ำ (Low) | ปานกลาง (Average) |
| 2-5 | ต่ำ (Low) | ปานกลาง (Average) | สูง (High) |
| >5 | ปานกลาง (Average) | สูง (High) | สูง (High) |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิธีการคิดระดับความซับซ้อนจะอ้างอิงจากตารางข้างต้น เช่น หากนับจำนวนไฟล์ที่มีการอัพเดทหรือใช้ในการอ้างถึง 2 ไฟล์ (FTR's) และมี 10 ประเภทองค์ประกอบข้อมูล (DET's) จะจัดว่า EI มีระดับความซับซ้อนอยู่ในเกณฑ์ปานกลาง (พิจารณาจาก FTR = 2 และ DET = 10 จากตารางข้างต้น)

ตารางที่ 12 ค่าตัวคูณปัจจัย (Factor Multiplier) ของค่าความซับซ้อนองค์ประกอบทั้ง 5 คือ ILF EIF EI EO และ EQ ในการประเมินค่าความพยายามด้วยวิธีฟังก์ชันพอยต์ Function Point (Menguy, 2001)

| | ต่ำ (Low) | ปานกลาง (Average) | สูง (High) |
|---|---|---|---|
| ILF | 7 | 10 | 15 |
| EIF | 5 | 7 | 10 |
| EI | 3 | 4 | 6 |
| EO | 4 | 5 | 7 |
| EQ | 3 | 4 | 6 |

เมื่อได้ระดับความซับซ้อนมาแล้ว จะนำระดับความซับซ้อนไปอ้างอิงกับตารางข้างต้นเพื่อหาค่าความซับซ้อนจะเป็นตัวคูณของค่าองค์ประกอบทั้ง 5 เช่น จากระดับความซับซ้อนของการหา EI ที่อยู่ในระดับปานกลาง (Average) เมื่อนำไปอ้างอิงในตาราง พบว่ามีค่าความซับซ้อนอยู่ที่ 4 ซึ่งค่าความซับซ้อนข้างต้นจะเป็นผลคูณของค่า EI ที่หาได้ จะได้ผลลัพธ์มาเป็นจำนวนฟังก์ชันพอยต์ที่เกิดจากค่า EI นั่นคือ

จำนวนฟังก์ชันพอยต์ที่เกิดจากค่า EI ในตัวอย่าง จะมีค่าเท่ากับ 2 x 4 = 8 ฟังก์ชันพอยต์ ผู้ประเมินจะทำการหาค่าฟังก์ชันพอยต์จนครบทุกองค์ประกอบ (EI, EO, EQ, ILF, EIF) ค่าฟังก์ชันพอยต์ของซอฟต์แวร์จะเท่ากับ ผลรวมของค่าฟังก์ชันพอยต์ที่ได้จากแต่ละองค์ประกอบ ค่าฟังก์ชันพอยต์ที่คำนวณได้จะนำมาคูณกับค่าในตารางภาษาของฟังก์ชันพอยต์ (Function point Language Table) แสดงตารางในภาคผนวก ก. (Management., 2017)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิธีการหาฟังชันพอยต์ (Function Point) ข้างต้น จะเป็นการหา ฟังก์ชันพอยต์ (Function Point) ที่เรียกว่าฟังก์ชันพอยต์ไม่ปรับค่า (Unadjusted Function Point หรือ UFP) ซึ่งถ้าหากต้องการหาขนาดฟังก์ชันพอยต์ (Function Point) ที่สอดคล้องกับซอฟต์แวร์ที่ต้องการประเมินมากยิ่งขึ้น จำเป็นต้องคำนวณ ปัจจัยอื่นๆ ที่ส่งผลต่อขนาดของซอฟต์แวร์ เพื่อให้ได้ฟังก์ชันพอยต์ปรับค่า (Adjusted Function Point)

วิธีการคำนวณ ฟังก์ชันพอยต์ปรับค่า (Adjusted Function Point) จะเกิดจากการคูณระหว่าง ฟังก์ชันพอยต์ไม่ปรับค่า (Unadjusted Function Point หรือ UFP) กับ ปัจจัยความซับซ้อนเชิงเทคนิค (Technical Complexity Factors หรือ TCF)

ฟังก์ชันพอยต์ปรับค่า (Adjusted Function Point) = UFP x TCF

ซึ่ง ปัจจัยความซับซ้อนเชิงเทคนิค (Technical Complexity Factors หรือ TCF) คือค่าของผลรวมปัจจัยที่ส่งผลต่อขนาดของซอฟต์แวร์ มีวิธีการคำนวณหา TCF ดังนี้

TCF = 0.65 + 0.01 x (ผลรวมของค่าความซับซ้อนจากปัจจัยทั้ง 14 ปัจจัย)

โดย ปัจจัยที่ส่งผลต่อขนาดของซอฟต์แวร์ทั้ง 14 ปัจจัย มีดังนี้

ปัจจัยที่ส่งผลต่อการปรับค่าของฟังก์ชันพอยต์ (Value Adjustment Factor)

สามารถคำนวณได้จากปัจจัยทั้ง 14 ปัจจัยดังนี้ (อานุภาพไตรรงค์, 2558)

1. การสื่อสารข้อมูล (Data Communication)

ปัจจัยนี้จะพิจารณาถึงการรับและส่งข้อมูลกับส่วนต่างๆ มีเกณฑ์ดังต่อไปนี้

ตารางที่ 13 เกณฑ์การพิจารณาด้านการสื่อสารข้อมูล

| คะแนน | คำอธิบายลำดับขั้นปัจจัยการสื่อสารข้อมูล |
|---|---|
| 0 | งานประยุกต์นั้นประมวลผลแบบกลุ่ม (Batch) ล้วนๆ หรือเป็นแบบ Standalone Pc |
| 1 | งานประยุกต์นั้นเป็นแบบกลุ่ม (Batch) แต่มีการบันทึกข้อมูล หรือ การสั่งพิมพ์ข้อมูลระยะไกล |
| 2 | งานประยุกต์นั้นเป็นแบบกลุ่ม (Batch) แต่มีการบันทึกข้อมูล และ การสั่งพิมพ์ระยะไกล |
| 3 | งานประยุกต์นั้นมีการเก็บข้อมูลออนไลน์ หรือ การประมวลผลทางไกลจากส่วนหน้าสู่ระบบกลุ่ม (Batch) หรือ ระบบสอบถาม |
| 4 | งานประยุกต์นั้นเป็นมากกว่าการประมวลผลส่วนหน้า แต่สนับสนุนโพรโตคอลการสื่อสารการประมวลผลทางไกลเพียงแบบเดียว |
| 5 | งานประยุกต์นั้นเป็นมากกว่าการประมวลผลส่วนหน้า และ สนับสนุนโพรโตคอลการสื่อสารการประมวลผลทางไกลมากกว่าเพียงแบบเดียว |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2. การประมวลผลข้อมูลแบบกระจาย (Distributed Data Processing)

ปัจจัยนี้จะพิจารณาถึงการออกแบบการกระจายหน่วยประมวลผลของระบบ มีเกณฑ์ดังต่อไปนี้

ตารางที่ 14 เกณฑ์การพิจารณาด้านการประมวลผลข้อมูลแบบกระจาย

| คะแนน | คำอธิบายระดับขั้นปัจจัยการประมวลผลข้อมูลแบบกระจาย |
|---|---|
| 0 | งานประยุกต์นั้นไม่ช่วยในการถ่ายข้อมูลหรือการประมวลผลระหว่างส่วนประกอบของระบบ |
| 1 | งานประยุกต์นั้นเตรียมข้อมูลสำหรับการประมวลผลของผู้ใช้สำหรับส่วนประกอบอื่น |
| 2 | มีการเตรียมข้อมูลเพื่อถ่ายโอน และประมวลผลบนส่วนประกอบอื่นของระบบ |
| 3 | การประมวลผลแบบกระจายและการถ่ายโอนข้อมูลเป็นแบบออนไลน์แต่เป็นในทิศทางเดียว |
| 4 | การประมวลผลแบบกระจายและการถ่ายโอนข้อมูลเป็นแบบออนไลน์และเป็นแบบสองทิศทาง |
| 5 | ฟังก์ชันการประมวลผลข้อมูลนั้นกระทำแบบพลวัตบนส่วนประกอบที่เหมาะสมที่สุดของระบบ |

3. สมรรถนะ (Performance)

ปัจจัยนี้จะพิจารณาถึงประสิทธิภาพ ทั้งการรับส่งข้อมูล การออกแบบระบบ การพัฒนาระบบ การบริการสนับสนุน การติดตั้งระบบ มีเกณฑ์ดังต่อไปนี้

ตารางที่ 15 เกณฑ์การพิจารณาด้านสมรรถนะ

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านสมรรถนะ |
|---|---|
| 0 | ไม่มีการระบุความต้องการด้านสมรรถนะที่ผู้ใช้กำหนดเป็นพิเศษ |
| 1 | มีการระบุความต้องการด้านสมรรถนะและการออกแบบ และได้รับการทบทวนแต่ไม่มีการกระทำอะไรเป็นพิเศษ |
| 2 | เวลาตอบสนองและปริมาณงานเป็นเรื่องวิกฤติ เฉพาะช่วงที่มีการใช้งานสูงสุด |
| 3 | เวลาตอบสนองและปริมาณงานเป็นเรื่องวิกฤติ ตลอดเวลาที่มีการใช้งานของธุรกิจ |
| 4 | นอกเหนือจาก 3 ในเรื่องของความต้องการเข้มข้นด้านสมรรถนะของผู้ใช้ ทำให้ต้องการการวิเคราะห์สมรรถนะในการออกแบบ |
| 5 | นอกเหนือจาก 4 ในเรื่องการใช้เครื่องมือวิเคราะห์สมรรถนะในขั้นตอนการออกแบบ การพัฒนาและ/หรือ การติดตั้ง เพื่อที่จะให้ตรงกับสมรรถนะที่ผู้ใช้ต้องการ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4. ความต้องการใช้งานหนักด้านฮาร์ดแวร์ (Heavily Used Configuration)

ปัจจัยนี้จะพิจารณาถึงการดำเนินการทั้งหมดเกี่ยวกับการกำหนดค่าของระบบ มีเกณฑ์ดังต่อไปนี้

ตารางที่ 16 เกณฑ์การพิจารณาด้านความต้องการใช้งานหนักด้านฮาร์ดแวร์

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านความต้องการใช้งานหนักด้านฮาร์ดแวร์ |
|---|---|
| 0 | ไม่มีเรื่องของความเข้มงวดในการปฏิบัติงานได้ ทั้งอย่างชัดเจนหรือเป็นนัย |
| 1 | มีความเข้มงวดในการปฏิบัติงาน แต่ความเข้มงวดน้อยกว่างานประยุกต์ทั่วไป ไม่ต้องมีความพยายามพิเศษในการได้มาซึ่งความเข้มงวดนั้น |
| 2 | มีการพิจารณาเรื่องความปลอดภัยบางอย่างหรือข้อจำกัดด้านเวลาด้วย |
| 3 | มีความต้องการหน่วยประมวลผล เฉพาะเจาะจง สำหรับงานประยุกต์นี้ |
| 4 | มีการระบุข้อจำกัดความต้องการหน่วยประมวลผลกลางหรือ หน่วยประมวลผลเฉพาะสำหรับงานประยุกต์นี้ |
| 5 | นอกเหนือจาก 4 มีข้อจำกัดพิเศษในส่วนประกอบแบบกระจายของระบบสำหรับงานประยุกต์นี้ |

5. อัตรารายการเปลี่ยนแปลง (Transaction Rate)

ปัจจัยนี้จะพิจารณาถึงจำนวนธุรกรรมข้อมูล ซึ่งส่งผลการออกแบบ การติดตั้ง การพัฒนา และการบริการด้านการสนับสนุน มีเกณฑ์ดังต่อไปนี้

ตารางที่ 17 เกณฑ์การพิจารณาด้านอัตรารายการเปลี่ยนแปลง

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านอัตรารายการเปลี่ยนแปลง |
|---|---|
| 0 | ไม่คาดหมายว่าจะมีช่วงที่มีระดับสูงสุดของรายการเปลี่ยนแปลง |
| 1 | มีการคาดหมายว่าจะมีช่วงที่มีระดับสูงสุดของรายการเปลี่ยนแปลง |
| 2 | มีการคาดหมายว่าจะมีช่วงที่มีระดับสูงสุดของรายการเปลี่ยนแปลงเป็นรายสัปดาห์ |
| 3 | มีการคาดหมายว่าจะมีช่วงที่มีระดับสูงสุดของรายการเปลี่ยนแปลงรายวัน |
| 4 | อัตรารายการเปลี่ยนแปลงที่สูงซึ่งกำหนดโดยผู้ใช้ ในงานประยุกต์นั้น หรือในข้อตกลงระดับการบริการมากถึงระดับที่จะต้องการการวิเคราะห์สมรรถนะในขั้นตอนการออกแบบ |
| 5 | อัตรารายการเปลี่ยนแปลงที่สูงซึ่งกำหนดโดยผู้ใช้ ในงานประยุกต์นั้น หรือในข้อตกลงระดับการบริการมากถึงระดับที่จะต้องการวิเคราะห์สมรรถนะในขั้นตอนการออกแบบ |

CHULALONGKORN UNIVERSITY

6. การบันทึกข้อมูลออนไลน์ (On-line Data Entry)

ปัจจัยนี้จะพิจารณาถึงฟังก์ชันการจัดการข้อมูล ภายในแอปพลิเคชัน มีเกณฑ์ดังต่อไปนี้

ตารางที่ 18 เกณฑ์การพิจารณาด้านการบันทึกข้อมูลออนไลน์

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านการบันทึกข้อมูลออนไลน์ |
|---|---|
| 0 | การประมวลผลรายการเปลี่ยนแปลงทั้งหมดเป็นแบบกลุ่ม |
| 1 | 1%-7% ของรายการเปลี่ยนแปลงเป็นการบันทึกข้อมูลแบบเชิงโต้ตอบ |
| 2 | 8%-15% ของรายการเปลี่ยนแปลงเป็นการบันทึกข้อมูลแบบเชิงโต้ตอบ |
| 3 | 16%-23% ของรายการเปลี่ยนแปลงเป็นการบันทึกข้อมูลแบบเชิงโต้ตอบ |
| 4 | 24%-30% ของรายการเปลี่ยนแปลงเป็นการบันทึกข้อมูลแบบเชิงโต้ตอบ |
| 5 | >30% ของรายการเปลี่ยนแปลงเป็นการบันทึกข้อมูลแบบเชิงโต้ตอบ |

7. ประสิทธิภาพเพื่อผู้ใช้ขั้นปลาย (End-User Efficiency)

ปัจจัยนี้จะพิจารณาถึงการออกแบบให้ผู้ใช้สามารถใช้งานได้ง่าย มีเกณฑ์ดังต่อไปนี้

- เครื่องมือช่วยนำทาง
- เมนู
- คำอธิบายและเอกสารออนไลน์
- การเคลื่อนย้ายตัวชี้ตำแหน่งอัตโนมัติ (Automated cursor movement)
- การเลื่อนแถบเลื่อน (Scrolling)
- การพิมพ์จากระยะไกลผ่านรายการเปลี่ยนแปลงออนไลน์ (Remote printing via online transactions)
- การกำหนดหน้าที่ใหม่ให้กับแป้นกำหนดหน้าที่ (Reassigned function keys)
- งานประมวลผลแบบกลุ่มที่ส่งจากรายการเปลี่ยนแปลงแบบออนไลน์ (Batched job submitted from online transactions)
- การเลือกตัวชี้ตำแหน่งของข้อมูลจอภาพ (Cursor selection of screen data)
- การใช้งานอย่างมากด้านวิดีทัศน์ผันกลับ การเน้นให้เด่น การขีดเส้นใต้สี หรือตัวชี้บอกอื่น (Heavy use of reverse video, highlight, color underlining, and other indicators)
- เอกสารสำหรับผู้ใช้ฉบับพิมพ์ของรายการเปลี่ยนแปลงออนไลน์ (Hard copy user documentation of online transactions)
- ส่วนต่อประสานด้วยเมาส์ (Mouse interface)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- หน้าต่างปอบอัพ (Popup windows)
- จอภาพน้อยที่สุดเท่าที่จะเป็นได้ ที่จะทำให้งานของธุรกิจสำเร็จ (As few screen as possible to accomplish a business function)
- ส่วนสนับสนุนการใช้งานสองภาษา (Bilingual support)
- ส่วนสนับสนุนการใช้งานหลายภาษา (Multilingual support)

ตารางที่ 19 เกณฑ์การพิจารณาด้านประสิทธิภาพเพื่อผู้ใช้ขั้นปลาย

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านประสิทธิภาพเพื่อผู้ใช้ขั้นปลาย |
|---|---|
| 0 | ไม่มีข้อกำหนดข้างต้น |
| 1 | มีข้อกำหนด 1-3 ข้อจากข้อกำหนดข้างต้น |
| 2 | มีข้อกำหนด 4-5 ข้อจากข้อกำหนดข้างต้น |
| 3 | มีข้อกำหนด 6 ข้อหรือมากกว่าจากข้อกำหนดข้างต้น แต่ผู้ใช้ไม่มีความต้องการเฉพาะเกี่ยวกับประสิทธิภาพ |
| 4 | มีข้อกำหนด 6 ข้อหรือมากกว่าจากข้อกำหนดข้างต้น และมีความต้องการของผู้ใช้เกี่ยวกับประสิทธิภาพสูงจนต้องการงานด้านมนุษย์ปัจจัยรวมเข้ามาด้วย |
| 5 | มากกว่า 6 ข้อจากข้อกำหนดข้างต้น และมีความต้องการของผู้ใช้เกี่ยวกับประสิทธิภาพสูง จนต้องการใช้เครื่องมือพิเศษและกระบวนการที่จะแสดงให้เห็นว่าได้บรรลุวัตถุประสงค์แล้ว |

8. การปรับให้เป็นปัจจุบันแบบออนไลน์ (On-line Update)

ปัจจัยนี้จะพิจารณาถึง การเปิดบริการให้อัพเดทข้อมูลภายในจากภายนอก มีเกณฑ์ดังต่อไปนี้

ตารางที่ 20 เกณฑ์การพิจารณาด้านการปรับให้เป็นปัจจุบันแบบออนไลน์

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านการปรับให้เป็นปัจจุบันแบบออนไลน์ |
|---|---|
| 0 | ไม่มีแฟ้มที่ต้องปรับปรุงให้เป็นปัจจุบันแบบออนไลน์ |
| 1 | มีแฟ้มที่ต้องปรับปรุงให้เป็นปัจจุบันแบบออนไลน์ 1-3 แฟ้ม ปริมาณการปรับให้เป็นปัจจุบันน้อยและการกู้คืนง่าย |
| 2 | มีแฟ้มที่ต้องปรับให้เป็นปัจจุบันแบบออนไลน์ ตั้งแต่ 4 แฟ้ม ปริมาณการปรับให้เป็นปัจจุบันน้อยและการกู้คืนง่าย |
| 3 | มีแฟ้มตรรกะภายในหลักที่ต้องปรับให้เป็นปัจจุบันแบบแแนไลน์รวมอยู่ด้วย |

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านการปรับให้เป็นปัจจุบันแบบออนไลน์ |
|---|---|
| 0 | ไม่มีแฟ้มที่ต้องปรับปรุงให้เป็นปัจจุบันแบบออนไลน์ |
| 4 | นอกเหนือจาก 3 การป้องกันข้อมูลสูญหาย มีความสำคัญและมีการออกแบบ |
| 5 | จากเหนือจาก 4 ปริมาณงานที่มาก ทำให้ต้องมีการนำเรื่องต้นทุนมาพิจารณาในกระบวนการกู้คืน รวมทั้งมีกระบวนการกู้คืนแบบอัตโนมัติที่มีมากโยการแทรกแซงด้วยผู้ควบคุมเครื่องน้อยที่สุด |

9. การประมวลผลที่ซับซ้อน (Complex Processing)

ปัจจัยนี้จะพิจารณาถึงความซับซ้อนของฟังก์ชันในการประมวลผลของแอปพลิเคชัน มีเกณฑ์ดังต่อไปนี้

-การควบคุมอ่อนไหว (Sensitive Control) หรือการประมวลผลเฉพาะเพื่อความปลอดภัยของงานประยุกต์นั้น

-มีตรรกะการประมวลผลที่ครอบคลุมกว้างขวาง

-มีการประมวลผลทางคณิตศาสตร์ที่ครอบคลุมกว้างขวาง

-การประมวลที่ผิดปกติ ทำให้เกิดรายการเปลี่ยนแปลงที่ไม่สมบูรณ์ และต้องทำการประมวลผลอีกครั้ง

-มีการประมวลที่ซับซ้อนเพื่อจัดการกับข้อมูลเข้า และผลลัพธ์เป็นพหุคูณ เช่นสื่อประสมหรือความเป็นอิสระกับอุปกรณ์

ตารางที่ 21 เกณฑ์การพิจารณาด้านการประมวลผลที่ซับซ้อน

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านการประมวลผลที่ซับซ้อน |
|---|---|
| 0 | ไม่มีคุณลักษณะดังกล่าวข้างต้น |
| 1 | มีคุณสมบัติดังกล่าวข้างต้นข้อใดก็ได้ 1 ข้อ |
| 2 | มีคุณสมบัติดังกล่าวข้างต้นข้อใดก็ได้ 2 ข้อ |
| 3 | มีคุณสมบัติดังกล่าวข้างต้นข้อใดก็ได้ 3 ข้อ |
| 4 | มีคุณสมบัติดังกล่าวข้างต้นข้อใดก็ได้ 4 ข้อ |
| 5 | มีคุณสมบัติดังกล่าวข้างต้นข้อใดก็ได้ 5 ข้อ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

10. ความสามารถนำกลับมาใช้ใหม่ (Reusability)

ปัจจัยนี้จะพิจารณาถึง การออกแบบและการพัฒนาซอฟต์แวร์ให้สามารถประยุกต์ใช้กับแอปพลิเคชันอื่นๆได้ มีเกณฑ์ดังต่อไปนี้

ตารางที่ 22 เกณฑ์การพิจารณาด้านความสามารถนำกลับมาใช้ใหม่

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านความสามารถนำกลับมาใช้ใหม่ |
|---|---|
| 0 | ไม่มีรหัสคำสั่งที่สามารถนำกลับมาใช้ใหม่ได้ |
| 1 | มีรหัสที่นำกลับมาใช้ใหม่ภายในงานประยุกต์นั้น |
| 2 | น้อยกว่า 10% ของงานประยุกต์นั้นได้มีการพิจารณาความต้องการของผู้ใช้อื่น |
| 3 | 10% หรือมากกว่าของงานประยุกต์นั้นได้มีการพิจารณาความต้องการของผู้ใช้อื่น |
| 4 | งานประยุกต์นั้นพัฒนารวมกันเป็นชุด และ/หรือ จัดทำเอกสารประกอบเพื่อสามารถนำกลับมาใช้ใหม่ได้ง่าย และงานประยุกต์นั้นสามารถทำตามต้องการของผู้ใช้โดยผู้ใช้ในระดับรหัสต้นฉบับ |
| 5 | งานประยุกต์นั้นพัฒนารวมกันเป็นชุด และ/หรือ จัดทำเอกสารประกอบเพื่อสามารถนำกลับมาใช้ใหม่ได้ง่าย และงานประยุกต์นั้นสามารถทำตามต้องการของผู้ใช้ด้วยผู้ใช้ โดยการเปลี่ยนพารามิเตอร์ |

11. ความสะดวกในการติดตั้ง (Installation Ease)

ปัจจัยนี้จะพิจารณาถึงการออกแบบให้ง่ายต่อการติดตั้งซอฟต์แวร์หรือติดตั้งระบบ มีเกณฑ์ดังต่อไปนี้

ตารางที่ 23 เกณฑ์การพิจารณาด้านความสะดวกในการติดตั้ง

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านความสะดวกในการติดตั้ง |
|---|---|
| 0 | ไม่มีลักษณะพิเศษในการติดตั้ง |
| 1 | ไม่มีลักษณะพิเศษในการติดตั้ง แต่มีความต้องการในการจัดเตรียมการติดตั้ง |
| 2 | มีการกำหนดความต้องการในการปรับเปลี่ยนและการติดตั้งจากผู้ใช้ และมีการจัดทำคู่มือในการติดตั้งและปรับเปลี่ยนที่ผ่านการตรวจสอบแล้ว ผลกระทบโครงการจากการปรับเปลี่ยนไม่มีความสำคัญ |
| 3 | มีการกำหนดความต้องการในการปรับเปลี่ยนและการติดตั้งจากผู้ใช้ และมีการจัดทำคู่มือในการติดตั้งและปรับเปลี่ยนที่ผ่านการตรวจสอบแล้ว ผลกระทบโครงการจากการปรับเปลี่ยนมีความสำคัญ |
| 4 | เพิ่มจาก 3 โดยมีการจัดหาและทดสอบเครื่องมือในการติดตั้งและปรับเปลี่ยนแปลงอัตโนมัติ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านความสะดวกในการติดตั้ง |
|---|---|
| 5 | เพิ่มจาก 4 โดยมีการจัดหาและทดสอบเครื่องมือในการติดตั้งและปรับเปลี่ยนแปลงอัตโนมัติ |

12. ความสะดวกในการปฏิบัติงาน (Operational Ease)

ปัจจัยนี้จะพิจารณาถึง การดำเนินการอื่นๆของแอปพลิเคชัน เช่น การสำรองข้อมูล, การกู้คืนข้อมูล มีเกณฑ์ดังต่อไปนี้

-มีการจัดการเริ่มต้นปฏิบัติงาน (Start-Up) การสำรองข้อมูล (Back-Up) และกู้คืน (Recovery) ให้ แต่ยังต้องมีการแทรกแซงของผู้ควบคุมเครื่องเข้า (1 คะแนน)

- มีการจัดการเริ่มต้นปฏิบัติงาน (Start-Up) การสำรองข้อมูล (Back-Up) และกู้คืน (Recovery) ให้ ไม่ต้องมีการแทรกแซงของผู้ควบคุมเครื่องเข้า (2 คะแนน)

-งานประยุกต์นั้นมีการใส่เทปน้อยที่สุด (1 คะแนน)

-งานประยุกต์นั้น มีการจัดการกระดาษน้อยที่สุด (1 คะแนน)

ตารางที่ 24 เกณฑ์การพิจารณาด้านความสะดวกในการปฏิบัติงาน

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านความสะดวกในการปฏิบัติงาน |
|---|---|
| 0 | ไม่มีการกำหนดความต้องการพิเศษจากผู้ใช้ นอกจากขั้นตอนการสำรองข้อมูลตามปกติ |
| 1 | คะแนนตามจำนวนข้อลักษณะพิเศษข้างต้น |
| 2 | คะแนนตามจำนวนข้อลักษณะพิเศษข้างต้น |
| 3 | คะแนนตามจำนวนข้อลักษณะพิเศษข้างต้น |
| 4 | คะแนนตามจำนวนข้อลักษณะพิเศษข้างต้น |
| 5 | การจัดการเริ่มต้นปฏิบัติงาน (Start-Up) การสำรองข้อมูล (Back-Up) และกู้คืน (Recovery) ให้ ไม่ต้องมีการแทรกแซงของผู้ควบคุมเครื่องเข้า และการปิดเครื่องงานประยุกต์สามารถจัดการกู้คืนความผิดพลาดได้เองอัตโนมัติ |

13. หลายสถานที่ (Multiple Sites)

ปัจจัยนี้จะพิจารณาถึงการออกแบบ การพัฒนาและการดำเนินการสนับสนุนที่จะติดตั้งในหลายๆ ที่ขององค์กร มีเกณฑ์ดังต่อไปนี้

ตารางที่ 25 เกณฑ์การพิจารณาด้านหลายสถานที่

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านหลายสถานที่ |
|---|---|
| 0 | ผู้ใช้ไม่มีความต้องการงานประยุกต์ให้ใช้งานมากกว่าผู้ใช้หนึ่งรายได้ หรือหลายสถานที่ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านหลายสถานที่ |
|---|---|
| 1 | มีการพิจารณาความจำเป็นที่จะต้องใช้งานหลายสถานที่ในขั้นตอนการออกแบบ และงานประยุกต์นั้นออกแบบมาให้ใช้ได้ภายใต้เฉพาะสิ่งแวดล้อมฮาร์ดแวร์และซอฟต์แวร์เดียวกัน |
| 2 | มีการพิจารณาความจำเป็นที่จะต้องใช้งานหลายสถานที่ฝนขั้นตอนการออกแบบ และงานประยุกต์นั้นออกแบบมาให้ใช้ได้ภายใต้สิ่งแวดล้อมฮาร์ดแวร์และซอฟต์แวร์ที่คล้ายกัน |
| 3 | มีการพิจารณาความจำเป็นที่จะต้องใช้งานหลายสถานที่ ในขั้นการออกแบบและงานประยุกต์นั้นออกแบบมาให้ใช้ได้ภายใต้สิ่งแวดล้อมฮาร์ดแวร์และซอฟต์แวร์ที่แตกต่างกัน |
| 4 | มีการจัดทำและทดสอบเอกสารประกอบและแผนสนับสนุน งานประยุกต์สำหรับใช้งานหลายสถานที่ งานประยุกต์นั้นเป็นแบบข้อ 1 หรือ ข้อ 2 |
| 5 | มีกาจัดทำและการทดสอบเอกสารประกอบและแผนสนับสนุน งานประยุกต์สำหรับใช้งานหลายสถานที่ งานประยุกต์นั้นเป็นแบบข้อ 3 |

14. คำอธิบายระดับขั้นปัจจัยอำนวยความสะดวกในการเปลี่ยนแปลง (Facilitate Change)

ปัจจัยนี้จะพิจารณาถึงการออกแบบแอปพลิเคชันให้รองรับการเปลี่ยนแปลงที่อาจเกิดขึ้น มีเกณฑ์ดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

-มีการสอบถาม (Query) ที่ยืดหยุ่นและสิ่งอำนวยความสะดวกในการออกแบบรายงานเพื่อให้สามารถจัดทำคำขออย่างง่ายๆได้ เช่น ตรรกะแบบ and/or ซึ่งใช้งานกับเพียง 1 Internal Logical File (1 คะแนน)

-มีการสอบถาม (Query) ที่ยืดหยุ่นและสิ่งอำนวยความสะดวกในการออกแบบรายงานเพื่อให้สามารถจัดทำคำขอที่ซับซ้อนปานกลางได้ เช่น ตรรกะแบบ and/or ซึ่งใช้งานมากกว่า 1 Internal Logical File (2 คะแนน)

-มีการสอบถาม (Query) ที่ยืดหยุ่นและสิ่งอำนวยความสะดวกในการออกแบบรายงานเพื่อให้สามารถจัดทำคำขอที่ซับซ้อนปานกลางได้ เช่น ตรรกะแบบ and/or ที่รวมกันมากกว่า 1 หรือใช้งานมากกว่า 1 Internal Logical File (1 คะแนน)

-ข้อมูลที่ใช้ในการควบคุมธุรกิจนั้นจะถูกเก็บไว้ในตารางโดยผู้ใช้จะทำการปรับปรุงผ่านกระบวนการปฏิสัมพันธ์แบบออนไลน์ การเปลี่ยนแปลงนี้จะมีผลในวันทำการถัดไป (1 คะแนน)

-ข้อมูลที่ใช้ในการควบคุมธุรกิจนั้นจะถูกเก็บไว้ในตารางโดยผู้ใช้จะทำการปรับปรุงผ่านกระบวนการปฏิสัมพันธ์แบบออนไลน์ การเปลี่ยนแปลงนี้จะมีผลทันที (2 คะแนน)

ตารางที่ 26 เกณฑ์การพิจารณาด้านอำนวยความสะดวกในการเปลี่ยนแปลง

| คะแนน | คำอธิบายระดับขั้นปัจจัยด้านอำนวยความสะดวกในการเปลี่ยนแปลง |
|---|---|
| 0 | ไม่มีคุณลักษณะใดตามที่กล่าวมาข้างต้น |
| 1 | มีคุณลักษณะตามที่กล่าวมาข้างต้น 1 ข้อ |
| 2 | มีคุณลักษณะตามที่กล่าวมาข้างต้น 2 ข้อ |
| 3 | มีคุณลักษณะตามที่กล่าวมาข้างต้น 3 ข้อ |
| 4 | มีคุณลักษณะตามที่กล่าวมาข้างต้น 4 ข้อ |
| 5 | มีคุณลักษณะตามที่กล่าวมาข้างต้น 5 ข้อ |

ผู้ประเมินจะประเมินปัจจัยทั้ง 14 และกำหนดค่าให้กับปัจจัยที่ส่งผลกระทบต่อระบบ โดยกำหนดค่าได้ตั้งแต่ 1-5 ดังนั้นค่าสูงสุดของผลรวม Adjustment Factor จะมีค่าได้เท่ากับ 70 และค่าต่ำสุดจะมีค่าได้เท่ากับ 0

**การหาค่าความพยายามในการพัฒนาซอฟต์แวร์จากฟังก์ชันพอยต์**

การหาค่าความพยายามในการพัฒนาซอฟต์แวร์จากฟังก์ชันพอยต์ ผู้ประเมินจะต้องหาฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ที่จะประเมินเสียก่อน แล้วจึงนำมาประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีใดวิธีหนึ่ง ดังตารางที่ 27

ตารางที่ 27 วิธีการคำนวณค่าความพยายามในการพัฒนาซอฟต์แวร์จากฟังก์ชันพอยต์

(อานุภาพไตรรงค์, 2558)

| ผู้วิจัย | วิธีการหาค่าความพยายาม |
|---|---|
| (Albrecht, 1983) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = FP / 0.0595 หรือ 16.81 x FP |
| (Behrens, 1983) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = 18.3 x FP |
| (Moser, 1996) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = 6.667 x FP |
| (Lokan, 2000) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = $21 \times FP^{(0.826)}$ |

| ผู้วิจัย | วิธีการหาค่าความพยายาม |
|---|---|
| (Maxwell, 2000) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = FP / 0.337 หรือ 2.97 x FP |
| (Jeffery R., 2001) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = 2.2 * FP |
| (Arnuphaptrirong, 2013) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (Effort) = FP / 1.09 หรือ 0.917 x FP |

3) ยูสเคสพอยต์ (Use Case Point)

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยการหาขนาดของซอฟต์แวร์ก่อนจึงค่อยหาค่าความพยายามในการพัฒนาซอฟต์แวร์เช่นเดียวกับ ฟังก์ชันพอยต์ (Function Point) โดยวิธีการประเมินจะใช้ยูสเคสไดอะแกรม (Use Case Diagram) มาช่วยในการประเมิน (Banerjee, 2001)

ขั้นตอนการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยยูสเคสพอยต์ทั้งหมดมีดังนี้

1) การหาค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight)
2) การหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Weights หรือ UUCW)
3) การหาค่ายูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Point)
4) การหาค่าความซับซ้อนทางเทคนิค (Technical Complexity Factor หรือ TCF)
    - การหาค่าปัจจัยทางเทคนิค (Technical Factor หรือ TFactor)
5) การหาค่าความซับซ้อนทางสิ่งแวดล้อม (Environment Complexity Factor หรือ ECF)
    - การหาค่าปัจจัยทางสิ่งแวดล้อม (Environment Factor หรือ EFactor)
6) การหาค่ายูสเคสพอยต์ที่ปรับค่าแล้ว (Adjusted Use Case Point หรือ UCP)
7) การคำนวณหาค่าความพยายามในการพัฒนาซอฟต์แวร์

**การหาค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight)**

ขั้นตอนแรกของการหายูสเคสพอยต์คือ การหาค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight หรือ UAW) โดยจะแบ่งผู้ประทำ (Actor) ของยูสเคส (Use Case) ออกเป็น 3 ประเภท คือ ไม่ซับซ้อน (Simple), ปานกลาง (Average) และ ซับซ้อน (Complex)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- แอคเตอร์แบบง่าย (Simple Actor) คือ ระบบอื่นๆ ที่มีการเชื่อมต่อกับขอบเขตของซอฟต์แวร์ที่จะประเมินผ่าน API (Application Programming Interface)
- แอคเตอร์แบบปานกลาง (Average Actor) คือ ระบบอื่นๆ ที่มีการเชื่อมต่อกับขอบเขตของซอฟต์แวร์ที่จะประเมิน ผ่านการการเชื่อมต่อทีซีพี/ไอพี (TCP/IP)
- ผู้ประทำแบบซับซ้อน (Complex Actor) คือ ผู้ใช้งานซอฟต์แวร์ที่มีการใช้งานซอฟต์แวร์ผ่านหน้าเว็บหรือหน้าต่างโปรแกรม

โดยน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่านั้นจะหาได้จากผลรวมของ จำนวนแอคเตอร์ที่แบ่งตามประเภทต่างๆ คูณด้วยค่าน้ำหนักแอคเตอร์ตามประเภทนั้นๆ ซึ่งถูกกำหนดไว้ดังแสดงในตารางที่ 28

ตารางที่ 28 ค่าน้ำหนัก (Weighting Factor) ของผู้แอคเตอร์ชนิดต่างๆ ในการประเมินค่าความพยายามด้วยยูสเคสพอยต์ (Use Case Point) (อานุภาพไตรรงค์, 2558)

| ประเภทของแอคเตอร์ (Actor type) | ค่าน้ำหนักปัจจัย (Weighting Factor) |
|---|---|
| ไม่ซับซ้อน (Simple) | 1 |
| ปานกลาง (Average) | 2 |
| ซับซ้อน (Complex) | 3 |

ตัวอย่างเช่น โครงการซอฟต์แวร์หนึ่งมีแอคเตอร์ประเภทปานกลาง 1 คน ประเภท ซับซ้อน 2 คน จะหาค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight หรือ UAW) ได้ดังนี้

$$UAW = (1 \times 2) + (2 \times 3) = 8$$

CHULALONGKORN UNIVERSITY

**การหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Weights หรือ UUCW)**

หลังจากที่เราหาค่า น้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight หรือ UAW) ได้แล้ว ในลำดับถัดไปผู้ประเมินจะหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Weights หรือ UUCW) โดยวิธีการหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่านั้น ผู้ประเมินจะแบ่งยูสเคสออกเป็น 3 ระดับ คือ ระดับไม่ซับซ้อน (Simple), ระดับปานกลาง (Average) และระดับซับซ้อน (Complex) และผู้ประเมินระดับของยูสเคสและหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ได้จากตารางที่ 29

ตารางที่ 29 15 ค่าน้ำหนักปัจจัย (Weighting Factor) ของยูสเคส (Use Case) ชนิดต่างๆ (อานุภาพ ไตรรงค์, 2558)

| Use Case Type | จำนวนรายการ (No of Transaction) | ค่าน้ำหนักปัจจัย (Weighting Factor) |
|---|---|---|
| ระดับไม่ซับซ้อน (Simple) | <=3 | 5 |
| ระดับกลาง (Average) | 4 to 7 | 10 |
| ระดับซับซ้อน (Complex) | >= 7 | 15 |

ตัวอย่างการหาค่ายูสเคสที่ยังไม่ได้ปรับค่า (UUCW) เช่น โครงการซอฟต์แวร์หนึ่งเมื่อนับจำนวนเส้นการกระทำของแอคเตอร์ในยูสเคสไดอะแกรม พบว่ามีจำนวน 5 รายการ จึงจัดว่าเป็นยูสเคสระดับปานกลาง และสามารถคำนวณหาค่า น้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่าของโครงการซอฟต์แวร์ตัวอย่าง ได้ดังนี้

น้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า (UUCW) = 5 x 2 = 10

**การหาค่ายูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Point)**

Unadjusted Use Case Point (UUCP) จะมีค่าเท่ากับ UUCW + UAW หรือเขียนได้ว่า

UUCP = UUCW + UAW

ผลลัพธ์ที่ได้จากการหาค่ายูสเคสพอยต์ที่ยังไม่ได้ปรับค่า จะยังไม่รวมผลกระทบจากปัจจัยที่เกี่ยวข้องทางด้านเทคนิคและทรัพยากร ดังนั้นจึงได้มีกระบวนการหายูสเคสพอยต์ที่มีการปรับค่าตามปัจจัยด้านสิ่งแวดล้อม (Environment Factor) และปัจจัยทางด้านเทคนิค (Technical Factor) เพื่อให้ได้ผลลัพธ์การประเมินที่คาดว่าจะแม่นยำมากขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การปรับค่าตามปัจจัยต่างๆ จะเกิดจากการพิจารณาถึงความสามารถในการพัฒนาของนักพัฒนาระบบ รวมไปถึงทรัพยากรที่ใช้ในการพัฒนาระบบ ผู้ประเมินอาจจะเป็นผู้กำหนดหรืออาจจะเกิดจากการกำหนดโดยทีมพัฒนา
ทั้งนี้ขึ้นอยู่กับว่าปัจจัยเหล่านั้นมีผลกระทบต่อค่าความพยายามในการพัฒนาซอฟต์แวร์มากน้อยเพียงใด
โดย

- 0 หมายถึงไม่มีผลกระทบ
- 3 หมายถึงผลกระทบในระดับกลาง
- 5 หมายถึงมีผลกระทบสูง

**การปรับค่าปัจจัยทางด้านเทคนิคและด้านสิ่งแวดล้อม**

การปรับค่ายูสเคสพอยต์เกิดจากสองปัจจัยคือปัจจัยทางด้านเทคนิคและปัจจัยทางด้านสิ่งแวดล้อม ทั้งนี้ผู้ประเมินจะต้องหาค่าความซับซ้อนทางด้านเทคนิค (Technical Complexity Factor หรือ TCF) และค่าความซับซ้อนทางด้านสิ่งแวดล้อม (Environmental Complexity Factor หรือ ECF)

**การหาค่าความซับซ่อนทางเทคนิค (Technical Complexity Factor หรือ TCF)**

ในขั้นแรกของการหาค่าความซับซ้อนทางเทคนิค (Technical Complexity Factor หรือ TCF) ผู้ประเมินจะต้องหาค่าปัจจัยทางเทคนิค (Technical Factor) เสียก่อน ซึ่งมีวิธีการหาดังนี้

**การหาค่าปัจจัยทางเทคนิค (Technical Factor หรือ TFactor)**

ผู้ประเมินจะต้องประเมินค่าปัจจัยทางเทคนิคทั้ง 13 ประการ (T1-T13) แสดงในตารางที่ 30 ตารางที่ 30 ค่าน้ำหนักของปัจจัยทางเทคนิค (Technical Factor) ทั้ง 13 ประการ ในการปรับค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยต์ (อานุภาพไตรรงค์, 2558)

| ปัจจัย (Factor) | คำอธิบาย (Description) | คำอธิบายเทคนิค | น้ำหนัก (Weight) |
|---|---|---|---|
| T1 | Distributed System | ความสามารถในการประมวลผลแบบกระจาย | 2 |
| T2 | Response Adjective | ความสามารถในการตอบสนองความต้องการ | 2 |
| T3 | End-user Efficiency | ความสามารถในประสิทธิภาพผู้ใช้งาน | 1 |
| T4 | Complex Processing | ความซับซ้อนในการประมวลผล | 1 |
| T5 | Reusable code | ความสามารถนำโค้ดกลับมาใช้ใหม่ | 1 |
| T6 | Easy to install | ความสะดวกในการติดตั้ง | 0.5 |
| T7 | Easy to use | ความสะดวกในการใช้งาน | 0.5 |
| T8 | Portable | ความยืดหยุ่นในการใช้งานหลายแพลตฟอร์ม | 2 |
| T9 | Easy to Change | ความสะดวกในการเปลี่ยนแปลง | 1 |
| T10 | Concurrent | ความสามารถในการรองรับการเข้าใช้งานพร้อมกัน | 1 |
| T11 | Security Features | ความสามารถทางด้านความปลอดภัย | 1 |
| T12 | Access for third Parties | ความสามารถในการเข้าถึงโดยบุคคลที่สาม | 1 |
| T13 | Special Training Required | ความต้องการในการอบรมผู้ใช้งาน | 1 |

ผลรวมของค่าปัจจัยทางเทคนิค (Technical Factor หรือ TFactor) จะมีค่าเท่ากับ ผลรวมของค่าที่ประเมินได้จาก Technical Factor แต่ละค่าคูณกับน้ำหนักของ Technical Factor เหล่านั้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$\text{TFactor} = \sum Ti(W)$$

เมื่อ Ti คือ ค่าที่ประเมินจากผู้ประเมินในแต่ละหัวข้อของปัจจัยทางด้านเทคนิค มีค่าตั้งแต่ 0-5 โดย 0 หมายถึง ไม่มีผลกระทบเลย และ 5 หมายถึงมีผลกระทบมากที่สุด

และ W คือ น้ำหนักของปัจจัยทางด้านเทคนิคในแต่ละข้อ

ค่าความซับซ้อนทางเทคนิค (Technical Complexity Factor หรือ TCF) จะเป็นค่าความซับซ้อนในเชิงเทคนิคที่ใช้ในการปรับค่ายูสเคสพอยต์โดยตรงคำนวณได้จาก

$$\text{TCF} = 0.6 + (0.01 \times \text{TFactor})$$

**การหาค่าความซับซ้อนทางสิ่งแวดล้อม (Environment Complexity Factor หรือ ECF)**

ในขั้นแรกของการหาค่าความซับซ้อนทางสิ่งแวดล้อม (Environment Complexity Factor หรือ ECF) ผู้ประเมินจะต้องหาค่าปัจจัยทางสิ่งแวดล้อม (Environment Factor) เสียก่อน ซึ่งมีวิธีการหาดังนี้

**การหาค่าปัจจัยทางสิ่งแวดล้อม (Environment Factor หรือ EFactor)**

ผู้ประเมินจะต้องประเมินค่าปัจจัยทางสิ่งแวดล้อมทั้ง 8 ประการ (E1-E8) โดยแต่ละปัจจัยจะสามารถกำหนดค่าได้ตั้งแต่ 0-5 โดย 0 หมายถึงไม่มีผลกระทบเลย และ 5 หมายถึงมีผลกระทบมากที่สุด โดยปัจจัยทั้ง 8 ปัจจัยแสดงในตารางที่ 31

ตารางที่ 31 ค่าน้ำหนักของปัจจัยทางด้านทรัพยากร (Environment Factor) ทั้ง 8 ประการ ในการปรับค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยต์ (อานุภาพไตรรงค์, 2558)

| ปัจจัย (Factor) | คำอธิบาย (Description) | คำอธิบายเทคนิค | น้ำหนัก (Weight) |
|---|---|---|---|
| E1 | Familiar with RUP | ความคุ้นเคยกับตัวแบบการพัฒนาระบบ | 1.5 |
| E2 | Application Experience | ประสบการณ์ในการพัฒนาแอปพลิเคชัน | 0.5 |
| E3 | Object-Oriented Experience | ประสบการณ์ด้านด้านเชิงวัตถุ | 1 |
| E4 | Lead Analyst Capability | ความสามารถในการวิเคราะห์ของผู้นำทีม | 0.5 |
| E5 | Motivation | แรงจูงใจในการทำงาน | 1 |
| E6 | Stable Requirements | ความต้องการที่คงที่ (ไม่เปลี่ยนแปลง) | 2 |
| E7 | Part-time workers | พนักงานนอกเวลา | -1 |
| E8 | Difficult Programming Language | ความยากของภาษาโปรแกรม | -1 |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ผลรวมของค่าปัจจัยทางสิ่งแวดล้อม (EFactor) จะคำนวณได้จาก ผลรวมของค่าที่ประเมินได้จาก Environmental Factor คูณกับน้ำหนักของ Environmental Factor เหล่านั้น

$$\text{EFactor} = \sum Ei(W)$$

เมื่อ Ei คือ ค่าที่ประเมินจากผู้ประเมินในแต่ละหัวข้อของปัจจัยทางด้านสิ่งแวดล้อม มีค่าตั้งแต่ 0-5 และ W คือ น้ำหนักของปัจจัยทางด้านเทคนิคในแต่ละข้อ

ค่าความซับซ้อนทางสิ่งแวดล้อม **(Environment Complexity Factor หรือ ECF)** จะเป็นค่าความซับซ้อนในเชิงสิ่งแวดล้อมที่ใช้ในการปรับค่ายูสเคสพอยต์โดยตรงคำนวณได้จาก

$$\text{ECF} = 1.4+(-0.03 \times \text{EFactor})$$

**การหาค่ายูสเคสพอยต์ที่ปรับค่าแล้ว (Adjusted Use Case Point หรือ UCP)**

การหาค่ายูสเคสพอยต์ที่ปรับค่าแล้วจะหาได้จากค่า ยูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (UUCP), ค่าความซับซ้อนทางเทคนิค (TCF) และ ค่าความซับซ้อนทางสิ่งแวดล้อม (ECF)

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{ECF}$$

**การคำนวณหาค่าความพยายามในการพัฒนาซอฟต์แวร์**

โดยทั่วไปจะใช้อัตราความสามารถในการผลิต (Productivity Rate) ของ Karner คือ 1 Use Case Point จะมีค่าเท่ากับ 20 ชั่วโมง ดังนั้น

ค่าความพยายามในการพัฒนาซอฟต์แวร์ = UCP x 20 (หน่วยเป็น Man-Month)

**2.3 งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความแม่นยำของค่าความพยามในการพัฒนาซอฟต์แวร์**

จากการทบทวนวรรณกรรมเกี่ยวกับการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ และวิธีตัวแบบขั้นตอนวิธี พบว่ามีงานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบค่าความพยายามในการพัฒนาซอฟต์แวร์ดังนี้

1) A Review of studies on expert estimation of software development effort (M. Jorgensen, 2002)
2) Are Software cost-estimation models accurate? (Kuster, 1990)
3) Software-Effort Estimation: An Exploratory Study of Expert Performance (Vicinanza, 1991)
4) Examining the Feasibility of a Case-Based Reasoning Model for Software Effort Estimation (Mukhopadhyay, 1992)
5) Using Function Points to Find Cost Analogies (Atkinson, 1994)

6) Performance of effort estimating techniques in current development environments (Pengelly, 1995)
7) Forecasting of software development work effort: Evidence on expert judgment and formal models (M. Jorgensen, 2007)
8) การประมาณการต้นทุนค่าใช้จ่ายในการพัฒนาเนื้อหาบทเรียนออนไลน์สำหรับเด็กผิดปกติ โดยวิธีแบบจำลอง COCOMO – II ร่วมกับวิธีการของ Web Model (เจิม บุญธนะพงศ์, 2558)

งานวิจัยของ (M. Jorgensen, 2002) กล่าวถึงการศึกษางานการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญรวมไปถึงการเปรียบเทียบประสิทธิภาพของวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัดสินของผู้เชี่ยวชาญ เทียบกับวิธีตัวแบบขั้นตอนวิธี งานวิจัยนี้ได้ทำการศึกษางานวิจัยอื่นๆ ทั้งหมด 15 งานวิจัย ซึ่งมีทั้งงานวิจัยที่เป็นการทดลอง งานวิจัยที่เป็นการสังเกตการณ์ และงานวิจัยที่เป็นการใช้แบบสอบถาม ผู้วิจัยจะเลือกศึกษาเฉพาะงานวิจัยที่เกี่ยวข้องกับการทดลองเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี ซึ่งมีทั้งหมด 5 งานวิจัย มีรายละเอียดดังนี้

งานวิจัยของ (Kuster, 1990) ได้มีการทดลองเปรียบเทียบความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของวิธีการประเมิน 4 วิธี ดังนี้

1) วิธีการตัดสินของผู้เชี่ยวชาญ
2) วิธีตัวแบบบีวายแอล (BYL)
3) วิธีตัวแบบเอสติแมก (Estimacs)
4) วิธีผสมระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี

1. วัตถุประสงค์การทดลอง
    1) เพื่อศึกษาหาความแม่นยำของแบบจำลองในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์
    2) เพื่อหาว่าแบบจำลองที่ใช้ในการทดลองหรือแบบจำลองใกล้เคียงเป็นที่ยอมรับในกลุ่ม Project Leader หรือไม่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3) เพื่อศึกษาว่าจำนวนบรรทัดที่ใช้ในการพัฒนาซอฟต์แวร์ (lines of code) สามารถเป็นดัชนีชี้วัดที่สำคัญในการกำหนดขนาดของโครงการซอฟต์แวร์สำหรับการพัฒนาหรือไม่

2. วิธีการทดลอง

นักวิจัยได้กำหนดเกณฑ์ในการเลือกแบบจำลองในการทดลองเอง ซึ่งแบบจำลองที่ผ่านเกณฑ์การเลือกของงานวิจัยนี้คือ บีวายแอล และ เอสติแมก

นักวิจัยจะให้ข้อมูลโครงการซอฟต์แวร์แก่ผู้ประเมินทั้ง 14 คน (งานวิจัยนี้ไม่ได้ระบุลักษณะของผู้ประเมิน) ข้อมูลโครงการซอฟต์แวร์เป็นโครงการซอฟต์แวร์ที่ชื่อ Bonus System ขององค์กรที่ดำเนินธุรกิจเกี่ยวกับการขาย (Sales Organization) มีรายละเอียดที่ประกอบไปด้วยสภาพแวดล้อมขององค์กร (Organization Environment), รายละเอียดฟังก์ชันการทำงาน (Functional Specification) และเป้าหมายของโครงการ (Goal) ข้อมูลเหล่านี้เป็นข้อมูลที่ทางทีมพัฒนาได้รับก่อนพัฒนาโครงการซอฟต์แวร์จริง ภายหลังการประเมิน นักวิจัยและผู้ประเมินจะมาอภิปรายผลลัพธ์เพื่อสรุปการประเมิน

3. สรุปผลการทดลอง

การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี มีความแม่นยำมากกว่า การประมาณด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

งานวิจัยของ (Vicinanza, 1991) กล่าวถึงการทดสอบความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ เปรียบเทียบกับวิธีตัวแบบขั้นตอนวิธี 2 แบบ คือ โคโคโม่ และ ฟังก์ชันพอยต์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

นักวิจัยงานวรรณกรรมชิ้นนี้ได้เชิญผู้ประเมินในตำแหน่งผู้จัดการซอฟต์แวร์ (Software Manager) ผู้ที่มีประสบการณ์ 5 คน มีประสบการณ์ในการทำงานขั้นต่ำ 10 ปี (งานวิจัยนี้ไม่ได้ระบุจำนวนโครงการของผู้ประเมินที่เคยทำ ภาษาที่ผู้ประเมินถนัด) มาเข้าร่วมการทดสอบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ และบันทึกวิธีการประเมินของผู้ประเมินว่าผู้ประเมิน ใช้วิธีใดในการประเมิน ซึ่งโครงการซอฟต์แวร์ที่นำมาให้ประเมิน มีขนาดตั้งแต่ 39,000 ถึง 450,000 lines of code มีจำนวนโครงการทั้งสิ้น 10 โครงการ และมีค่าความพยายามในการพัฒนาซอฟต์แวร์ตั้งแต่ 23 - 1,107 คน-เดือน

1. วัตถุประสงค์ของการทดลอง

1) เพื่อหาว่าผู้ประเมินสามารถประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ได้อย่างแม่นยำหรือไม่
2) เพื่อศึกษาวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของผู้ประเมิน

2. วิธีการทดลอง

งานวิจัยนี้ ทำการทดลองโดยให้ผู้ดำเนินโครงการซอฟต์แวร์ผู้ที่มีประสบการณ์ 5 คนมาเข้าร่วมการทดสอบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ และบันทึกวิธีการประเมินของผู้ดำเนินโครงการซอฟต์แวร์ ว่าใช้วิธีใดในการประเมิน ซึ่งโครงการซอฟต์แวร์ที่นำมาให้ประเมิน มีขนาดตั้งแต่ 39,000 ถึง 450,000 บรรทัด และมีค่าความพยายามในการพัฒนาซอฟต์แวร์ตั้งแต่ 23 - 1,107 คน-เดือน

ผู้ประเมินทุกคนมีประสบการณ์ในการทำงานทางด้านการพัฒนาซอฟต์แวร์ ขั้นต่ำ 10 ปี และมีประสบการณ์ทางด้านการบริหารจัดการโครงการซอฟต์แวร์หรือความรับผิดชอบในการประเมินต้นทุน รวมไปถึงได้รับการยอมรับในทีมว่ามีความสามารถในการประเมินที่แม่นยำ

โครงการซอฟต์แวร์ที่นำมาทำการทดลองเป็นโครงการที่มาจาก (Kemerer, 1987) มีขนาดโครงการตั้งแต่ 39,000 – 450,000 บรรทัด (100 ถึง 2,300 Function Points) และมีค่าความพยายามในการพัฒนา 23 ถึง 1,107 คน-เดือน

ในการทดลองการประเมิน ผู้ประเมินจะอยู่ในห้องเงียบส่วนตัว เพื่อไม่ให้ปัจจัยจากสภาพแวดล้อมภายนอกเข้ามารบกวนการประเมิน ผู้ประเมินจะได้รับชิ้นงานสำหรับการประเมินในลำดับที่เหมือนกันทุกคน ซึ่งชิ้นงานเหล่านั้นจะประกอบด้วยปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ หน้าที่ของผู้ประเมินคือจะต้องเรียงลำดับความสำคัญของปัจจัยเหล่านั้น และประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ซึ่งในระหว่างที่ผู้ประเมินทำการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์นั้น ผู้ประเมินจะต้องอธิบายแนวความคิดในการประเมินโดยการพูดผ่านเครื่องอัดเสียงในห้อง (Verbal Protocol)

3. สรุปผลการทดลอง

จากการทดลองสรุปได้ว่า การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่า วิธีตัวแบบขั้นตอนวิธี และจากการทดลองพบว่า 4 ใน 5 ของผู้ประเมิน ใช้การตัดสินของผู้เชี่ยวชาญ โดยใช้หลักการของ แผนการขั้นตอนวิธี (Algorithmic Strategies) ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ส่วนผู้ประเมิน อีกท่านใช้หลักการของแผนการอุปมา (Analogy Strategies) ซึ่ง ผู้ประเมินที่ใช้แผนการอุปมา (Analogy Strategies) นั้นสามารถประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ได้แม่นยำมากที่สุด (แต่ไม่

สามารถสรุปได้ว่าการใช้วิธีการตัดสินของผู้เชี่ยวชาญ ด้วยแผนการอุปมา นั้นมีความแม่นยำมากกว่าแผนการขั้นตอนวิธี เนื่องจากผู้ประเมินที่ใช้วิธีแผนการขั้นตอนวิธี มีเพียงคนเดียว)

งานวิจัยของ (Mukhopadhyay, 1992) กล่าวถึงการสร้างแบบจำลองของการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ ชื่อว่า เอสเตอร์ (Estor) เป็นขั้นตอนวิธีการให้เหตุผลตามกรณี (Case-Based Reasoning Model หรือ CBR) มาจากการระดมความคิดของผู้เชี่ยวชาญ

เพื่อทดสอบความแม่นยำและความแน่นอนของผลลัพธ์ที่ได้จากการประเมินด้วย เอสเตอร์ (Estor) นักวิจัยโครงงานชิ้นนี้จึงนำเอสเตอร์มาทำการทดลองเปรียบเทียบกับวิธีการตัดสินของผู้เชี่ยวชาญ และ วิธีตัวแบบขั้นตอนวิธี อีก 2 ชนิดคือ โคโคโม่ และ ฟังก์ชันพอยต์ โดยใช้โครงการซอฟต์แวร์ที่สำเร็จแล้ว 5 โครงการ

1. วัตถุประสงค์การทดลอง

เพื่อเสนอตัวแบบในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ใหม่ ที่ชื่อว่า เอสเตอร์ และเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เทียบกับวิธีการตัดสินของผู้เชี่ยวชาญ, โคโคโม่ และ ฟังก์ชันพอยต์

2. วิธีการทดลอง

เอสเตอร์ (Estor) สร้างมาจาก ตัวแทนผู้เชี่ยวชาญที่มีความแม่นยำและความแน่นอนของผลลัพธ์ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์สูง เรียกว่า “ผู้เชี่ยวชาญอ้างอิง (referent expert)” โดยบุคคลนี้มีตำแหน่งหัวหน้าทีมพัฒนาซอฟต์แวร์ (Software Development Manager) มีประสบการณ์ 10 ปีจากการทำงานในสายงานทางด้านการจัดการซอฟต์แวร์ (Software Management) และมีประสบการณ์ 9 ปี ในสายงานเกี่ยวกับแอปพลิเคชันเกี่ยวกับการประมวลผลข้อมูล (Data Processing Application)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

เครื่องบันทึกเสียงจะบันทึกการประเมินค่าความพยายามของ ผู้เชี่ยวชาญอ้างอิง ที่ทำโจทย์ประเมินโครงการซอฟต์แวร์ 10 ข้อ โดยใช้แบบจำลองการให้เหตุผลตามกรณี ที่ได้มาจากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ในโครงการซอฟต์แวร์ก่อนหน้านี้ โดยมีกระบวนการทำงาน 5 ขั้นตอนดังนี้

1. Construct : แสดงรายละเอียดคุณสมบัติต่างๆของโครงการซอฟต์แวร์ในแต่ละโครงการ
2. Retrieve : เลือกโครงการซอฟต์แวร์ที่มีลักษณะใกล้เคียงกันกับโครงการซอฟต์แวร์ที่ต้องการประเมิน
3. Transfer : นำค่าคุณสมบัติจากโครงการซอฟต์แวร์ที่เลือกในขั้นตอน Retrieve มาใช้กับโครงการซอฟต์แวร์ที่ต้องการประเมิน

4. Map : เปรียบเทียบคุณสมบัติของโครงการซอฟต์แวร์ที่ต้องการประเมินกับโครงการซอฟต์แวร์ที่ได้รับจากฐานความรู้ (knowledge-base)
5. Adjust : เอสเตอร์จะสร้างกฎสำหรับการปรับค่าที่ไม่สอดคล้องกันระหว่างโครงการซอฟต์แวร์ที่ต้องการประเมินกับโครงการซอฟต์แวร์ที่นำมาจากฐานความรู้ (knowledge-base)

โดยมีฐานความรู้ 3 กลุ่ม คือ ฐานความรู้ สำหรับการนำเสนอกรณีศึกษาของโครงการซอฟต์แวร์ ฐานความรู้สำหรับเลือกโครงการซอฟต์แวร์จากโครงการซอฟต์แวร์ที่ผ่านมาในอดีต และ ฐานความรู้สำหรับปรับค่าการประเมินจากลักษณะของซอฟต์แวร์กรณีศึกษากับซอฟต์แวร์ที่ผ่านมาในอดีต

3. สรุปผลการทดลอง

ผลการทดลองสามารถเรียงลำดับความแม่นยำได้ดังนี้วิธีการตัดสินของผู้เชี่ยวชาญ เอสเตอร์โคโคโม่ และ ฟังก์ชันพอยต์ ตามลำดับ

งานวิจัยของ (Atkinson, 1994) กล่าวถึงการเปรียบเทียบความแม่นยำของการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินของผู้เชี่ยวชาญ ฟังก์ชันพอยต์ และ วิธีอุปมา โดยทำการทดลองกับโครงการซอฟต์แวร์จริง 21 โครงการ

1. วัตถุประสงค์การทดลอง

เพื่อเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีอุปมา, วิธีการตัดสินของผู้เชี่ยวชาญ และวิธีตัวแบบขั้นตอนวิธี

2. วิธีการทดลอง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

งานวิจัยนี้ทำการทดลองโดยรวบรวมโครงการซอฟต์แวร์จำนวน 21 โครงการ เพื่อมาเป็นฐานข้อมูลในสำหรับการวิเคราะห์ค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีอุปมา ซึ่งในการค้นหาโครงการซอฟต์แวร์ที่คล้ายคลึงกันนั้นจะใช้ วิธีการวัดระยะทางแบบยุคลิด (Euclidean Distance)

3. สรุปผลการทดลอง

การเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์สามารถเรียงลำดับความแม่นยำได้ดังนี้ วิธีการอุปมา, วิธีการตัดสินของผู้เชี่ยวชาญ และ วิธีตัวแบบขั้นตอนวิธี ตามลำดับ

งานวิจัยของ (Pengelly, 1995) ได้กล่าวถึงการทดลองการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของโครงการจัดการคุณภาพการให้บริการทางด้านเทคโนโลยีสารสนเทศ (the Quality of Service management Information system) ซึ่งเป็นการทดลองเปรียบเทียบการประเมินระหว่างวิธี

1) ประสิทธิภาพจำนวนบรรทัดในการเขียนโค้ด (Effective lines of code หรือ ELOC)
2) วิธีจากล่างขึ้นบน (Bottom-up estimation)
3) ตัวแบบโดตี้ (Doty Model)
4) โคโคโม่ (COCOMO)
5) ฟังก์ชั้นพอยต์ (Function Point Analysis หรือ FPA)
6) ตัวแบบพัทนัมสลิม (Putnum SLIM Model)

1. วัตถุประสงค์การทดลอง

เพื่อเปรียบเทียบความแม่นยำในการระเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธี ประสิทธิภาพจำนวนบรรทัดในการเขียนโค้ด, วิธีจากล่างขึ้นบน, ตัวแบบโดตี้, โคโคโม่, ฟังก์ชันพอยต์ และ ตัวแบบพัทนัมสลิม

2. วิธีการทดลอง

การทดลองจะดำเนินการตามวิธีการประเมินของแต่ละวิธีเพื่อให้ได้ค่าประมาณจากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ หลังจากนั้นจะวิเคราะห์ผลลัพธ์ที่ได้จากการประเมินของแต่ละวิธีและอธิปรายขอแนะนำการปรับปรุงในการทดลองแต่ละวิธี

3. สรุปผลการทดลอง

การประเมินด้วยวิธีจากล่างขึ้นบน มีความแม่นยำมากกว่าการประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี ตัวแบบอื่นๆ

นอกจากนี้ ผู้วิจัยได้ทำการศึกษางานวิจัยของ (M. Jorgensen, 2007) เพิ่มเติมพบว่างานวิจัยของ Jorgensen กล่าวถึงการทบทวนวรรณกรรมของงานที่เกี่ยวข้องเกี่ยวกับการใช้ วิธีการตัดสินของผู้เชี่ยวชาญ, ตัวแบบขั้นตอนวิธี และ การรวมวิธีทั้งสอง ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ จากการศึกษาวรรณกรรมที่เกี่ยวข้องทั้ง 16 วรรณกรรม พบวรรณกรรมที่อ้างถึงวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่า ตัวแบบขั้นตอนวิธี จำนวน 10 วรรณกรรม (คิดเป็น 62.5%) และ ความแม่นยำเฉลี่ยของการใช้วิธีการประเมินค่าความพยายามแบบผสมมีความ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

แม่นยำใกล้เคียงกับผลลัพธ์จากการประเมินที่ดีที่สุดของแต่ละกรณี ส่วนงานวิจัยที่ทำการทดลองในสภาวะแวดล้อมในประเทศไทย

ผู้วิจัยได้ทำการศึกษาเพิ่มเติมในงานวิจัยของ (เจิมบุญธนะพงศ์, 2558) กล่าวถึงการทดลองการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอตฟ์แวร์ด้วยวิธี COCOMO-II ร่วมกับ Web Model และ วิธีการตัดสินใจของผู้เชี่ยวชาญ ในโครงการซอฟต์แวร์เนื้อหาบทเรียนออนไลน์

1.วัตถุประสงค์การทดลอง

เพื่อเปรียบเทียบความแม่นยำในการพัฒนาซอฟต์แวร์ระหว่างวิธี COCOMO-II ร่วมกับ Web Model และ วิธีการตัดสินของผู้เชี่ยวชาญ

2.วิธีการทดลอง

งานวิจัยชิ้นนี้ออกแบบการทดลองโดยการสัมภาษณ์ นักพัฒนาซอฟต์แวร์เพื่อเก็บข้อมูล เช่น ข้อมูลเนื้อหารายวิชา ระยะเวลาที่ใช้ในการวิเคราะห์เนื้อหาเพื่อสร้างเอกสาร เป็นต้น ภายหลังจากการเก็บข้อมูล งานชิ้นนี้ไม่ได้ระบุแน่ชัดถึงวิธีการประเมินของวิธีตัวแบบขั้นตอนวิธี และวิธีการตัดสินของผู้เชี่ยวชาญ

3.สรุปผลการทดลอง

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี โคโคโม่ 2 ร่วมกับ Web Model มีค่าความแม่นยำมากกว่าวิธีการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

**2.4 สรุปการทบทวนวรรณกรรม**

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากการทบทวนวรรณกรรมพบว่างานวิจัยที่เกี่ยวข้องของการเปรียบเทียบค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญ และวิธีตัวแบบขั้นตอนวิธี ทั้งหมด 8 งานวิจัยที่เกี่ยวข้อง พบว่ามี 4 งานวิจัยที่วิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี และมี 4 งานวิจัยที่วิธีตัวแบบขั้นตอนวิธี มีความแม่นยำมากกว่าวิธีการตัดสินของผู้เชี่ยวชาญ โดยการทดลองทั้ง 8 งานวิจัยดังกล่าวได้สรุปเป็นตารางที่ 32 ดังนี้

ตารางที่ 32 ผลสรุปวรรณกรรมเกี่ยวกับการทดลองการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์

| งานวิจัย | จำนวนโครงการ | จำนวน/ลักษณะ ผู้เชี่ยวชาญ ที่เข้าร่วมการทดลอง | คำถามวิจัย/สิ่งที่ต้องการศึกษา | การวัดผล | ภาษาที่ใช้ | Algorithmic Model | ผลลัพธ์ |
|---|---|---|---|---|---|---|---|
| Kusters (1990) | 1 โครงการ คือ 'Bonus System' | บุคคลทำงานในตำแหน่ง Project Leader จำนวน 14 คน | ทำการทดลองเพื่อหาแบบจำลองในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ | • ค่าเฉลี่ยของค่าความพยายาม | - | บีวายแอล, เอสติแมกซ์ | การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี มีความแม่นยำมากกว่า การประมาณด้วยวิธีการตัดสินของผู้เชี่ยวชาญ |
| Vicinanza (1991) | 10 โครงการ แรกจาก Kemerer, 1987 | บุคคลทำงานในตำแหน่ง Project Manager/ Cost Estimator 5 คน มีประสบการณ์ในการทำงานขั้นต่ำ 10 ปี | ทำการทดลองทดสอบ Software Manager ผู้ที่มีประสบการณ์ว่าสามารถประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ได้อย่างแม่นยำจริงหรือไม่และบันทึกวิธีการประมาณค่าความพยายามของ Software Manager | • MRE<br>• Correlation | COBAL | โคโคโม่, ฟังก์ชันพอยต์ | การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| Mukhopadhyay (1992) | 15 โครงการ จาก (Kemerer, 1987) | Software Development Manager 1 คน ทำงานในบริษัทซอฟต์แวร์ขนาดใหญ่ โดยมีประสบการณ์ทำงาน 10 ปี และมีประสบการณ์ 9 ปีในการทำงานเกี่ยวกับ Data Processing Application | เปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย เอสเตอร์, วิธีการตัดสินของผู้เชี่ยวชาญ , โคโคโม่ และ ฟังก์ชันพอยต์ | • MRE<br>• Correlation<br>• Significance | - | โคโคโม่, ฟังก์ชันพอยต์ | การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| Attkinson and Shepperd (1994) | 1 โครงการ | - | เปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีอุปมา, ฟังก์ชันพอยต์ และ วิธีการตัดสินของผู้เชี่ยวชาญ | • MMRE<br>• Pred(.25) | 4GL, COBAL,C | ฟังก์ชันพอยต์ | การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| Pengelly (1995) | 1 โครงการ | - | เพื่อเปรียบเทียบความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีประสิทธิภาพขนาดบรรทัดในการเขียนโค้, วิธีการตัดสินของผู้เชี่ยวชาญ, ตัวแบบโดตี้, โคโคโม่, ฟังก์ชันพอยต์, ตัวแบบพัทนัมสลิม | • MRE | Omnis 7 | ประสิทธิภาพขนาดบรรทัดในการเขียนโค้ด, วิธีตัวแบบโดตี, โคโคโม, ฟังก์ชันพอยต์, ตัวแบบพัทนัมสลิม | การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี |
| ขวัญชนก ธรณธรรมกุล และปิติพัฒน์ เจิมบุญธนะพงศ์ (2558) | 1 โครงการ | - | เพื่อเปรียบเทียบความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธี โคโคโม่2 ร่วมกับ Web Model กับ วิธีการตัดสินของผู้เชี่ยวชาญ | • MAPE | - | โคโคโม่ 2 ร่วมกับ Web Model | การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธี มีความแม่นยำมากกว่า การประมาณด้วยวิธีการตัดสินของผู้เชี่ยวชาญ |

แม้ว่าจะมีงานวิจัยที่ทำการทดลองเกี่ยวข้องกับการเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี กับ การตัดสินของผู้เชี่ยวชาญ แล้วแต่งานวิจัยเหล่านั้นได้มีการทดลองเมื่อนานมาแล้ว ซึ่งโครงการซอฟต์แวร์ในการทดลองดังกล่าวได้ใช้ภาษาโปรแกรมมิ่งที่ไม่ได้เป็นที่นิยมใช้ในปัจจุบัน อีกทั้งงานวิจัยของ Attkinson and Shepperd, Kusters et al. และ Pengelly ได้ใช้โครงการซอฟต์แวร์ในการทดลองเพียง 1 โครงการ ทำให้ข้อมูลผลลัพธ์ดังกล่าวไม่น่าเชื่อถือเท่าที่ควร

จากการศึกษางานวิจัยที่กล่าวมาข้างต้น จึงไม่สามารถสรุปได้ว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีตัวแบบขั้นตอนวิธี และวิธีการตัดสินของผู้เชี่ยวชาญ วิธีใดแม่นยำกว่ากัน ภายใต้สภาวะแวดล้อมการพัฒนาซอฟต์แวร์ในประเทศไทย การทดลองเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จึงมีความน่าสนใจและได้ตั้งคำถามงานวิจัยดังนี้

**การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าหรือน้อยกว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี ภายใต้สภาวะแวดล้อมของไทย**

การศึกษางานวิจัยของ (Macdonell, 2003) ได้ทำการทดลองเปรียบเทียบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี, วิธีการตัดสินของผู้เชี่ยวชาญ และวิธีผสมระหว่างวิธีทั้งสอง ซึ่งพบว่าวิธีแบบผสมมีความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์มากที่สุด แต่ไม่มีการทดลองที่ทำการทดสอบถึงการติดสินใจของผู้ประเมินในวิธีการตัดสินของผู้เชี่ยวชาญ เมื่อทราบผลการประเมินของวิธีตัวแบบขั้นตอนวิธี ด้วยเหตุนี้จึงได้มีการตั้งคำถามการทดลองเพื่อหาคำตอบดังนี้

**การนำค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ จะช่วยให้การประเมินแม่นยำขึ้นหรือไม่**

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 3 ระเบียบวิธีวิจัย

บทนี้จะนำเสนอระเบียบวิธีวิจัยซึ่งประกอบไปด้วย

1. คำถามวิจัย
2. การตั้งสมมติฐาน
3. การออกแบบการทดลองเพื่อทดสอบสมมติฐาน
    - ภาพรวมขั้นตอนการดำเนินการทดลอง
    - การเลือกผู้เชี่ยวชาญและโครงการซอฟต์แวร์
    - การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี
    - การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญและบันทึกผล
    - การนำค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีมาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ
    - การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยผู้เชี่ยวชาญคนเดิมใหม่อีกครั้ง
4. การวิเคราะห์ผลการทดลอง
    - การหาค่าความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้รับจากการคำนวณด้วยวิธีตัวแบบขั้นตอนวิธีและค่าประเมินจากผู้เชี่ยวชาญ
    - การทดสอบสมมติฐาน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

## 3.1 คำถามวิจัย

จากการทบทวนวรรณกรรม ทำให้เกิดคำถามวิจัยในสภาวะแวดล้อมการพัฒนาซอฟต์แวร์ในประเทศไทย ดังนี้

1) การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่พัฒนาโดยนักพัฒนาคนไทยด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำแตกต่างกันหรือไม่กับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี ภายใต้สภาวะแวดล้อมของไทย

2) การนำค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ จะช่วยให้การประเมินแม่นยำขึ้นหรือไม่

**3.2 การตั้งสมมติฐาน**

จากคำถามวิจัยในข้อ 3.1 ผู้วิจัยได้ตั้งสมมติฐานดังนี้

1) การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ภายในสภาวะแวดล้อมที่ผู้เชี่ยวชาญเป็นคนไทยด้วยวิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าหรือน้อยกว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี

จากการทบทวนวรรณกรรมจาก 6 งานวิจัย พบว่า มี 4 งานวิจัย คือ Vicinanza et al. (1991), Mukhopadhyay et al. (1992), Attkinson and Shpperd (1994) และ Pengelly (1995) ที่วิธีการตัดสินของผู้เชี่ยวชาญ มีความแม่นยำมากกว่าตัวแบบขั้นตอนวิธี ส่วนอีก 2 งานวิจัย คือ Kusters et al. (1990) และ Myrtveit and Stensrud (1999) พบว่าวิธีตัวแบบขั้นตอนวิธีมีความแม่นยำมากกว่าวิธีการตัดสินของผู้เชี่ยวชาญ ผู้วิจัยจึงมีความคิดเห็นว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ อาจจะมีความแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธี

ผู้วิจัยได้ตั้งสมมติฐานการทดลองไว้ดังนี้

**<u>สมมติฐานที่ 1</u>**

$H_0$ : ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี **มากกว่าหรือเท่ากับ** ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

$H_1$ : ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี **น้อยกว่า** ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

2) การนำค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ จะช่วยให้การประเมินแม่นยำขึ้นหรือไม่

งานวิจัยของ (Grimstad, 2007) พบว่าผู้ประเมินที่ถนัดในด้านการคิดวิเคราะห์ จะมีความเชื่อมั่นในค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี แต่หากผู้ประเมินเชื่อในความคิดของตนเอง จะเชื่อมั่นในค่าประเมินที่ตนประเมินได้ โดยผู้ประเมินส่วนใหญ่จะเชื่อมั่นในค่าประเมินของตนเองมากกว่าและจะไม่เปลี่ยนใจ

ผู้วิจัยจึงได้กำหนดสมมติฐานที่ 2 ดังนี้

สมมติฐานที่ **2.1**

$H_0$ **:** ค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี **ทำให้ผู้เชี่ยวชาญเปลี่ยนแปลงค่าประเมินมากกว่าครึ่งหนึ่งของจำนวนการทดลองทั้งหมด** (**จำนวนการทดลองทั้งหมด 50 การทดลอง**) จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินใจของผู้เชี่ยวชาญ

$H_1$ **:** ค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี **ทำให้ผู้เชี่ยวชาญเปลี่ยนแปลงค่าประเมินไม่เกินครึ่งหนึ่งของจำนวนการทดลองทั้งหมด** (**จำนวนการทดลองทั้งหมด 50 การทดลอง**) จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินของผู้เชี่ยวชาญ

จากการศึกษางานวิจัยของ (**Grimstad,** 2007) พบว่า การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญนั้น มีความไม่แน่นอนของผลลัพธ์ค่อนข้างสูง ผู้วิจัยจึงมีความเห็นว่าหากผู้ประเมินได้เห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีมาประกอบการประเมิน ซึ่งอาจจะส่งผลให้ผู้ประเมินเปลี่ยนแปลงค่าประเมิน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จากการศึกษางานวิจัยของ (**Vicinanza,** 1991) พบว่าการประเมินโดยใช้แผนการขั้นตอนวิธี (Algorithmic Strategies) โดยผู้เชี่ยวชาญเป็นผู้ประเมิน ให้ค่าความแม่นยำน้อยกว่า การใช้แผนการอุปมา (Analogy Strategies) ผู้วิจัยจึงมีความเห็นว่าการนำตัวแบบขั้นตอนวิธีมาประกอบการพิจารณาอาจส่งผลให้ความแม่นยำในการประเมินลดลง ผู้วิจัยจึงกำหนด สมมติฐานที่ 2.2 ดังนี้

**สมมติฐานที่ 2.2**

$H_0$ : หากผู้เชี่ยวชาญได้เห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำไม่แตกต่างจาก** ค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

$H_1$ : หากผู้เชี่ยวชาญได้เห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำแตกต่างจาก** ค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

**การศึกษาเพิ่มเติมเกี่ยวกับคุณสมบัติของผู้ประเมิน**

จากงานวิจัยของ (Grimstad, 2007) สรุปได้ว่าการประเมินค่าความพยายามด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความไม่แน่นอนของผลลัพธ์ค่อนข้างสูง ความแม่นยำในการประเมินจะขึ้นอยู่กับผู้เชี่ยวชาญเป็นหลัก ทำให้ผู้วิจัยสนใจที่จะศึกษาปัจจัยต่างๆ ที่อาจส่งผลต่อความแม่นยำในการประเมินของผู้เชี่ยวชาญ

จากงานวิจัยของ (M. Jorgensen, 2005) ได้พบว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ในโครงการซอฟต์แวร์ที่มีลักษณะใกล้เคียงกันกับโครงการซอฟต์แวร์ที่ผู้ประเมินเคยพัฒนาหรือมีประสบการณ์มาก่อน จะมีความแม่นยำมากกว่า การประเมินค่าความพยายามในโครงการซอฟต์แวร์ที่ไม่เคยมีประสบการณ์มาก่อน (จากการสำรวจในบริษัทต่างๆ การประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์คลาดเคลื่อนเฉลี่ย 12% จากโครงการซอฟต์แวร์ที่ผู้ประเมินเคยมีประสบการณ์มาก่อน และการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ คลาดเคลื่อน เฉลี่ย 39% จากโครงการซอฟต์แวร์ที่ผู้ประเมินไม่เคยมีประสบการณ์มาก่อน)

ดังนั้นผู้วิจัยจึงต้องการจะศึกษาลักษณะของผู้เชี่ยวชาญที่จะส่งผลต่อความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ลักษณะของผู้เชี่ยวชาญที่ผู้วิจัยเลือกศึกษามีดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1) จำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์

ผู้วิจัยมีความเห็นว่าจำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์มีผลต่อการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ เนื่องจากผู้เชี่ยวชาญที่มีจำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์จะทราบถึงขอบเขตของการพัฒนาซอฟต์แวร์และระยะเวลาในการพัฒนาซอฟต์แวร์

2) จำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม

ผู้วิจัยมีความเห็นว่าผู้เชี่ยวชาญที่มีจำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์ไม่มากนัก แต่ได้พัฒนาโครงการซอฟต์แวร์เป็นจำนวนมาก จะทราบถึงความสัมพันธ์ของขอบเขตการพัฒนาซอฟต์แวร์และระยะเวลาในการพัฒนาซอฟต์แวร์

3) ตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์

ผู้วิจัยมีความเห็นว่าตำแหน่งหน้าที่มีผลต่อความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ เนื่องจาก มุมมองของผู้พัฒนา มุมมองผู้ดูแลหรือผู้จัดการโครงการ หรือมุมมองของผู้มีส่วนร่วมในโครงการซอฟต์แวร์ในตำแหน่งอื่นๆ อาจมีความแตกต่างกัน

4) ภาษาการเขียนโปรแกรมที่ถนัด

ผู้วิจัยมีความเห็นว่าภาษาการเขียนโปรแกรมที่ถนัดส่งผลต่อการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ เนื่องจาก ผู้เชี่ยวชาญที่ถนัดในภาษาการเขียนโปรแกรมหนึ่ง อาจมีมุมมองในการประเมินค่าความพยายามไม่เหมือนกันกับผู้เชี่ยวชาญที่ถนัดอีกภาษาการเขียนโปรแกรมหนึ่ง

จากการศึกษาของ (Vicinanza, 1991) พบว่าผู้เชี่ยวชาญที่มีประสบการณ์ในการพัฒนาซอฟต์แวร์ จะประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่คล้ายคลึงกับโครงการซอฟต์แวร์ที่เคยพัฒนามาได้อย่างแม่นยำ ด้วยเหตุนี้ผู้วิจัยมีความคิดเห็นว่าปัจจัยอื่นๆ ที่เกี่ยวข้องกับผู้เชี่ยวชาญ คือ

1) จำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์
2) จำนวนโครงการซอฟต์แวร์ที่เคยประเมินมาหรือมีส่วนร่วม
3) ตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์
4) ภาษาการเขียนโปรแกรมที่ถนัด

อาจส่งผลต่อความแม่นยำในการประเมิน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

**3.3 การออกแบบการทดลอง**

การออกแบบการทดลอง เพื่อตอบคำถามวิจัย ผู้วิจัยใช้งานวิจัยของ (**Vicinanza,** 1991) เป็นต้นแบบโดยจะติดต่อผู้เชี่ยวชาญมาจำนวน 5 คน มาประเมินโครงการที่ได้ทำสำเร็จแล้ว เพื่อนำผลการประเมินมาวิเคราะห์และทดสอบสมมติฐาน โดยการทดลองนี้มีจำนวนโครงการซอฟต์แวร์ ทั้งสิ้น 10 โครงการ ทุกโครงการมีรายละเอียดความต้องการซอฟต์แวร์ด้านหน้าที่ (Functional Requirements) และรายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่ (Non-Functional Requirements) เพื่อเป็นรายละเอียดประกอบการประเมินของผู้เชี่ยวชาญ เหตุที่ต้องใช้เพียงรายละเอียดความต้องการซอฟต์แวร์ด้านหน้าที่ และรายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้าน

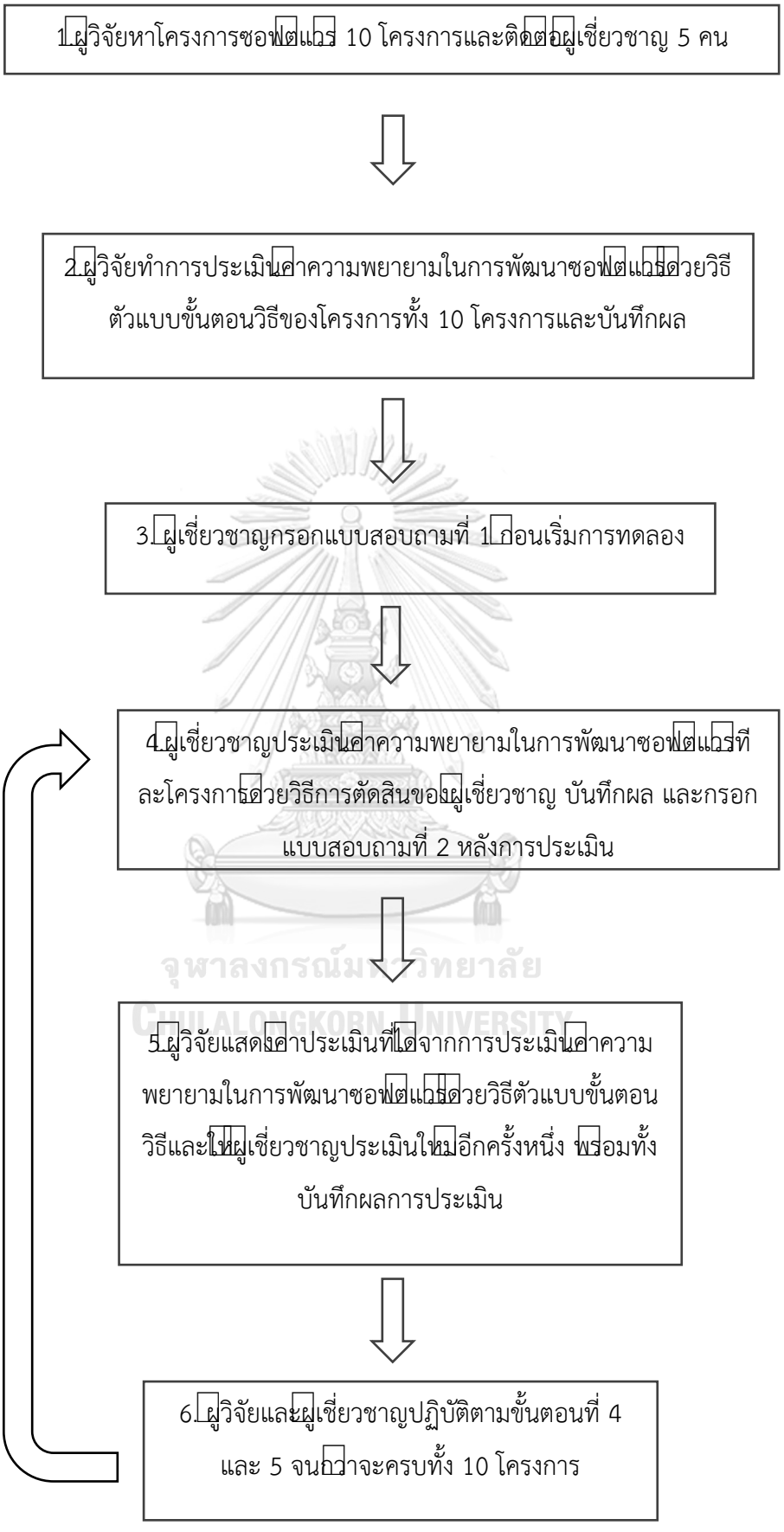
หน้าที่ เพราะว่างานวิจัยนี้มีวัตถุประสงค์เพื่อวิจัยเฉพาะความแม่นยำในช่วงเริ่มต้นของโครงการหรือช่วงการประมูลโครงการซอฟต์แวร์ เนื่องจากในเอกสารการจัดซื้อจัดจ้างซอฟต์แวร์ในปัจจุบันจะมีการระบุถึงรายละเอียดความต้องการซอฟต์แวร์ด้านหน้าที่ และรายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่ ด้วยเสมอ

สามารถเขียนขั้นตอนการดำเนินการได้ดังนี้

**ภาพรวมขั้นตอนการดำเนินการทดลอง**



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY




จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.3.1 ผู้วิจัยหาโครงการซอฟต์แวร์ 10 โครงการ และติดต่อผู้เชี่ยวชาญ 5 คน

จากการทบทวนวรรณกรรมที่เกี่ยวข้องกับการเปรียบเทียบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ทั้ง 5 โครงการ สามารถแสดงข้อสรุปผลการเลือกผู้เชี่ยวชาญและจำนวนโครงการซอฟต์แวร์ได้ดังตารางที่ 33

ตารางที่ 33 จำนวนโครงการที่ใช้ในการทดลองและจำนวนผู้เชี่ยวชาญที่เข้าร่วมการทดลองในงานวิจัยทั้ง **5** งานวิจัย

| งานวิจัย | จำนวนโครงการซอฟต์แวร์ที่ใช้ในการทดลอง | จำนวนผู้เชี่ยวชาญที่เข้าร่วมการทดลอง |
|---|---|---|
| Kusters et al., 1990 | 1 โครงการ ชื่อ Bonus System | บุคคลทำงานในตำแหน่งหัวหน้าโครงการ (Project Leader) จำนวน 14 คน |
| Vicinanza et al., 1991 | 10 โครงการแรกจาก (Kemerer, 1987) | บุคคลทำงานในตำแหน่ง ผู้จัดการโครงการ (Project Manager) ผู้เชี่ยวชาญต้นทุนโครงการ (Cost Manager) 5 คน มีประสบการณ์ในการทำงานขั้นต่ำ 10 ปี |
| Mukhopadhyay et al., 1992 | 15 โครงการ จาก (Kemerer, 1987) | ผู้จัดการการพัฒนาซอฟต์แวร์ 1 คน ทำงานในบริษัทซอฟต์แวร์ขนาดใหญ่ มีประสบการณ์ทำงาน 10 ปี และมีประสบการณ์ 9 ปีในงานที่เกี่ยวข้องกับแอปพลิเคชันการประมวลผลข้อมูล (Data Processing Application) |
| Atkinson and Shepperd, 1994 | 1 โครงการ | ไม่มีข้อมูล |
| Pengelly, 1995 | 1 โครงการ | ไม่มีข้อมูล |
| ขวัญชนก ธรณธรรมกุล และปิติพัฒน์ เจิมบุญธนะพงศ์ (2558) | 1 โครงการ | ไม่มีข้อมูล |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ด้วยเหตุนี้ งานวิจัยชิ้นนี้จึงมีการเลือกจำนวนโครงการซอฟต์แวร์และจำนวนผู้เชี่ยวชาญในการประเมินดังต่อไปนี้

1) การเลือกโครงการซอฟต์แวร์

งานวิจัยของ Kusters et al. (1990), Atkinson and Shepperd (1994) และ Pengelly (1995) มีการใช้โครงการซอฟต์แวร์ในการทดลอง เพียงโครงการเดียว ซึ่งทำให้ผลการประเมินที่เกิดขึ้นไม่น่าเชื่อถือเท่าที่ควร

งานวิจัยของ Vicinanza el at. (1991) และ Mukhopadhyay et al. (1992) ได้ใช้จำนวนโครงการซอฟต์แวร์จำนวน 10 และ 15 โครงการ ตามลำดับ แต่ล้วนเป็นข้อมูลโครงการจาก Kemerer (1987) ทั้งสิ้น เนื่องด้วยข้อมูลโครงการจาก Kemerer เป็นข้อมูลที่เก่า เทคโนโลยีที่ใช้ในการพัฒนาซอฟต์แวร์ในสมัยนั้นแตกต่างจากยุคสมัยปัจจุบัน อีกทั้งข้อมูลของ Kemerer นั้นไม่ได้มีรายละเอียดความต้องการซอฟต์แวร์ด้านหน้าที่ และรายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่ ผู้วิจัยจึงไม่นำข้อมูลของ Kemerer มาใช้ในการทดลองนี้

โครงการซอฟต์แวร์ที่นำมาใช้ในการทดลองเป็นโครงการซอฟต์แวร์ที่เกิดขึ้นจริง ผู้วิจัยจะติดต่อบริษัทที่ดำเนินธุรกิจทางด้านซอฟต์แวร์เพื่อขอข้อมูลการพัฒนาโครงการซอฟต์แวร์ทั้งหมด 10 โครงการ เนื่องจากงานวิจัยของ Vicinanza et al. (1991) ได้ใช้จำนวนโครงการในการทดลอง 10 โครงการ ผู้วิจัยมีความเห็นว่าในงานวิจัยนี้ผู้วิจัยสามารถหาโครงการซอฟต์แวร์ที่จะนำมาใช้ในการทดลองได้ 10 โครงการ เทียบเท่ากับงานวิจัยของ Vicinanza และมีรายละเอียดของความต้องการซอฟต์แวร์ด้านหน้าที่ กับ รายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยดำเนินการรวบรวมข้อมูลโครงการซอฟต์แวร์ ทั้ง 10 โครงการ ได้แก่ ชื่อโครงการ, ประเภทโครงการ, ภาษาที่ใช้ในการพัฒนา, จำนวนบรรทัดในการเขียน (Source Line of Code) โครงการซอฟต์แวร์ทั้ง 10 โครงการนี้มาจากโครงการซอฟต์แวร์ที่ผู้วิจัยพัฒนาเองหรือมีส่วนร่วมในการพัฒนาโครงการซอฟต์แวร์เหล่านี้เป็นโครงการที่พึ่งพัฒนาสำเร็จตั้งแต่ปี 2557 เป็นต้นมา ผู้วิจัยนำข้อมูลจากการสอบถามเพิ่มเติมในทีมร่วมพัฒนาและข้อมูลจากเอกสารจัดซื้อจัดจ้าง มาสรุปในรูปแบบของความต้องการซอฟต์แวร์ด้านหน้าที่และไม่ใช่ด้านหน้าที่ เพื่อนำไปเป็นข้อมูลนำเข้าในการทดลองต่อไป

**ตัวอย่างของความต้องการซอฟต์แวร์ด้านหน้าที่และไม่ใช่ด้านหน้าที่ มีตัวอย่างอยู่ในภาคผนวก ข.**

โครงการซอฟต์แวร์ทั้งหมดเป็นโครงการซอฟต์แวร์ที่ผ่านวงจรชีวิตของการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) แบบ อไจล์ (Agile) ซึ่งมีงานวิจัยที่ใช้วิธีการตัดสินของ

ผู้เชี่ยวชาญหรือตัวแบบขั้นตอนวิธีเป็นวิธีในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์แบบอไจล์อยู่ในปัจจุบัน ตัวอย่างเช่น

งานวิจัยของ (Popli, 2014) กล่าวถึงการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ โดยซอฟต์แวร์ในการทดลองใช้ อไจล์ เป็นวงจรชีวิตของการพัฒนาซอฟต์แวร์ และงานวิจัยของ (Kang, 2010) กล่าวถึงการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี Function Point โดยซอฟต์แวร์ในการทดลองใช้ อไจล์ เป็นวงจรชีวิตของการพัฒนาซอฟต์แวร์

โครงการซอฟต์แวร์ที่ผู้วิจัยนำมาใช้ในการทดลอง ผู้วิจัยใช้คำสั่งลินิกซ์ในการนับบรรทัด โดยนับเฉพาะจำนวนบรรทัดของซอฟต์แวร์ที่นักพัฒนาได้พัฒนาขึ้นมา (ไม่นับรวมการทำงานในส่วนอื่นๆที่นักพัฒนาเจ้าของโครงการซอฟต์แวร์มิได้เป็นผู้เขียนขึ้น) รวมไปถึงจัดทำเอกสารความต้องการซอฟต์แวร์ทั้ง 10 โครงการ ส่งมอบให้นักพัฒนาโครงการซอฟต์แวร์หรือเจ้าของโครงการซอฟต์แวร์นั้นๆ ได้พิจารณาความถูกต้องและครอบคลุมของเอกสารความต้องการในการพัฒนาซอฟต์แวร์ซึ่งประกอบด้วยรายละเอียดของความต้องการซอฟต์แวร์ด้านหน้าที่ กับ รายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่ เพื่อให้ผู้วิจัยได้มั่นใจว่าเอกสารความต้องการโครงการซอฟต์แวร์มีความถูกต้องและครอบคลุมพร้อมสำหรับการทดลอง รายละเอียดโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงดังตารางที่ 34

ตารางที่ 34 รายละเอียดซอฟต์แวร์ที่นำมาใช้ในการทดลองทั้ง 10 โครงการซอฟต์แวร์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| ลำดับโครงการซอฟต์แวร์ | ชื่อโครงการซอฟต์แวร์ | ชนิดของโครงการซอฟต์แวร์ | ภาษาที่ใช้ในการพัฒนา | จำนวนบรรทัดของซอฟต์แวร์ (Source Line of Code) | ค่าความพยายามที่ใช้จริง (Actual Man-month) |
|---|---|---|---|---|---|
| 1 | HHTP | Cross platform mobile application | JavaScript | 8371 | 2.5 |
| 2 | PTT | Android Application | Java | 11131 | 7.5 |
| 3 | KAPP | Cross platform mobile application | JavaScript | 9258 | 8.5 |
| 4 | EATD | Android Application | Java | 6532 | 5 |
| 5 | ST | Web application | JavaScript | 35974 | 14 |

| ลำดับโครงการซอฟต์แวร์ | ชื่อโครงการซอฟต์แวร์ | ชนิดของโครงการซอฟต์แวร์ | ภาษาที่ใช้ในการพัฒนา | จำนวนบรรทัดของซอฟต์แวร์ (Source Line of Code) | ค่าความพยายามที่ใช้จริง (Actual Man-month) |
|---|---|---|---|---|---|
| 6 | CINOUT | Cross platform mobile application | JavaScript | 1596 | 3 |
| 7 | MINSH | Web application | JavaScript, PHP | 10115 | 3 |
| 8 | LMS | Cross platform mobile application | JavaScript | 4210 | 3 |
| 9 | LEAP | Web Application | JavaScript | 3409 | 5.5 |
| 10 | TM | Web Application | JavaScript | 2440 | 4 |

- Cross platform mobile application คือ แอปพลิเคชันที่ถูกพัฒนาเพียงหนึ่งครั้ง แต่สามารถประมวลผลได้ในแพลตฟอร์มทั้ง ระบบปฏิบัติการแอนดรอยด์และระบบปฏิบัติการไอโอเอส
- Android Application คือ แอปพลิเคชันที่ถูกพัฒนาสำหรับประมวลผลในระบบปฏิบัติการแอนดรอยด์
- Web Application คือ แอปลิเคชันสำหรับการใช้งานบนเว็บเพจ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 35 คำอธิบายโครงการซอฟต์แวร์โดยย่อ

| ลำดับโครงการซอฟต์แวร์ | ชื่อโครงการซอฟต์แวร์ | คำอธิบายโดยย่อ |
|---|---|---|
| 1 | HHTP | โปรแกรมประยุกต์บนอุปกรณ์เคลื่อนที่สำหรับการบันทึกข้อมูลผลการผลิตในโรงงานอุตสาหกรรม |
| 2 | KAPP | โปรแกรมประยุกต์บนอุปกรณ์เคลื่อนที่สำหรับการแสดงข่าวสาร แผนที่ โปรโมชันร้านค้าต่างๆรอบมหาวิทยาลัย |
| 3 | MINSH | เว็บแอปพลิเคชันสำหรับซื้อเกมออนไลน์บนแพลตฟอร์มของสตรีม |
| 4 | PTT | โปรแกรมประยุกต์บนอุปกรณ์เคลื่อนที่สำหรับถ่ายรูป และตกแต่งรูปภาพ |
| 5 | ED | โปรแกรมประยุกต์บนอุปกรณ์เคลื่อนที่สำหรับแนะนำมื้ออาหาร |

| ลำดับโครงการซอฟต์แวร์ | ชื่อโครงการซอฟต์แวร์ | คำอธิบายโดยย่อ |
|---|---|---|
| 6 | BS | เว็บแอปพลิเคชันส่งเสริมการขายของธุรกิจก่อสร้าง |
| 7 | LMS | โปรแกรมประยุกต์บนอุปกรณ์เคลื่อนที่สำหรับการส่งเสริมการเรียนรู้ |
| 8 | ST | เว็บแอปพลิเคชันสำหรับการซื้อสินทรัพย์ดิจิทัลและสร้างกระเป๋าสำหรับเก็บสินทรัพย์ดิจิทัล |
| 9 | 5Bath | เว็บแอปพลิเคชันสำหรับซื้อสินทรัพย์ดิจิทัลและส่งให้กับปลายทาง |
| 10 | Trafero | เว็บแอปพลิเคชันช่วยอำนวยความสะดวกในการซื้อสินทรัพย์ดิจิทัล |

**ตัวอย่างความต้องการซอฟต์แวร์ด้านหน้าที่และไม่ใช่ด้านหน้าที่ จะอยู่ในภาคผนวก ข.**

2) การเลือกผู้เชี่ยวชาญ

งานวิจัยของ Vicinanza et al. (1991) ได้ใช้ผู้เชี่ยวชาญในการประเมินค่าความพยายาม 5 คน โดยได้ระบุคุณสมบัติของผู้เชี่ยวชาญ คือ เป็นบุคคลที่ทำงานในบริษัทซอฟต์แวร์ขนาดใหญ่ มีประสบการณ์ทำงาน 10 ปีและมีประสบการณ์ทำงานมากกว่า 9 ปีในโครงการซอฟต์แวร์ที่เกี่ยวข้องกับแอปพลิเคชันการประมวลผลข้อมูล (Data Processing Application)

จากงานวิจัยของ Mukhopadhyay et al. (1992) ได้กำหนดผู้เชี่ยวชาญเป็นผู้จัดการนักพัฒนาซอฟต์แวร์ (Software Developer Manager) จำนวน 1 คน ที่ทำงานอยู่ในบริษัทซอฟต์แวร์ขนาดใหญ่ เป็นผู้เชี่ยวชาญค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ค่าประเมินจากผู้เชี่ยวชาญเพียง 1 คน จึงไม่น่าเชื่อถือเท่าที่ควร

งานวิจัยของ Walkerden and Jeffery (1999) และ Bowden et al. (2000) ได้กำหนดผู้เชี่ยวชาญเป็นนักศึกษา มีจำนวน 25 คน ส่วน Bowden ทำการทดลองโดยใช้นักศึกษาทั้งหมด 24 กลุ่ม โดยแบ่งกลุ่มละ 3-5 คน ซึ่งผู้เชี่ยวชาญเป็นนักศึกษาที่ไม่ได้ระบุคุณสมบัติของผู้เชี่ยวชาญ ค่าประเมินที่ได้จากผู้เชี่ยวชาญที่เป็นนักศึกษานั้น ไม่น่าเชื่อถือ เนื่องมาจากนักศึกษาอาจไม่ใช่ผู้เชี่ยวชาญในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์

จากงานวิจัยของ Jorgensen (2005) ได้พบว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ในโครงการซอฟต์แวร์ที่มีลักษณะใกล้เคียงกันกับโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีประสบการณ์มาก่อน จะมีความแม่นยำมากกว่า การประเมินค่าความพยายามในโครงการซอฟต์แวร์ที่ไม่เคยมีประสบการณ์มาก่อน (จากการสำรวจในบริษัทต่างๆ การประมาณค่าความพยายามในการ

พัฒนาซอฟต์แวร์คลาดเคลื่อนเฉลี่ย 12% จากโครงการซอฟต์แวร์ที่ผู้ประเมินเคยมีประสบการณ์มาก่อน และการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์คลาดเคลื่อน เฉลี่ย 39% จากโครงการซอฟต์แวร์ที่ผู้ประเมินไม่เคยมีประสบการณ์มาก่อน)

ด้วยเหตุนี้ผู้วิจัยจึงออกแบบการทดลอง โดยติดต่อผู้เข้าร่วมการทดลองที่เป็นผู้เชี่ยวชาญมีประสบการณ์ในการพัฒนาซอฟต์แวร์ในซอฟต์แวร์ชนิดเดียวกันกับซอฟต์แวร์ที่ผู้วิจัยเตรียมไว้สำหรับการทดลองจำนวน 5 ท่าน จำนวนเทียบเท่ากับผู้เชี่ยวชาญในการทดลองของ Vicinanza มาจากบริษัทที่แตกต่างกัน 5 บริษัทที่ดำเนินธุรกิจเกี่ยวกับการพัฒนาซอฟต์แวร์ในประเทศไทย ผู้เชี่ยวชาญทั้ง 5 ท่าน มีประสบการณ์หรือมีส่วนร่วมในการพัฒนาซอฟต์แวร์ด้านแอปพลิเคชันการประมวลผลข้อมูล (Data Processing Application) ไม่ต่ำกว่า 10 ปี มีตำแหน่งงานในบริษัทล่าสุดในตำแหน่งงานเกี่ยวกับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ หรือ ตำแหน่งระดับผู้จัดการโครงการหรือใกล้เคียง

ผู้วิจัยจะติดต่อผู้เชี่ยวชาญที่มีประสบการณ์ในการพัฒนาซอฟต์แวร์ในยุคสมัยปัจจุบันที่มีประสบการณ์ในการพัฒนาซอฟต์แวร์ด้วยภาษาหรือเครื่องมือที่ช่วยในการพัฒนาซอฟต์แวร์สมัยใหม่ และผู้เชี่ยวชาญเหล่านั้นได้รับการยอมรับในกลุ่มนักพัฒนาซอฟต์แวร์ในบริษัทว่ามีประสบการณ์ในการพัฒนาซอฟต์แวร์ ตัวอย่างเช่น งานวิจัยของ Vicinanza et al. (1991) ที่ระบุคุณสมบัติไว้ว่าผู้เชี่ยวชาญ จะได้รับการยอมรับว่าเป็นผู้ที่สามารถประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ได้อย่างแม่นยำจากบุคคลอื่นๆ ในทีม การเลือกผู้เชี่ยวชาญจะติดต่อกับบริษัทที่ผู้วิจัยรู้จักและให้ความร่วมมือ

3.3.2 ผู้วิจัยประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีของโครงการจำนวน 10 โครงการและบันทึกผล

โครงการซอฟต์แวร์ทั้ง 10 โครงการจะนำมาประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย 3 ตัวแบบขั้นตอนวิธี คือ โคโคโม่ (COCOMO Basic Model), ฟังก์ชันพอยต์ (Function Point), และยูสเคสพอยต์ (Use Case Point) ประเมินโดยผู้วิจัยและตรวจสอบความถูกต้องโดยอาจารย์ที่ปรึกษา

3.3.3 ผู้เชี่ยวชาญกรอกแบบสอบถามที่ 1 ก่อนเริ่มการทดลอง

ผู้เชี่ยวชาญจะได้รับแบบสอบถามเพื่อให้ข้อมูลเบื้องต้นของผู้เชี่ยวชาญ แบบสอบถามมีคำถามดังนี้

1. คำถามข้อมูลเบื้องต้นของผู้เชี่ยวชาญเพื่อใช้พิจารณาประกอบความคล้ายคลึงกันของกลุ่มทดลอง
   1) เพศ

2) อายุ

3) วุฒิการศึกษาสูงสุด

2. คำถามเกี่ยวกับปัจจัยที่ผู้วิจัยคาดว่าน่าจะเกี่ยวข้องกับความแม่นยำในการประเมินค่าความพยายาม

4) ท่านเคยทำงานอยู่ตำแหน่งใดบ้าง ระยะเวลาทำงานในแต่ละตำแหน่งนานเท่าไร

5) ท่านเคยพัฒนาหรือมีส่วนร่วมในโครงการซอฟต์แวร์ทั้งหมดกี่โครงการ แต่ละโครงการซอฟต์แวร์นั้นท่านมีส่วนร่วมกี่เดือน (ตัวอย่างคำตอบเช่น 5 โครงการ โดย โครงการที่1 – 2 เดือน โครงการที่ 2 – 3เดือน ...)

6) เรียงลำดับภาษาการเขียนโปรแกรมที่ท่านถนัดจากความถนัดมาก -> น้อย จากภาษาที่กำหนดให้ต่อไปนี้ Javascript, PHP, C#, ClojureScript, JAVA, Python

3. คำถามเกี่ยวกับความรู้ในเรื่องตัวแบบขั้นตอนวิธี ใช้ในการพิจารณาประกอบในการทดสอบสมมติฐานที่ 3.1 ว่าผู้เชี่ยวชาญรู้จักการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีทั้งสามมาก่อนหรือไม่ ก่อนจะตัดสินใจเปลี่ยน/ไม่เปลี่ยนแปลงค่าประเมิน ภายหลังการเห็นค่าประเมินจากตัวแบบขั้นตอนวิธีทั้งสาม

7) ท่านรู้จักวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่ (COCOMO) มาก่อนหรือไม่

8) ท่านรู้จักวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีฟังก์ชันพอยต์ (Function Point) มาก่อนหรือไม่

9) ท่านรู้จักวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยต์ (Use Case Point)

**รายละเอียดแบบสอบถามที่ 1 ก่อนเริ่มการทดลองจะอยู่ในภาคผนวก ค.**

3.3.4 ผู้เชี่ยวชาญประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ บันทึกผล และกรอกแบบสอบถามที่ 2 หลังการประเมิน

มีขั้นตอนดังนี้

ก. ผู้เชี่ยวชาญรับเอกสารความต้องการซอฟต์แวร์ ที่ประกอบด้วยรายละเอียดของความต้องการซอฟต์แวร์ด้านหน้าที่ กับ รายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่

ผู้เชี่ยวชาญรับเอกสารความต้องการของโครงการซอฟต์แวร์ โดยระบุการอธิบายภาพรวมของซอฟต์แวร์ว่าโครงการซอฟต์แวร์นั้นเป็นซอฟต์แวร์อะไร และระบุรายละเอียดของความต้องการซอฟต์แวร์ด้านหน้าที่ กับ รายละเอียดความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่

ข. ผู้เชี่ยวชาญประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ และบันทึกผล

ผู้เชี่ยวชาญประเมินค่าความพยายามแยกห้องกัน ภายในห้องปิด และไม่มีเสียงรบกวน เมื่อผู้เชี่ยวชาญประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เสร็จ ผู้เชี่ยวชาญจะบันทึกค่าประเมินในแบบฟอร์มบันทึกค่าประเมิน ดังตารางที่ 36 เมื่อผู้เชี่ยวชาญบันทึกเรียบร้อยแล้วจะแจ้งผู้วิจัยเพื่อรับแบบสอบถามที่ 2 หลังการประเมิน

ตารางที่ 36 แบบฟอร์มบันทึกค่าประเมิน

| ลำดับโครงการซอฟต์แวร์ | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (หน่วย คน-เดือน) | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (หน่วย คน-เดือน) ในการประเมินหลังจากเห็นค่าประเมินจากวิธีตัวแบบขั้นตอนวิธี |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ค. ผู้เชี่ยวชาญกรอกแบบสอบถามที่ 2 หลังการประเมิน

เมื่อผู้เชี่ยวชาญประเมินเสร็จในแต่ละโครงการ ผู้เชี่ยวชาญจะได้รับแบบสอบถามหลังการประเมิน แบบสอบถามมีลักษณะดังนี้

1) ท่านเคยพัฒนาโครงการซอฟต์แวร์ที่มีฟังก์ชั่นการทำงานใกล้เคียง หรือคล้ายคลึงกันหรือไม่อย่างไร (คำถามนี้ใช้พิจารณาประกอบผลการทดลอง อาจเป็นไปได้ว่าผู้ประเมินที่มี

ประสบการณ์พัฒนาซอฟต์แวร์ที่มีลักษณะการทำงานใกล้เคียงกันมาก่อน จะประเมินได้แม่นยำมากกว่าผู้ประเมินที่ไม่มีประสบการณ์ในซอฟต์แวร์ที่ใกล้เคียงซอฟต์แวร์ในการทดลองนี้)

2) ในการประเมินครั้งนี้ ท่านมีวิธีการประเมินอย่างไร อธิบาย (คำถามนี้ เป็นคำถามปลายเปิด อาจมาใช้ในการพิจารณาประกอบผลการทดลอง)

**รายละเอียดแบบสอบถามที่ 2 หลังการประเมินจะอยู่ในภาคผนวก ง.**

3.3.5 ผู้วิจัยแสดงค่าประเมินที่ได้จากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีและให้ผู้เชี่ยวชาญใช้ประกอบการประเมินใหม่อีกครั้งหนึ่ง พร้อมทั้งบันทึกผลการประเมิน

เมื่อผู้เชี่ยวชาญได้ทำการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เสร็จเรียบร้อยแล้ว ผู้วิจัยจะแสดงค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีทั้ง 3 ตัวแบบขั้นตอนวิธีโคโคโม่, ฟังก์ชันพอยต์ และ ยูสเคสพอยต์ เพื่อให้ผู้เชี่ยวชาญทำการติดสินใจใหม่อีกครั้งหนึ่งว่าจะเปลี่ยนแปลงคำตอบที่ได้จากการ ประเมินของตัวเองหรือไม่ ซึ่งผู้เชี่ยวชาญสามารถเลือกที่จะยืนยันค่าประเมินเดิม หรือ เปลี่ยนค่าประเมิน ผู้เชี่ยวชาญจะบันทึกค่าประเมินลงในแบบฟอร์มการบันทึกข้อมูลค่าประเมิน หากผู้เชี่ยวชาญเลือกที่จะประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ใหม่

3.3.6 ผู้วิจัยและผู้เชี่ยวชาญปฏิบัติตามขั้นตอนที่ 3.3.4 และ 3.3.5 จนครบทั้ง 10 โครงการ

ผู้เชี่ยวชาญจะทำการประเมินตามข้อ 3.3.4 และ 3.3.5 จนกว่าผู้เชี่ยวชาญจะประเมินโครงการซอฟต์แวร์ครบทั้ง 10 โครงการ

**การทดลองนำร่อง**

ผู้วิจัยจะทำการทดลองนำร่องเพื่อทดสอบว่าแผนการทดลองจะไม่มีอุปสรรค ปัญหาใดๆ โดยทำการทดลองกับผู้ประเมินที่เป็นเพื่อนของผู้วิจัยเอง จำนวน 2 ท่าน ตามขั้นตอนที่ 3.3.1 ถึง 3.3.6 ผู้วิจัยจะดำเนินการตามขั้นตอนการทดลอง และบันทึกผลการทดลอง

**3.4 วิเคราะห์ผลการทดลอง**

3.4.1 การหาค่าความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีและค่าประเมินที่ได้รับจากผู้เชี่ยวชาญ

หลังจากที่ผู้เชี่ยวชาญทำการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เสร็จเรียบร้อยแล้ว ผู้วิจัยจะสรุปผลค่าประเมินที่ได้จากวิธีต่างๆ และค่าพยายามในการพัฒนาซอฟต์แวร์จริง ดังตารางที่ 37

ตารางที่ 37 สรุปผลค่าประเมินของตัวแบบขั้นตอนวิธีทั้ง 3 วิธี ค่าประเมินของผู้เชี่ยวชาญ และค่าความพยายามในการพัฒนาซอฟต์แวร์จริง

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าประเมินของวิธีตัวแบบขั้นตอนวิธี (หน่วย คน-เดือน) | | | ค่าประเมินของผู้เชี่ยวชาญ (หน่วย คน-เดือน) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | โคโคโม่ | ฟังก์ชันพอยต์ | ยูสเคสพอยต์ | ผู้เชี่ยวชาญคนที่ 1 | | ผู้เชี่ยวชาญคนที่ 2 | | ผู้เชี่ยวชาญคนที่ 3 | | ผู้เชี่ยวชาญคนที่ 4 | | ผู้เชี่ยวชาญคนที่ 5 | |
| | | | | | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** |
| 1 | HHTP | | | | | | | | | | | | | |
| 2 | PTT | | | | | | | | | | | | | |
| 3 | KAPP | | | | | | | | | | | | | |
| 4 | EATD | | | | | | | | | | | | | |
| 5 | ST | | | | | | | | | | | | | |
| 6 | CINOUT | | | | | | | | | | | | | |
| 7 | MINSH | | | | | | | | | | | | | |
| 8 | LMS | | | | | | | | | | | | | |
| 9 | LEAP | | | | | | | | | | | | | |
| 10 | TM | | | | | | | | | | | | | |

CHULALONGKORN UNIVERSITY

เมื่อผู้ประเมินทุกคนได้ทำการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จนครบทุกโครงการแล้ว ผู้วิจัยจะคำนวณค่าความคลาดเคลื่อนสัมพัทธ์ หรือ Magnitude Relative Error (MRE) ในแต่ละค่าการประเมินของผู้เชี่ยวชาญแต่ละท่าน (Fenton and Pfleeger, 1997)

สูตรการหาค่า MRE

$$MRE = \left| \frac{MMest - MMact}{MMact} \right|$$

โดย

MMact คือ ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง

MMest คือ ค่าประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จากผู้เชี่ยวชาญ

ค่าความคลาดเคลื่อนสัมพัทธ์ (MRE) จะบันทึกลงในตารางที่ 38

ตารางที่ 38 ฟอร์มบันทึกค่าความคลาดเคลื่อนสัมพัทธ ของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบขั้นตอนวิธี

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี | | | ค่าความคลาดเคลื่อนสัมพัทธ์ของผู้เชี่ยวชาญในการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | โคโคโม่ | ฟังก์ชันพอยต์ | ยูสเคสพอยต์ | ผู้เชี่ยวชาญคนที่ 1 | | ผู้เชี่ยวชาญคนที่ 2 | | ผู้เชี่ยวชาญคนที่ 3 | | ผู้เชี่ยวชาญคนที่ 4 | | ผู้เชี่ยวชาญคนที่ 5 | |
| | | | | | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** | **ครั้งที่ 1** | **ครั้งที่ 2** |
| 1 | HHTP | | | | | | | | | | | | | |
| 2 | PTT | | | | | | | | | | | | | |
| 3 | KAPP | | | | | | | | | | | | | |
| 4 | EATD | | | | | | | | | | | | | |
| 5 | ST | | | | | | | | | | | | | |
| 6 | CINOUT | | | | | | | | | | | | | |
| 7 | MINSH | | | | | | | | | | | | | |
| 8 | LMS | | | | | | | | | | | | | |
| 9 | LEAP | | | | | | | | | | | | | |
| 10 | TM | | | | | | | | | | | | | |
| MMRE | | | | | | | | | | | | | | |

Chulalongkorn University จุฬาลงกรณ์มหาวิทยาลัย

3.4.2 การทดสอบสมมติฐาน

**ก) การทดสอบสมมติฐานที่ 1**

**<u>คำถามวิจัย</u>**

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ภายในสภาวะแวดล้อมที่ผู้เชี่ยวชาญเป็นคนไทยด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าหรือน้อยกว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี ?

**<u>สมมติฐาน</u>**

$H_0$ : ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีขั้นตอนวิธี **มากกว่าหรือเท่ากับ**ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

$H_1$ : ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีขั้นตอนวิธี (Algorithmic Model) **น้อยกว่า** ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

กำหนดให้

$\mu_1$ คือ ค่าเฉลี่ยของ การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่

$\mu_2$ คือ ค่าเฉลี่ยของ การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีฟังก์ชันพอยต์

$\mu_3$ คือ ค่าเฉลี่ยของ การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยต์

$\mu_4$ คือ ค่าเฉลี่ยของ การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

สามารถเขียนสมมติฐานได้ว่า

$H_0 = \mu_1 = \mu_2 = \mu_3 = \mu_4$

$H_1 = \mu_i \neq \mu_j$ อย่างน้อย 1 ค่า เมื่อ $i \neq j\ ;\ i,j = 1,2,3,4$

**<u>การทดสอบสมมติฐาน</u>**

ในการทดลองนี้ ผู้วิจัยจะใช้การทดสอบการกระจายของข้อมูล (Normality Test) หากการกระจายของข้อมูลมีการกระจายแบบปกติ ผู้วิจัยจะใช้การวิเคราะห์ความแปรปรวน (Analysis of variance:ANOVA) แต่หากข้อมูลมีการกระจายแบบไม่ปกติ ผู้วิจัยจะใช้การทดสอบ Kuskal-Wallis ผู้วิจัยได้สมมติตัวอย่างของข้อมูลในตารางที่ 3.7 และ 3.8 และคำนวณหาค่าความคลาดเคลื่อนเฉลี่ยดังแสดงในตารางที่ 39

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 39 ข้อมูลสมมติของผลการคำนวณค่าความคลาดเคลื่อนสัมพัทธ์และค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของค่าประเมินด้วยวิธีตัวแบบวิธี

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง (คน-เดือน) | ค่าความพยายามในการพัฒนาซอฟต์แวร์จากการประเมินด้วยวิธีตัวแบบโคโคโม่ (คน-เดือน) | ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามด้วยวิธีตัวแบบโคโคโม่ (MRE of COCOMO) | ค่าความพยายามในการพัฒนาซอฟต์แวร์จากการประเมินด้วยวิธีตัวแบบฟังก์ชันพอยต์ (คน-เดือน) | ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามด้วยวิธีตัวแบบฟังก์ชันพอยต์ (MRE of Function Point) | ค่าความพยายามในการพัฒนาซอฟต์แวร์จากการประเมินด้วยวิธีตัวแบบยูสเคสพอยต์ (คน-เดือน) | ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามด้วยวิธีตัวแบบยูสเคสพอยต์ (MRE of Use Case Point) |
|---|---|---|---|---|---|---|---|---|
| 1 | HHTP | 2.5 | 9 | 2.6 | 7 | 1.8 | 7 | 1.8 |
| 2 | PTT | 8.5 | 18 | 1.12 | 19 | 1.24 | 10 | 0.18 |
| 3 | KAPP | 3 | 6 | 1 | 7 | 1.33 | 5 | 0.66 |
| 4 | EATD | 7.5 | 19 | 1.53 | 16 | 1.13 | 16 | 1.13 |
| 5 | ST | 3.5 | 13 | 2.71 | 13 | 2.71 | 17 | 3.86 |
| 6 | CINOUT | 15.5 | 33 | 1.13 | 27 | 0.74 | 14 | 0.1 |
| 7 | MINSH | 3 | 13 | 3.33 | 15 | 4 | 27 | 8 |
| 8 | LMS | 2.5 | 8 | 2.2 | 10 | 3 | 7 | 1.8 |
| 9 | LEAP | 2.5 | 7.5 | 2 | 6 | 1.4 | 9 | 2.6 |
| 10 | TM | 3.5 | 9 | 2 | 10 | 1.86 | 8 | 1.29 |
| ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (MMRE) | | | | 1.96 | | 1.92 | | 2.14 |

ตารางที่ 40 ข้อมูลสมมติของผลการคำนวณค่าความคลาดเคลื่อนสัมพัทธ์และค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของค่าประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง (คน-เดือน) | ค่าความพยายามและค่าความคลาดเคลื่อนจากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ค่าประเมินผู้เชี่ยวชาญคนที่ 1 (คน-เดือน) | MRE ของผู้เชี่ยวชาญคนที่ 1 | ค่าประเมินผู้เชี่ยวชาญคนที่ 2 (คน-เดือน) | MRE ของผู้เชี่ยวชาญคนที่ 2 | ค่าประเมินผู้เชี่ยวชาญคนที่ 3 (คน-เดือน) | MRE ผู้เชี่ยวชาญคนที่ 3 | ค่าประเมินผู้เชี่ยวชาญคนที่ 4 (คน-เดือน) | MRE ผู้เชี่ยวชาญคนที่ 4 | ค่าประเมินผู้เชี่ยวชาญคนที่ 5 (คน-เดือน) | MRE ผู้เชี่ยวชาญคนที่ 5 |
| 1 | HHTP | 2.5 | 3.5 | 0.4 | 4 | 0.6 | 4 | 0.6 | 2 | 0.2 | 4 | 0.6 |
| 2 | PTT | 8.5 | 10 | 0.18 | 6 | 0.18 | 7 | 0.18 | 5 | 0.42 | 10 | 0.18 |
| 3 | KAPP | 3 | 5 | 0.66 | 4 | 0.33 | 4 | 0.33 | 1.5 | 0.5 | 4 | 0.33 |
| 4 | EATD | 7.5 | 9 | 0.2 | 5 | 0.33 | 5 | 0.33 | 4 | 0.47 | 5 | 0.5 |
| 5 | ST | 3.5 | 4 | 0.14 | 5 | 0.43 | 6 | 0.71 | 2 | 0.43 | 5 | 0.43 |
| 6 | CINOUT | 15.5 | 17 | 0.1 | 5 | 0.68 | 9 | 0.42 | 5 | 0.68 | 11 | 0.29 |
| 7 | MINSH | 3 | 5 | 0.66 | 6 | 1 | 6 | 1 | 2 | 0.33 | 4 | 0.33 |
| 8 | LMS | 2.5 | 5 | 1 | 4 | 0.6 | 6 | 1.4 | 3 | 0.2 | 6 | 1.4 |
| 9 | LEAP | 2.5 | 4 | 0.6 | 6 | 0.58 | 6 | 0.58 | 3 | 0.2 | 7 | 1.8 |
| 10 | TM | 3.5 | 7 | 1 | 6 | 0.71 | 6 | 0.71 | 3 | 0.14 | 8 | 1.29 |
| | | | | | | | | | | | | |
| ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (MMRE) | | | | 0.49 | | 0.54 | | 0.63 | | 0.58 | | 0.71 |

ตารางที่ 41 ตัวอย่างตารางสรุปผลค่าความคลาดเคลื่อนเฉลี่ย ของแต่ละวิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์

| วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ | จำนวนโครงการ | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (MMRE) | PRED (.25) % |
|---|---|---|---|
| โคโคโม่ | 10 | 1.96 | 0 |
| ฟังก์ชันพอยต์ | 10 | 1.92 | 0 |
| ยูสเคสพอยต์ | 10 | 2.14 | 20 |
| วิธีการตัดสินของผู้เชี่ยวชาญ | 50 | 0.59 | 22 |

การทดสอบการแจกแจงปกติของข้อมูลสามารถทำได้โดยใช้การทดสอบ Shapiro-Wilk และ/หรือ Kolmogorov-Sminov โดยจะปฏิเสธ $H_0$ หากค่า p-value มีค่าน้อยกว่า 0.05

**การวิเคราะห์ความแปรปรวน (ANOVA) ของการทดสอบสมมติฐานที่ 1**

หากผลการทดสอบการแจงแจกปกติพบว่าข้อมูลมีการแจกแจงแบบปกติแล้ว ผู้วิจัยจะใช้วิธีการวิเคราะห์ความแปรปรวนเป็นสถิติในการทดสอบโดยมีวิธีคำนวณดังตารางที่ 3.10 โดยในชุดข้อมูลตัวอย่างนั้นผู้วิจัยจะสมมติให้การแจกแจงของข้อมูลเป็นแบบปกติ

ตารางที่ 42 สรุปการวิเคราะห์ความแปรปรวน

| แหล่งที่มา | **ผลรวมของกำลังสอง** (Sum of Square) | **องศาอิสระ** (Degree of Freedom) | **ค่าประมาณความแปรปรวน** (Variance Estimate) | **อัตราส่วนเอฟ** (F ratio) |
|---|---|---|---|---|
| **ระหว่างกลุ่ม** | $SS_B$ | K-1 | $\mathrm{MS}_B = \frac{SS_B}{K-1}$ | $\frac{MS_B}{MS_w}$ |
| **ภายในกลุ่ม** | $SS_W$ | N-K | $\mathrm{MS}_W = \frac{SS_W}{N-K}$ | |
| ทั้งหมด | $SS_T = SS_B + SS_W$ | N-1 | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$SS_B$ คือ ผลรวมของกำลังสองของค่าเบี่ยงเบนระหว่างกลุ่ม

$SS_W$ คือ ผลรวมของกำลังสองของค่าเบี่ยงเบนภายในกลุ่ม

$SS_T$ คือ ผลรวมของกำลังสองของค่าเบี่ยงเบนทั้งระหว่างกลุ่มและภายในกลุ่ม

K คือ จำนวนกลุ่ม

N คือ จำนวนข้อมูลในกลุ่ม

$MS_B$ คือ ความแปรปรวนระหว่างกลุ่ม

$MS_W$ คือ ความแปรปรวนภายในกลุ่ม

**ตัวอย่างการทดสอบสมมติฐานของข้อมูลสมมติในสมมติฐานที่ 1 แสดงในภาคผนวก จ**

**ข) การทดสอบสมมติฐานที่ 2**

**ข) การทดสอบสมมติฐานที่ 2**

**<u>คำถามวิจัย</u>**

ค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญจะช่วยให้การประเมินแม่นยำขึ้นหรือไม่ **?**

**<u>สมมติฐาน</u>**

**สมมติฐานที่ 2.1**

$H_0$ **:** ผลลัพธ์ที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี **ทำให้ผู้เชี่ยวชาญเปลี่ยนแปลงค่าประเมินไม่เกินครึ่งหนึ่งของจำนวนการทดลองทั้งหมด** จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินของผู้เชี่ยวชาญ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$H_1$ **:** ผลลัพธ์ที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี **ทำให้ผู้เชี่ยวชาญเปลี่ยนแปลงค่าประเมินมากกว่าครึ่งหนึ่ง** จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินใจของผู้เชี่ยวชาญ

**สมมติฐานที่ 2.2** (กรณีผู้เชี่ยวชาญเปลี่ยนแปลงผลลัพธ์จากการประเมินเมื่อได้เห็นค่าประกอบจากการประเมินด้วยวิธีการตัดสินใจของผู้เชี่ยวชาญ)

$H_0$ : หากผู้เชี่ยวชาญได้เห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำไม่แตกต่าง** จากค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

$H_1$ : หากผู้เชี่ยวชาญได้เห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำแตกต่าง** จากค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

**<u>การทดสอบสมมติฐาน</u>**

ในการทดสอบสมมติฐานที่ 2.1 ผู้วิจัยจะทดสอบโดยดูผลลัพธ์จากการประเมินของผู้เชี่ยวชาญ โดยใช้ z-test ในการทดสอบสัดส่วนของจำนวนครั้งที่มีการเปลี่ยนแปลงค่าประเมินต่อจำนวนครั้งที่มีการประเมินทั้งหมด หากค่าซี น้อยกว่าหรือเท่ากับ 0.5 จะยอมรับ $H_0$

ในการทดสอบสมมติฐานที่ 2.2 หากการแจกแจงของข้อมูลเป็นการแจกแจงแบบปกติผู้วิจัยจะใช้การทดสอบ paired t-test แต่หากการแจกแจงของข้อมูลไม่เป็นการแจกแจงแบบปกติ ผู้วิจัยจะใช้การทดสอบ 1 sample sign ในการทดสอบสมมติฐาน

**ตัวอย่างการทดสอบสมมติฐานของข้อมูลสมมติในสมมติฐานที่ 2.2 แสดงในภาคผนวก ช**

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# บทที่ 4 ผลการวิเคราะห์ข้อมูล

ผู้วิจัยได้หาโครงการซอฟต์แวร์จำนวน 10 โครงการ เป็นโครงการซอฟต์แวร์ที่เกิดขึ้นจริง พัฒนาด้วยภาษาการเขียนโปรแกรมที่ได้รับความนิยมในปัจจุบัน ผู้วิจัยเก็บรวบรวมข้อมูลทั้งความต้องการด้านหน้าที่และไม่ใช่ด้านหน้าที่ ค่าความพยายามในการพัฒนา จำนวนบรรทัด ประเภทโครงการซอฟต์แวร์

ในการทดลองนำร่องผู้วิจัยดำเนินการพิมพ์เอกสารความต้องการซอฟต์แวร์ด้านหน้าที่และไม่ใช่ด้านหน้าที่ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ทำการทดลองผู้ประเมินที่เป็นเพื่อนของผู้วิจัยเองจำนวน 2 ท่าน ผู้วิจัยพบว่าใช้ปริมาณกระดาษจำนวนมากและการแจกเอกสารดังกล่าวใช้ระยะเวลาในการดำเนินการค่อนข้างมาก อีกทั้งเพิ่มความสับสนในการทำงานของผู้วิจัยเอง ผู้วิจัยจึงไม่พิมพ์แจกเอกสารความต้องการซอฟต์แวร์ด้านหน้าที่และไม่ใช่ด้านหน้าที่ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ แต่ดำเนินการจัดทำไฟล์นำเสนอที่ระบุข้อมูลดังกล่าวในการนำเสนอแก่ผู้ประเมิน ในวันทดลองจริง นอกจากนี้การทดลองนำร่องชี้ให้ผู้วิจัยเห็นว่าภาระในการดำเนินงานของผู้วิจัยมีปริมาณมาก ในการทดลองจริงผู้วิจัยจึงหาผู้ช่วยผู้วิจัยมาช่วยในการดำเนินการทดลอง

ผู้วิจัยได้ติดต่อผู้เชี่ยวชาญ 5 คนผ่านเฟสบุ๊คและผ่านอาจารย์ที่ปรึกษา มาเป็นผู้ประเมิน มีประสบการณ์ในการพัฒนาซอฟต์แวร์หรือทำงานในธุรกิจที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์มากกว่า 10 ปี ผ่านโครงการซอฟต์แวร์ 5-40 โครงการ มีตำแหน่งงานส่วนใหญ่อยู่ในตำแหน่ง Project Manager มีความคุ้นเคยกับการพัฒนาซอฟต์แวร์ด้วยภาษาการเขียนโปรแกรมที่ได้รับความนิยมในปัจจุบัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในวันทำการทดลองผู้ประเมินทั้ง 5 คน ประเมินโครงการซอฟต์แวร์ในห้องเดียวกัน โดยผู้ประเมินทั้ง 5 คนจะไม่พูดคุยกันในระหว่างการประเมิน ผู้วิจัยได้อธิบายลำดับขั้นตอนของการทดลอง ผลตอบแทนที่จะได้รับภายหลังการประเมิน ผู้ประเมินที่ประเมินแม่นยำที่สุด 3 อันดับแรกจะได้รับรางวัลที่มีมูลค่าตามอันดับ 3 รางวัล ผู้ประเมินอันดับที่ 4 และ 5 จะได้รับผลตอบแทนในมูลค่าเท่ากัน

การทดลองเริ่มจากผู้ประเมินจะกรอกแบบสอบถามที่ 1 ภายหลังจากผู้ประเมินกรอกแบบสอบถามที่ 1 เสร็จแล้ว ผู้วิจัยได้อธิบายความต้องการด้านหน้าที่และความต้องการไม่ใช่ด้านหน้าที่ของโครงการซอฟต์แวร์ ที่ 1 การนำเสนอเป็นการนำเสนอหน้าห้องผ่านสไลด์การนำเสนอ ผู้ประเมินทั้ง 5 สามารถถามคำถามผู้วิจัยได้ตลอดเวลา แต่ไม่สามารถพูดคุยหรือปรึกษากันระหว่างผู้ประเมินได้ เมื่อผู้ประเมิน ทำการประเมินจนเสร็จ ผู้วิจัยจะเก็บผลประเมินและให้ผู้ประเมินกรอก

แบบสอบถามที่ 2 พร้อมทั้งแสดงค่าประเมินจากตัวแบบขั้นตอนวิธีทั้ง 3 ตัวแบบ ผู้ประเมินจะประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์อีกครั้งหนึ่ง ภายหลังจากเห็นค่าประเมินด้วยตัวแบบขั้นตอนวิธี หลังจากนั้น ผู้วิจัยจะเก็บค่าประเมินและนำเสนอโครงการซอฟต์แวร์ลำดับถัดไป เพื่อให้ผู้ประเมินทำการประเมินและจะทำตามขั้นตอนแบบนี้ไปเรื่อยๆ จนกระทั่งครบ 10 โครงการ ในระหว่างดำเนินการทดลอง ผู้วิจัยจะมีผู้ช่วยผู้วิจัยประเมินค่าความคลาดเคลื่อนสัมพัทธ์และเรียงลำดับผู้ประเมินตามความแม่นยำ ภายหลังจากเสร็จสิ้นการทดลองผู้วิจัยจะมอบรางวัลตามลำดับความแม่นยำ เป็นอันเสร็จสิ้นการทดลอง การดำเนินการวิเคราะห์ข้อมูลจากการทดลองข้างต้นประกอบด้วยหัวข้อต่อไปนี้

4.1 รายละเอียดโครงการซอฟต์แวร์ทั้ง 10 โครงการ

4.2 รายละเอียดผู้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จำนวน 5 คน

4.3 การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัว แบบขั้นตอนวิธีทั้งสามวิธี คือ ฟังก์ชันพอยต์ (Function Point) โคโคโม่ (COCOMO) ยูสเคสพอยต์ (Use Case Point)

4.4 ความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากตัวแบบวิธีทั้งสาม

4.5 การประเมินด้วยวิธีการตัดสินของผู้ประเมิน

4.6 ความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากวิธีการตัดสินของผู้ประเมิน

4.7 การทดสอบสมมติฐาน

4.8 สรุปการวิเคราะห์ข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**4.1 รายละเอียดโครงการซอฟต์แวร์ทั้ง 10 โครงการ**

ผู้วิจัยได้ติดต่อนักพัฒนาซอฟต์แวร์และบริษัทที่ประกอบธุรกิจที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ เพื่อขอข้อมูลโครงการซอฟต์แวร์จำนวน 10 โครงการ โครงการซอฟต์แวร์ทั้ง 10 โครงการนี้ เป็นโครงการซอฟต์แวร์ที่ได้มีการพัฒนาขึ้นจริง มีการนับจำนวนบรรทัดที่ได้พัฒนาด้วยคำสั่งทางลินิกส์ ผู้วิจัยทราบค่าความพยายามในการพัฒนาซอฟต์แวร์และความต้องการด้านหน้าที่และไม่ใช่ด้านหน้าที่ของโครงการซอฟต์แวร์ทั้ง 10 โครงการผ่านการสัมภาษณ์ รวมถึงการตรวจสอบความสอดคล้องของค่าความพยายามในการพัฒนาซอฟต์แวร์ในระบบควบคุมเวอร์ชั่น (Version Control) ซึ่งซอฟต์แวร์ทั้ง 10 โครงการนี้ มีค่าความพยายามในการพัฒนาซอฟต์แวร์ตั้งแต่ 2.5 ถึง 14 คน-

เดือน เป็นซอฟต์แวร์ที่ได้มีการพัฒนาขึ้นตั้งแต่ปี พ.ศ. 2557 เป็นต้นมา มีรายละเอียดซอฟต์แวร์ดังตารางที่ 43

ตารางที่ 43 ข้อมูลโครงการซอฟต์แวร์ทั้ง 10 โครงการ ประกอบไปด้วย จำนวนบรรทัด, ประเภทโครงการซอฟต์แวร์ จำนวนความต้องการซอฟต์แวร์ด้านหน้าที่และค่าความพยายามที่ใช้ในการพัฒนาซอฟต์แวร์จริง

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | จำนวนบรรทัดของซอฟต์แวร์ | ประเภทของโครงการซอฟต์แวร์ | จำนวนความต้องการซอฟต์แวร์ด้านหน้าที่ | จำนวนความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่ | ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง (คน-เดือน) |
|---|---|---|---|---|---|---|
| 1 | HHTP | 8371 | Cross platform mobile application | 13 | 3 | 2.5 |
| 2 | PTT | 11131 | Android Application | 8 | 0 | 7.5 |
| 3 | KAPP | 9258 | Cross platform mobile application | 11 | 1 | 8.5 |
| 4 | EATD | 6532 | Android Application | 10 | 3 | 5 |
| 5 | ST | 35974 | Web application | 29 | 0 | 14 |
| 6 | CINOUT | 1596 | Cross platform mobile application | 7 | 2 | 3 |
| 7 | MINSH | 10115 | Web application | 11 | 2 | 3 |
| 8 | LMS | 4210 | Cross platform mobile application | 14 | 2 | 3 |
| 9 | LEAP | 3409 | Web Application | 10 | 2 | 5.5 |
| 10 | TM | 2440 | Web Application | 12 | 4 | 4 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**4.2 รายละเอียดผู้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จำนวน 5 คน**

ผู้วิจัยได้หาผู้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ผ่านการประกาศบนเฟสบุ๊ค (Facebook) ผ่านคณาจารย์ในภาควิชารวมถึงผ่านอาจารย์ที่ปรึกษา จนกระทั่งได้ผู้ประเมินรวมทั้งสิ้น 5 คน เป็นผู้ชาย 4 คน และผู้หญิง 1 คน มีประสบการณ์ในการทำงานที่เกี่ยวข้องกับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ วุฒิการศึกษาของแต่คนอยู่ในระดับปริญญาโท มีส่วนร่วมในโครงการซอฟต์แวร์ตั้งแต่ 5 โครงการ ไปจนถึง 40 โครงการ มีถนัดในภาษาการเขียนโปรแกรมที่ทันสมัยและเป็นที่นิยม เช่น C#, Java ตำแหน่งงานปัจจุบันส่วนใหญ่อยู่ในตำแหน่งระดับ Project Manager มีผู้ประเมินเพียง 1 ท่านอยู่ในตำแหน่ง System Analyst ผู้ประเมินส่วนใหญ่รู้จักการประเมินด้วยตัวแบบขั้นตอนวิธีมาก่อน ทั้ง โคโคโม่ ฟังก์ชันพอยต์ และยูสเคสพอยต์ ข้อมูลคุณสมบัติผู้ประเมินทั้ง 5 ท่านมีดังต่อไปนี้

| **ผู้ประเมินคนที่ 1** | |
|---|---|
| เพศ | ชาย |
| อายุ | 39 ปี |
| วุฒิการศึกษาสูงสุด | ปริญญาโท |
| ตำแหน่งและจำนวนปีประสบการณ์ | Developer 5 ปี, System Analyst 5 ปี, Project Manager 7 ปี |
| ตำแหน่งงานล่าสุด | Project Manager |
| จำนวนโครงการที่พัฒนาหรือมีส่วนร่วม | 40 โครงการ |
| ภาษาการเขียนโปรแกรมที่ถนัดสามอันดับแรก | C#, Java, Javascript |
| รู้จักโคโคโม่ | ใช่ |
| รู้จักฟังก์ชันพอยต์ | ใช่ |
| รู้จักยูสเคสพอยต์ | ใช่ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| **ผู้ประเมินคนที่ 2** | |
|---|---|
| เพศ | ชาย |
| อายุ | 38 ปี |
| วุฒิการศึกษาสูงสุด | ปริญญาโท |
| ตำแหน่งและจำนวนปีประสบการณ์ | Programmer 2 ปี, Software Engineer 5 ปี, System Analyst 5 ปี, Assistant Manager 1 ปี |
| ตำแหน่งงานล่าสุด | Assistant Manager |
| จำนวนโครงการที่พัฒนาหรือมีส่วนร่วม | 5 โครงการ |
| ภาษาการเขียนโปรแกรมที่ถนัดสามอันดับแรก | VB.Net, C#, VBA |
| รู้จักโคโคโม่ | ใช่ |
| รู้จักฟังก์ชันพอยต์ | ใช่ |
| รู้จักยูสเคสพอยต์ | ใช่ |

| **ผู้ประเมินคนที่ 3** | |
|---|---|
| เพศ | หญิง |
| อายุ | 42 ปี |
| วุฒิการศึกษาสูงสุด | ปริญญาโท |
| ตำแหน่งและจำนวนปีประสบการณ์ | Software Manager 2 ปี, Project Manager 9 ปี |
| ตำแหน่งงานล่าสุด | Project Manager |
| จำนวนโครงการที่พัฒนาหรือมีส่วนร่วม | 10 โครงการ |
| ภาษาการเขียนโปรแกรมที่ถนัดสามอันดับแรก | Java, Javascript, C# |
| รู้จักโคโคโม่ | ใช่ |
| รู้จักฟังก์ชันพอยต์ | ใช่ |
| รู้จักยูสเคสพอยต์ | ไม่แน่ใจ |

| **ผู้ประเมินคนที่ 4** | |
|---|---|
| เพศ | ชาย |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| | |
|---|---|
| อายุ | 40 ปี |
| วุฒิการศึกษาสูงสุด | ปริญญาโท |
| ตำแหน่งและจำนวนปีประสบการณ์ | Programmer 4 ปี, System Analyst 4 ปี, Project Manager 8 ปี |
| ตำแหน่งงานล่าสุด | Project Manager |
| จำนวนโครงการที่พัฒนาหรือมีส่วนร่วม | 20 โครงการ |
| ภาษาการเขียนโปรแกรมที่ถนัดสามอันดับแรก | PHP, Java, Javascript |
| รู้จักโคโคโม่ | ใช่ |
| รู้จักฟังก์ชันพอยต์ | ใช่ |
| รู้จักยูสเคสพอยต์ | ใช่ |

| **ผู้ประเมินคนที่ 5** | |
|---|---|
| เพศ | ชาย |
| อายุ | 33 ปี |
| วุฒิการศึกษาสูงสุด | ปริญญาตรี |
| ตำแหน่งและจำนวนปีประสบการณ์ | Application Development 2 ปี, Analyst 1 ปี, IT Operation 5 ปี, System Analyst 2 ปี |
| ตำแหน่งงานล่าสุด | System Analyst |
| จำนวนโครงการที่พัฒนาหรือมีส่วนร่วม | มากกว่า 10 โครงการ |
| ภาษาการเขียนโปรแกรมที่ถนัดสามอันดับแรก | C#, Javascript, C |
| รู้จักโคโคโม่ | ไม่ |
| รู้จักฟังก์ชันพอยต์ | ไม่ |
| รู้จักยูสเคสพอยต์ | ไม่ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**4.3 การประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธีทั้งสามวิธี คือ ฟังก์ชันพอยต์ (Function Point) โคโคโม่ (COCOMO) ยูสเคสพอยต์ (Use Case Point)**

ผู้วิจัยต้องประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีทั้ง 3 ตัวแบบคือ ฟังก์ชันพอยต์ โคโคโม่ และยูสเคสพอยต์ เพื่อใช้เปรียบเทียบความแม่นยำในการประเมินกับวิธีการตัดสินของผู้ประเมิน อีกทั้งยังสามารถนำค่าประเมินไปประกอบการตัดสินใจของผู้ประเมินว่าจะเปลี่ยนแปลงค่าประเมินเมื่อเห็นค่าประเมินของตัวแบบขั้นตอนวิธีทั้งสามหรือไม่ การประเมิน

ด้วยวิธีตัวแบบขั้นตอนวิธีทั้งสามตัวแบบ ประเมินโดยผู้วิจัยและได้มีการตรวจสอบความถูกต้องโดยอาจารย์ที่ปรึกษาและผศ.ดร.วัฒนา วิริยสิทธาวัฒน์

4.3.1 ฟังก์ชันพอยต์

ผู้วิจัยได้พิจารณาองค์ประกอบหลักทั้ง 5 คือ EI, EO, EQ, ILF's, EIF's ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ สำหรับการวิเคราะห์ความซับซ้อนของฟังก์ชันพอยต์ โดยค่าขององค์ประกอบทั้ง 5 แสดงในตารางที่ 44

ตารางที่ 44 การหาค่าองค์ประกอบทั้ง 5 ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการ/ค่าองค์ประกอบ | EI | EO | EQ | ILF | EIF |
|---|---|---|---|---|---|---|
| 1 | HHTP | 10 | 4 | 0 | 3 | 1 |
| 2 | PTT | 3 | 5 | 0 | 1 | 1 |
| 3 | KAPP | 5 | 14 | 2 | 2 | 4 |
| 4 | EATD | 6 | 9 | 0 | 3 | 1 |
| 5 | ST | 7 | 10 | 4 | 4 | 4 |
| 6 | CINOUT | 3 | 4 | 0 | 0 | 2 |
| 7 | MINSH | 13 | 10 | 0 | 5 | 0 |
| 8 | LMS | 24 | 11 | 0 | 4 | 1 |
| 9 | LEAP | 11 | 6 | 1 | 2 | 0 |
| 10 | TM | 9 | 4 | 2 | 0 | 0 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยได้หาค่าความซับซ้อนจากผลการประเมินองค์ประกอบทั้ง 5 ของโครงการซอฟต์แวร์ทั้ง 10 โครงการดังตารางที่ 45

ตารางที่ 45 ค่าความซับซ้อนขององค์ประกอบทั้ง 5, ค่า DET, FTR/RET ของ 10 โครงการซอฟต์แวร์ (H คือ High, A คือ Average และ L คือ Low)

| ลำดับที่ | ชื่อโครงการ/ค่าความซับซ้อน | DET | FTR/RET | EI | EO | EQ | ILF | EIF |
|---|---|---|---|---|---|---|---|---|
| 1 | HHTP | 17 | 2 | H | A | A | L | L |
| 2 | PTT | 7 | 0 | L | L | L | L | L |
| 3 | KAPP | 13 | 4 | H | H | H | L | L |

| ลำดับที่ | ชื่อโครงการ/ค่าความซับซ้อน | DET | FTR/RET | EI | EO | EQ | ILF | EIF |
|---|---|---|---|---|---|---|---|---|
| 4 | EATD | 8 | 3 | H | A | A | L | L |
| 5 | ST | 35 | 8 | H | H | H | H | H |
| 6 | CINOUT | 7 | 2 | A | A | A | L | L |
| 7 | MINSH | 21 | 4 | H | H | H | A | A |
| 8 | LMS | 25 | 5 | H | H | H | A | A |
| 9 | LEAP | 16 | 3 | H | A | A | L | L |
| 10 | TM | 12 | 3 | H | A | A | L | L |

หลังจากที่ได้ค่าความซับซ้อนขององค์ประกอบทั้ง 5 แล้ว ผู้วิจัยจะคิดค่าฟังก์ชันพอยต์ของแต่ละองค์ประกอบโดยนำค่าของแต่ละองค์ประกอบ คูณด้วยค่าตัวคูณปัจจัย (Factor Multiplier) ของค่าความซับซ้อนขององค์ประกอบทั้ง 5 ซึ่งค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์จะเท่ากับผลรวมของค่าฟังก์ชันพอยต์ที่ได้จากแต่ละองค์ประกอบ

ผลลัพธ์จากการหาค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงในตารางที่ 46 ถึง 55 และสรุปค่าฟังก์ชันพอยต์ที่ยังไม่ปรับค่าในตารางที่ 56

ตารางที่ 46 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ HHTP

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 10 | High = 6 | 60 |
| 2 | EO | 4 | Average = 5 | 20 |
| 3 | EQ | 0 | Low = 3 | 0 |
| 4 | ILF | 3 | Low = 7 | 21 |
| 5 | EIF | 1 | Low =5 | 5 |
| | ผลรวม | | | 106 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 47 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ PTT

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 3 | Low = 3 | 9 |
| 2 | EO | 5 | Low = 4 | 20 |
| 3 | EQ | 0 | Low = 3 | 0 |
| 4 | ILF | 1 | Low = 7 | 7 |
| 5 | EIF | 1 | Low = 5 | 5 |
| | ผลรวม | | | 41 |

ตารางที่ 48 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ KAPP

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 5 | High = 6 | 30 |
| 2 | EO | 14 | High = 7 | 98 |
| 3 | EQ | 2 | High = 6 | 12 |
| 4 | ILF | 2 | Low = 7 | 14 |
| 5 | EIF | 4 | Low = 5 | 20 |
| | ผลรวม | | | 174 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 49 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ EATD

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 6 | High = 6 | 36 |
| 2 | EO | 9 | Average = 5 | 45 |
| 3 | EQ | 0 | Average = 4 | 0 |
| 4 | ILF | 3 | Low = 7 | 21 |

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 5 | EIF | 1 | Low = 5 | 5 |
| | ผลรวม | | | 107 |

ตารางที่ 50 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ ST

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 27 | High = 6 | 162 |
| 2 | EO | 10 | High =7 | 70 |
| 3 | EQ | 4 | High = 6 | 24 |
| 4 | ILF | 4 | High = 15 | 60 |
| 5 | EIF | 4 | High = 10 | 40 |
| | ผลรวม | | | 350 |

ตารางที่ 51 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ CINOUT

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 3 | Average = 4 | 12 |
| 2 | EO | 4 | Average = 5 | 20 |
| 3 | EQ | 0 | Average = 4 | 0 |
| 4 | ILF | 0 | Low = 10 | 0 |
| 5 | EIF | 2 | Low = 5 | 10 |
| | ผลรวม | | | 42 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 52 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ MINSH

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 13 | High = 6 | 78 |
| 2 | EO | 10 | High = 7 | 70 |
| 3 | EQ | 0 | High = 6 | 0 |
| 4 | ILF | 5 | Average = 10 | 50 |
| 5 | EIF | 0 | Average = 7 | 0 |
| | ผลรวม | | | 198 |

ตารางที่ 53 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ LMS

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 24 | High = 6 | 144 |
| 2 | EO | 11 | High =7 | 77 |
| 3 | EQ | 0 | High =6 | 0 |
| 4 | ILF | 4 | Average = 10 | 40 |
| 5 | EIF | 1 | Average = 7 | 7 |
| | ผลรวม | | | 268 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 54 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ LEAP

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 11 | High = 6 | 66 |
| 2 | EO | 6 | Average = 5 | 30 |
| 3 | EQ | 1 | Average = 5 | 4 |
| 4 | ILF | 2 | Low = 7 | 14 |
| 5 | EIF | 0 | Low = 5 | 0 |
| | ผลรวม | | | 114 |

ตารางที่ 55 ค่าฟังก์ชันพอยต์ของโครงการซอฟต์แวร์ TM

| ลำดับที่ | องค์ประกอบทั้ง 5 | จำนวนองค์ประกอบทั้ง 5 | ค่าความซับซ้อน | ค่าฟังก์ชันพอยต์ |
|---|---|---|---|---|
| 1 | EI | 9 | High = 6 | 54 |
| 2 | EO | 4 | Average = 5 | 20 |
| 3 | EQ | 2 | Average = 4 | 8 |
| 4 | ILF | 0 | Low = 7 | 0 |
| 5 | EIF | 0 | Low = 5 | 0 |
| | ผลรวม | | | 82 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 56 สรุปค่าฟังก์ชันพอยต์ไม่ปรับค่าของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าฟังก์ชันพอยต์ไม่ปรับค่า (Unadjusted Function Point) |
|---|---|---|
| 1 | HHTP | 106 |
| 2 | PTT | 41 |
| 3 | KAPP | 174 |
| 4 | EATD | 107 |
| 5 | ST | 350 |

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าฟังก์ชันพอยต์ไม่ปรับค่า (Unadjusted Function Point) |
|---|---|---|
| 6 | CINOUT | 42 |
| 7 | MINSH | 198 |
| 8 | LMS | 268 |
| 9 | LEAP | 114 |
| 10 | TM | 82 |

ผลลัพธ์จากการหาฟังก์ชันพอยต์ข้างต้น จะเป็นการค่าฟังชันพอยต์ที่ยังไม่ปรับค่า ซึ่งผู้วิจัยต้องการหาค่าฟังก์ชันพอยต์ปรับค่า (Adjusted Function Point) เพื่อให้ผลลัพธ์ที่ได้สอดคล้องกับซอฟต์แวร์ที่ต้องการประเมินมากยิ่งขึ้น

การหาค่าฟังก์ชันพอยต์แบบปรับค่าสามารถคำนวณได้จากสูตรที่ได้กล่าวไว้ในบทที่ 2 ดังนี้

ฟังก์ชันพอยต์ปรับค่า (Adjusted Function Point) = UFP X TCF

โดย TCF หรือ Technical Complexity Factors คือค่าปัจจัยความซับซ้อนเชิงเทคนิค ผู้วิจัยประเมินค่าความซับซ้อนเชิงเทคนิคทั้ง 14 ปัจจัยของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ดังตารางที่ 57-70

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 57 ค่าความซับซ้อนเชิงเทคนิค การสื่อสารข้อมูล (Data Communication) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Data Communication |
|---|---|---|
| 1 | HHTP | 4 |
| 2 | PTT | 3 |
| 3 | KAPP | 5 |
| 4 | EATD | 5 |
| 5 | ST | 5 |
| 6 | CINOUT | 5 |
| 7 | MINSH | 4 |
| 8 | LMS | 5 |
| 9 | LEAP | 4 |
| 10 | TM | 5 |

ตารางที่ 58 ค่าความซับซ้อนเชิงเทคนิค การประมวลผลข้อมูลแบบกระจาย (Distributed Data Processing) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Data Communication |
|---|---|---|
| 1 | HHTP | 4 |
| 2 | PTT | 1 |
| 3 | KAPP | 5 |
| 4 | EATD | 5 |
| 5 | ST | 5 |
| 6 | CINOUT | 4 |
| 7 | MINSH | 4 |
| 8 | LMS | 5 |
| 9 | LEAP | 4 |
| 10 | TM | 4 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 59 ค่าความซับซ้อนเชิงเทคนิค สมรรถนะ (Performance) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Performance |
|---|---|---|
| 1 | HHTP | 1 |
| 2 | PTT | 5 |
| 3 | KAPP | 3 |
| 4 | EATD | 1 |
| 5 | ST | 5 |
| 6 | CINOUT | 1 |
| 7 | MINSH | 2 |
| 8 | LMS | 3 |
| 9 | LEAP | 2 |
| 10 | TM | 1 |

ตารางที่ 60 ค่าความซับซ้อนเชิงเทคนิค ความต้องการใช้งานหนักด้านฮาร์ดแวร์ (Havily Used Configuration) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Heavily Used Configuration |
|---|---|---|
| 1 | HHTP | 3 |
| 2 | PTT | 3 |
| 3 | KAPP | 5 |
| 4 | EATD | 3 |
| 5 | ST | 4 |
| 6 | CINOUT | 3 |
| 7 | MINSH | 2 |
| 8 | LMS | 5 |
| 9 | LEAP | 2 |
| 10 | TM | 2 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 61 ค่าความซับซ้อนเชิงเทคนิค อัตรารายการเปลี่ยนแปลง (Transaction Rate) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Transaction rate |
|---|---|---|
| 1 | HHTP | 0 |
| 2 | PTT | 0 |
| 3 | KAPP | 0 |
| 4 | EATD | 0 |
| 5 | ST | 1 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 0 |
| 8 | LMS | 0 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

ตารางที่ 62 ค่าความซับซ้อนเชิงเทคนิค การบันทึกข้อมูลออนไลน์ (On-line Data Entry) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน On-line Data Entry |
|---|---|---|
| 1 | HHTP | 5 |
| 2 | PTT | 2 |
| 3 | KAPP | 5 |
| 4 | EATD | 4 |
| 5 | ST | 5 |
| 6 | CINOUT | 5 |
| 7 | MINSH | 5 |
| 8 | LMS | 5 |
| 9 | LEAP | 5 |
| 10 | TM | 5 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 63 ค่าความซับซ้อนเชิงเทคนิค ประสิทธิภาพเพื่อผู้ใช้ขั้นปลาย (End-User Efficiency) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน End-User Efficiency |
|---|---|---|
| 1 | HHTP | 2 |
| 2 | PTT | 1 |
| 3 | KAPP | 2 |
| 4 | EATD | 1 |
| 5 | ST | 2 |
| 6 | CINOUT | 2 |
| 7 | MINSH | 2 |
| 8 | LMS | 2 |
| 9 | LEAP | 2 |
| 10 | TM | 2 |

ตารางที่ 64 ค่าความซับซ้อนเชิงเทคนิค การปรับให้เป็นปัจจุบันแบบออนไลน์ (On-line Update) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน On-line Update |
|---|---|---|
| 1 | HHTP | 0 |
| 2 | PTT | 0 |
| 3 | KAPP | 0 |
| 4 | EATD | 0 |
| 5 | ST | 0 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 0 |
| 8 | LMS | 0 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 65 ค่าความซับซ้อนเชิงเทคนิค การประมวลผลที่ซับซ้อน (Complexity Processing) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Complexity Processing |
|---|---|---|
| 1 | HHTP | 2 |
| 2 | PTT | 0 |
| 3 | KAPP | 2 |
| 4 | EATD | 0 |
| 5 | ST | 5 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 2 |
| 8 | LMS | 2 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

ตารางที่ 66 ค่าความซับซ้อนเชิงเทคนิค ความสามารถนำกลับมาใช้ใหม่ (Reusability) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Reusability |
|---|---|---|
| 1 | HHTP | 0 |
| 2 | PTT | 2 |
| 3 | KAPP | 2 |
| 4 | EATD | 0 |
| 5 | ST | 3 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 1 |
| 8 | LMS | 2 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 67 ค่าความซับซ้อนเชิงเทคนิค ความสะดวกในการติดตั้ง (Installation Ease) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางด้านเทคนิคด้าน Installation Ease |
|---|---|---|
| 1 | HHTP | 0 |
| 2 | PTT | 0 |
| 3 | KAPP | 1 |
| 4 | EATD | 0 |
| 5 | ST | 1 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 0 |
| 8 | LMS | 1 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

ตารางที่ 68 ค่าความซับซ้อนเชิงเทคนิค ความสะดวกในการปฏิบัติงาน (Operational Ease) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Operational Ease |
|---|---|---|
| 1 | HHTP | 0 |
| 2 | PTT | 0 |
| 3 | KAPP | 0 |
| 4 | EATD | 0 |
| 5 | ST | 2 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 0 |
| 8 | LMS | 0 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 69 ค่าความซับซ้อนเชิงเทคนิค หลายสถานที่ (Multiple Sites) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Multiple Sites |
|---|---|---|
| 1 | HHTP | 0 |
| 2 | PTT | 0 |
| 3 | KAPP | 0 |
| 4 | EATD | 0 |
| 5 | ST | 1 |
| 6 | CINOUT | 0 |
| 7 | MINSH | 0 |
| 8 | LMS | 0 |
| 9 | LEAP | 0 |
| 10 | TM | 0 |

ตารางที่ 70 ค่าความซับซ้อนเชิงเทคนิค คำอธิบายระดับขั้นปัจจัยอำนวยความสะดวกในการเปลี่ยนแปลง (Facilitate Change) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าปัจจัยทางเทคนิคด้าน Facilitate Change |
|---|---|---|
| 1 | HHTP | 3 |
| 2 | PTT | 3 |
| 3 | KAPP | 3 |
| 4 | EATD | 3 |
| 5 | ST | 3 |
| 6 | CINOUT | 3 |
| 7 | MINSH | 3 |
| 8 | LMS | 3 |
| 9 | LEAP | 3 |
| 10 | TM | 3 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การคำนวณค่าปัจจัยความซับซ้อนเชิงเทคนิค (Technical Complexity Factors หรือ TCF) และค่าฟังก์ชันพอยต์ปรับค่าแล้ว (Adjusted Function Point) โดยมีสูตรดังนี้

สูตรคำนวณค่าปัจจัยความซับซ้อนเชิงเทคนิค

TCF = 0.65 + 0.01 x (ผลรวมของค่าความซับซ้อนจากปัจจัยทั้ง 14 ปัจจัย)

สูตรคำนวณค่าฟังก์ชันพอยต์ปรับค่าแล้ว

ฟังก์ชันพอยต์ปรับค่า (Adjusted Function Point) = UFP x TCF

ค่าปัจจัยความซับซ้อนเชิงเทคนิคและค่าฟังก์ชันพอยต์ปรับค่าแล้วของโครงการซอฟต์แวร์ทั้ง 10 โครงการ มีค่าดังนี้

1. โครงการซอฟต์แวร์ HHTP

ปัจจัยความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x (4+4+1+3+0+5+2+0+2+0+0+0+0+3)

= 0.89

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 106 x 0.89 = 94.34

2. โครงการซอฟต์แวร์ PTT

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x (3+1+5+3+0+2+1+0+0+2+0+0+0+3)

= 0.85

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 41 x 0.85 = 34.85

3. โครงการซอฟต์แวร์ KAPP

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x (5+5+3+5+0+5+2+0+2+2+1+0+0+3)

= 0.98

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 174 x 0.98 = 170.52

4. โครงการซอฟต์แวร์ EATD

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x

(5+5+1+3+0+4+1+0+0+0+0+0+0+3)

= 0.87

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 107 x 0.87 = 93.09

5. โครงการซอฟต์แวร์ ST

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x

(5+5+5+4+1+5+2+0+5+3+1+2+1+3)

= 1.07

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 350 x 1.07 = 374.5

6. โครงการซอฟต์แวร์ CINOUT

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x

(5+4+1+3+0+5+2+0+0+0+0+0+0+3)

= 0.88

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 42 x 0.88 = 36.96

7. โครงการซอฟต์แวร์ MINSH

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x

(4+4+2+2+0+5+2+0+2+1+0+0+0+3)

= 0.9

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 198 x 0.9 = 178.2

8. โครงการซอฟต์แวร์ LMS

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x
(5+5+3+5+0+5+2+0+2+2+1+0+0+3)

= 0.98

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 268 x 0.98 = 262.64

9. โครงการซอฟต์แวร์ LEAP

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x
(4+4+2+2+0+5+2+0+0+0+0+0+0+3)

= 0.87

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 114 x 0.87 = 99.18

10. โครงการซอฟต์แวร์ TM

ปัจจัยค่าความซับซ้อนเชิงเทคนิค = 0.65 + 0.01 x
(5+4+1+2+0+5+2+0+0+0+0+0+0+3)

= 0.87

ค่าฟังก์ชันพอยต์ปรับค่าแล้ว = 82 x 0.87 = 71.34

ผลของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบฟังก์ชันพอยต์แสดงในตารางที่ 71

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 71 ค่าฟังชันพอยต์ที่ยังไม่ปรับค่าและฟังก์ชันพอยต์ที่ปรับค่าแล้วของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | ชื่อโครงการซอฟต์แวร์ | ค่าฟังก์ชันพอยต์ที่ยังไม่ปรับค่า (Unadjusted Function Point) | ค่าฟังก์ชันพอยต์ที่ปรับค่าแล้ว (Adjusted Function Point) |
|---|---|---|---|
| 1 | HHTP | 106 | 94.34 |
| 2 | PTT | 41 | 34.85 |
| 3 | KAPP | 174 | 170.52 |
| 4 | EATD | 107 | 93.09 |
| 5 | ST | 350 | 374.5 |
| 6 | CINOUT | 42 | 36.96 |
| 7 | MINSH | 198 | 178.2 |
| 8 | LMS | 268 | 262.64 |
| 9 | LEAP | 114 | 99.18 |
| 10 | TM | 82 | 71.34 |

เมื่อผู้วิจัยได้ค่าฟังก์ชันพอยต์มาแล้ว ผู้วิจัยจะหาค่าความพยายามในการพัฒนาซอฟต์แวร์จากฟังก์ชันพอยต์ สูตรการหาค่าความพยายามในการพัฒนาซอฟต์แวร์จากฟังก์ชันพอยต์มีหลายวิธีดังตารางที่ 72

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 72 วิธีการหาค่าความพยายามในการพัฒนาซอฟต์แวร์จากค่าฟังก์ชันพอยต์

| ลำดับที่ | งานวิจัย | ค่าความพยายาม (คน-ชั่วโมง) |
|---|---|---|
| 1 | (Albrecht, 1983) | Effort = FP / 0.0595 |
| 2 | (Behrens, 1983) | Effort = 18.3 * FP |
| 3 | (Moser, 1996) | Effort = 6.667 * FP |
| 4 | (Lokan, 2000) | Effort = 21 * FP |

| ลำดับที่ | งานวิจัย | ค่าความพยายาม (คน-ชั่วโมง) |
|---|---|---|
| 5 | (Maxwell, 2000) | Effort = FP * 2.97 |
| 6 | (Jeffery R., 2001) | Effort = 2.2 * FP |
| 7 | (Arnuphaptrairong, 2018) | Effort = FP/1.09 |

งานวิจัยของ (Jeffery R., 2001) เหมาะสมที่สุดสำหรับการประเมินค่าความพยายามภายใต้สภาวะแวดล้อมของคนไทย (อานุภาพไตรรงค์, 2558) ผู้วิจัยจึงใช้วิธีการหาค่าประเมินด้วยฟังก์ชันพอยต์ด้วยการคูณด้วยค่า 2.2 จะได้เป็นค่าประเมินในหน่วยคน-ชั่วโมง ผู้วิจัยได้ปรับค่าหน่วยจาก คน-ชั่วโมง เป็น คน-เดือน โดยการนำค่าที่ได้ในหน่วย คน-เดือน มาหารด้วย 152 ผลลัพธ์แสดงในตารางที่ 73

ตารางที่ 73 ค่าความพยายามในการพัฒนาซอฟต์แวร์ที่ได้จากการประเมินด้วยวิธีฟังก์ชันพอยต์ (ผู้วิจัยปรับหน่วยจาก คน-ชั่วโมง เป็น คน-เดือน แล้ว)

| ลำดับที่ | โครงการซอฟต์แวร์ทั้ง 10 โครงการ | ค่าความพยายามในการพัฒนาซอฟต์แวร์ ที่ได้จากการประเมินด้วยฟังก์ชันพอยต์ (คน-เดือน) |
|---|---|---|
| 1 | HHTP | 1.5 |
| 2 | PTT | 1 |
| 3 | KAPP | 2.5 |
| 4 | EATD | 1.5 |
| 5 | ST | 5.5 |
| 6 | CINOUT | 1 |
| 7 | MINSH | 3 |
| 8 | LMS | 4 |
| 9 | LEAP | 1.5 |
| 10 | TM | 1.5 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.3.2 โคโคโม่

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่นั้น ผู้วิจัยจำเป็นต้องหาขนาดของซอฟต์แวร์เสียก่อน ซึ่งผู้วิจัยจะใช้ค่าประเมินที่ได้ในหน่วยฟังก์ชันพอยต์ มาหาขนาดของซอฟต์แวร์ในหน่วย Kilo Source Line of Code (KLOC) โดยการนำมาคูณกับค่าที่ได้จากตาราง Function Point Languages Table (ข้อมูลตารางแสดงในภาคผนวก ก.) ตามสูตรต่อไปนี้

KLOC = Function Point x ค่าที่ได้จาก Function Point Language Table

การหาขนาดของโครงการซอฟต์แวร์ทั้ง 10 โครงการ แสดงในตารางที่ 74

ตารางที่ 74 ค่าฟังก์ชันพอยต์, ภาษา, ค่า Kilo Source Line of Code ต่อ 1 ฟังก์ชันพอยต์ และขนาดของโครงการซอฟต์แวร์ทั้ง 10 โครงการ (หน่วย Kilo Source Line of Code)

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าฟังก์ชันพอยต์ที่ปรับค่าแล้ว | ภาษาการเขียนโปรแกรม | ค่าจากตาราง Function Point Language Table | จำนวนบรรทัด |
|---|---|---|---|---|---|
| 1 | HHTP | 94.34 | Javascript | 47 | 4,433 |
| 2 | PTT | 34.85 | C# | 54 | 1,881 |
| 3 | KAPP | 170.52 | Javascript | 47 | 8,014 |
| 4 | EATD | 93.09 | Java | 53 | 4,933 |
| 5 | ST | 374.5 | Javascript | 47 | 17,601 |
| 6 | CINOUT | 36.96 | Javascript | 47 | 1,737 |
| 7 | MINSH | 178.2 | Javascript | 47 | 8,375 |
| 8 | LMS | 262.64 | Javascript | 47 | 12,344 |
| 9 | LEAP | 99.18 | Javascript | 47 | 4,661 |
| 10 | TM | 71.34 | Javascript | 47 | 3,352 |

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์โดยใช้โคโคโม่ ต้องพิจารณาปัจจัยทั้ง 22 ปัจจัย แบ่งออกเป็น 4 ด้านดังตารางที่ 75

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 75 หมวดหมู่และปัจจัยในแต่ละหมวดหมู่ของการประเมินค่าความพยายามด้วยวิธีตัวแบบโคโคโม่

| หมวดหมู่ปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ | จำนวน | ปัจจัยในแต่ละหมวดหมู่ |
|---|---|---|
| ปัจจัยด้านผลิตภัณฑ์ | 6 | RCPX, RELY, DATA, CPLX, DOCU, RUSE |
| ปัจจัยด้านแพลตฟอร์ม | 4 | TIME, PDIF, STOR, PVOL |
| ปัจจัยด้านบุคคล | 8 | PERS, PREX, ACAP, PCAP, Personal Continuity, AEXP, PEXP, LEXP |
| ปัจจัยด้านโครงการซอฟต์แวร์ | 4 | FCIL, TOOL, SITE, SCED |

เกณฑ์การให้ค่าปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์มี 5 ระดับคือ Very low (VL), Low (L), Nominal(N), High(H), Very High(VH) ผลการประเมินค่าปัจจัยทั้ง 22 ปัจจัยของโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงดังตารางที่ 76-85

ตารางที่ 76 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ HHTP

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 77 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ PTT

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | H | | |
| | | | | LEXP | N | | |

ตารางที่ 78 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ KAPP

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | H | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 79 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ EATD

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | H | | |
| | | | | LEXP | N | | |

ตารางที่ 80 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ ST

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | VH | TIME | N | PERS | H | FCIL | N |
| RELY | VH | PDIF | N | PREX | H | TOOL | N |
| DATA | VH | STOR | N | ACAP | H | SITE | N |
| CPLX | VH | PVOL | N | PCAP | H | SCED | N |
| DOCU | VH | | | Personal Continuity | H | | |
| RUSE | N | | | AEXP | H | | |
| | | | | PEXP | H | | |
| | | | | LEXP | H | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 81 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ CINOUT

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

ตารางที่ 82 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ MINSH

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 83 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ LMS

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | H | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

ตารางที่ 84 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ LEAP

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 85 การประเมินปัจจัยที่ส่งผลต่อค่าความพยายามในการพัฒนาซอฟต์แวร์ : โครงการ TM

| ปัจจัยด้านผลิตภัณฑ์ | ระดับค่าปัจจัย | ปัจจัยด้านแพลตฟอร์ม | ระดับค่าปัจจัย | ปัจจัยด้านบุคคล | ระดับค่าปัจจัย | ปัจจัยด้านโครงการซอฟต์แวร์ | ระดับค่าปัจจัย |
|---|---|---|---|---|---|---|---|
| RCPX | N | TIME | N | PERS | N | FCIL | N |
| RELY | N | PDIF | N | PREX | N | TOOL | N |
| DATA | N | STOR | N | ACAP | N | SITE | N |
| CPLX | N | PVOL | N | PCAP | N | SCED | N |
| DOCU | N | | | Personal Continuity | N | | |
| RUSE | N | | | AEXP | N | | |
| | | | | PEXP | N | | |
| | | | | LEXP | N | | |

จากบทที่ 2 การหาค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่มีสูตรดังต่อไปนี้

$$PM = (2.94\ (Size)^{E}) \times EM$$

ผู้วิจัยใช้เครื่องมือจากเว็บไซต์ http://csse.usc.edu/tools/cocomoii.php เพื่อให้สะดวกในการประเมินด้วยวิธี COCOMO II โดยค่าประเมินที่ได้แสดงในตารางที่ 86

ตารางที่ 86 ค่าประเมินจากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีโคโคโม่ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (คน-เดือน) |
|---|---|---|
| 1 | HHTP | 15.1 |
| 2 | PTT | 5.4 |
| 3 | KAPP | 25.5 |
| 4 | EATD | 19.8 |
| 5 | ST | 117.6 |

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าความพยายามในการพัฒนาซอฟต์แวร์ (คน-เดือน) |
|---|---|---|
| 6 | CINOUT | 5.4 |
| 7 | MINSH | 30.4 |
| 8 | LMS | 41 |
| 9 | LEAP | 16.0 |
| 10 | TM | 11.1 |

4.3.3 ยูสเคสพอยต์

การประเมินค่าความพยายามด้วยยูสเคสพอยต์ของโครงการซอฟต์แวร์ทั้ง 10 โครงการสามารถดำเนินการได้ตามขั้นตอนในบทที่ 2 ดังนี้

1) การหาค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight)
2) การหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Weights หรือ UUCW)
3) การหาค่ายูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Point)
4) การหาค่าความซับซ้อนทางเทคนิค (Technical Complexity Factor หรือ TCP)
5) การหาค่าความซับซ้อนทางสิ่งแวดล้อม (Environments Factors)
6) การหาค่า ยูสเคสพอยต์ที่ปรับค่าแล้ว (Adjusted Use Case Point หรือ UCP)
7) การคำนวณหาค่าความพยายามในการพัฒนาซอฟต์แวร์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1) การหาค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่า (Unadjusted Actor Weight)

ผู้วิจัยได้ประเมินค่าน้ำหนักแอคเตอร์ที่ยังไม่ได้ปรับค่าของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ดังแสดงในตารางที่ 87-96

ตารางที่ 87 หนักค่าน้ำแอคเตอร์ของโครงการซอฟต์แวร์ HHTP

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่านำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 2 | 3 | 6 |
| Average | 0 | 2 | 0 |
| Simple | 1 | 1 | 1 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | **7** |

ตารางที่ 88 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ PTT

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 1 | 3 | 3 |
| Average | 0 | 2 | 0 |
| Simple | 1 | 1 | 1 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | **4** |

ตารางที่ 89 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ KAPP

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 2 | 3 | 6 |
| Average | 1 | 2 | 2 |
| Simple | 3 | 1 | 3 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | **11** |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 90 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ EATD

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 1 | 3 | 3 |
| Average | 2 | 2 | 4 |
| Simple | 1 | 1 | 1 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | **8** |

ตารางที่ 91 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ ST

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 4 | 3 | 12 |
| Average | 2 | 2 | 4 |
| Simple | 2 | 1 | 2 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | 18 |

ตารางที่ 92 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ CINOUT

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 2 | 3 | 6 |
| Average | 2 | 2 | 4 |
| Simple | 1 | 1 | 1 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | 11 |

ตารางที่ 93 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ MINSH

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 3 | 3 | 9 |
| Average | 0 | 2 | 0 |
| Simple | 0 | 1 | 0 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | 9 |

ตารางที่ 94 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ LMS

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 3 | 3 | 9 |
| Average | 0 | 2 | 0 |
| Simple | 1 | 1 | 1 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | 10 |

ตารางที่ 95 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ LEAP

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 2 | 3 | 6 |
| Average | 0 | 2 | 0 |
| Simple | 0 | 1 | 0 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | 6 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 96 ค่าน้ำหนักแอคเตอร์ของโครงการซอฟต์แวร์ TM

| ประเภทของแอคเตอร์ | จำนวน | ค่าน้ำหนัก | ค่าน้ำหนักแอคเตอร์ |
|---|---|---|---|
| Complex | 1 | 3 | 3 |
| Average | 1 | 2 | 2 |
| Simple | 0 | 1 | 0 |
| | | | |
| **ผลรวมของค่าน้ำหนักปัจจัย** | | | 5 |

2) การหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Weights หรือ UUCW)

การหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า จะต้องแบ่งยูสเคสออกเป็น 3 ระดับ คือ ระดับไม่ซับซ้อน (Simple), ระดับปานกลาง (Average) และ ระดับซับซ้อน (Complex) ค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่าจะหาได้จากการนำค่าน้ำหนักปัจจัยคูณกับจำนวนรายการ การหาค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่าของโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงในตารางที่ 97-106

ตารางที่ 97 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ HHTP

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 6 | 5 | 30 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 30 |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 98 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ PTT

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 7 | 5 | 35 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 35 |

ตารางที่ 99 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ KAPP

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 11 | 5 | 55 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 55 |

ตารางที่ 100 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ EATD

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 9 | 5 | 45 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 45 |

ตารางที่ 101 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ ST

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 15 | 5 | 75 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 75 |

ตารางที่ 102 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ CINOUT

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 4 | 5 | 20 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 20 |

ตารางที่ 103 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ MINSH

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 4 | 5 | 20 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 20 |

ตารางที่ 104 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ LMS

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 12 | 5 | 60 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 60 |

ตารางที่ 105 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ LEAP

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 7 | 5 | 35 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 35 |

ตารางที่ 106 ประเภทของยูสเคส, ค่าน้ำหนักปัจจัย, จำนวนรายการ และค่าน้ำหนักของยูสเคสที่ยังไม่ได้ปรับค่า ของโครงการซอฟต์แวร์ TM

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| ประเภทของยูสเคส | จำนวน | ค่าน้ำหนัก | ค่า UUCW |
|---|---|---|---|
| Simple | 7 | 5 | 35 |
| Average | 0 | 10 | |
| Complex | 0 | 15 | |
| ผลรวมทั้งหมด | | | 35 |

3) การหาค่ายูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (Unadjusted Use Case Point หรือ UUCP)

การหาค่ายูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (UUCP) จะหาได้จากการนำค่านำหนักของแอคเตอร์ที่ยังไม่ได้ปรับค่า (UAW) บวกกับค่าน้ำหนักของยูสเคสที่ไม่ได้ปรับค่า (UUCW) ดังสูตรต่อไปนี้

UUCP = UAW + UUCW

โดยการหายูสเคสพอยต์ที่ยังไม่ได้ปรับค่าของโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงในตารางที่ 107

ตารางที่ 107 ผลลัพธ์ค่าประเมินของยูสเคสพอยต์ที่ยังไม่ได้ปรับค่า

| ลำดับโครงการ | ชื่อโครงการ | UAW | UUCW | UUCP |
|---|---|---|---|---|
| 1 | HHTP | 7 | 30 | 37 |
| 2 | PTT | 4 | 35 | 39 |
| 3 | KAPP | 11 | 55 | 66 |
| 4 | EATD | 8 | 45 | 53 |
| 5 | ST | 18 | 75 | 93 |
| 6 | CINOUT | 11 | 20 | 31 |
| 7 | MINSH | 9 | 35 | 44 |
| 8 | LMS | 10 | 60 | 70 |
| 9 | LEAP | 6 | 35 | 41 |
| 10 | TM | 5 | 35 | 40 |

4) การหาค่าความซับซ้อนทางเทคนิค (Technical Complexity Factor หรือ TCF)

จากบทที่ 2 การคำนวณค่าความซับซ้อนทางเทคนิคจะหาได้จากผลรวมของการประเมินค่าปัจจัยทางเทคนิค 13 ประการ (T1 – T13) คูณกับค่าน้ำหนักของแต่ละปัจจัย ดังสูตรต่อไปนี้

$$\text{TFactor} = \sum T_i(W_i)$$

ซึ่งค่าประเมินค่าปัจจัยทางเทคนิค 13 ประการของโครงการซอฟต์แวร์ทั้ง 10 โครงการ แสดงในตารางที่ 108

ตารางที่ 108 ผลลัพธ์ของค่าประเมินค่าความซับซ้อนทางเทคนิค ทั้ง 13 ปัจจัย ของโครงการซอฟต์แวร์ในการทดลอง ทั้ง 10 โครงการ

| TFactor/Software | HHTP | PTT | KAPP | EATD | ST | CINOUT | MINSH | LMS | LEAP | TM |
|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 0x2 | 0x2 | 3x2 | 0x2 | 3x2 | 0x2 | 0x2 | 3x2 | 0x2 | 0x2 |
| T2 | 0x2 | 0x2 | 3x2 | 0x2 | 3x2 | 0x2 | 0x2 | 3x2 | 0x2 | 3x2 |
| T3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T4 | 0 | 0 | 3 | 0 | 5 | 0 | 0 | 3 | 0 | 0 |
| T5 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 3 | 0 | 0 |
| T6 | 5x0.5 | 5x0.5 | 5x0.5 | 5x0.5 | 0x0.5 | 5x0.5 | 0x0.5 | 3x0.5 | 0x0.5 | 0x0.5 |
| T7 | 0x0.5 | 0x0.5 | 3x0.5 | 0x0.5 | 5x0.5 | 0x0.5 | 0x0.5 | 3x0.5 | 0x0.5 | 3x0.5 |
| T8 | 5x2 | 0x2 | 5x2 | 0x2 | 3x2 | 5x2 | 3x2 | 5x2 | 3x2 | 3x2 |
| T9 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| T10 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| T11 | 3 | 3 | 3 | 3 | 5 | 3 | 3 | 3 | 3 | 3 |
| T12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T13 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 3 | 0 | 0 |

คำนวณหา TCF ได้จากสูตรดังนี้

$$TCF = 0.6 + (0.01 \times TFactor)$$

ค่าประเมินค่าความซับซ้อนทางเทคนิค ของโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงในตารางที่ 109

ตารางที่ 109 ผลลัพธ์ของค่าประเมินค่าความซับซ้อนทางเทคนิค ภายหลังจากการคำนวณของทั้ง 13 ปัจจัย ของโครงการซอฟต์แวร์ในการทดลอง ทั้ง 10 โครงการ

| Software | HHTP | PTT | KAPP | EATD | ST | CINOUT | MINSH | LMS | LEAP | TM |
|---|---|---|---|---|---|---|---|---|---|---|
| TCF | 0.735 | 0.735 | 1.03 | 0.765 | 1.045 | 0.835 | 0.77 | 1.05 | 0.77 | 0.845 |

5) การหาค่าความซับซ้อนทางสิ่งแวดล้อม (Environment Complexity Factor หรือ ECF)

จากบทที่ 2 การคำนวณค่าความซับซ้อนทางสิ่งแวดล้อมจะหาได้จากผลรวมของการประเมินค่าปัจจัยทางสิ่งแวดล้อม 8 ประการ (E1 – E8) คูณกับค่าน้ำหนักของแต่ละปัจจัย ดังสูตรต่อไปนี้

$$\text{EFactor} = \sum E_i(W_i)$$

ซึ่งค่าประเมินค่าปัจจัยทางสิ่งแวดล้อม 8 ประการของโครงการซอฟต์แวร์ทั้ง 10 โครงการ แสดงในตารางที่ 110

ตารางที่ 110 ผลลัพธ์ของค่าประเมินค่าความซับซ้อนทางสิ่งแวดล้อมทั้ง 8 ปัจจัย ของโครงการซอฟต์แวร์ ทั้ง 10 โครงการ

| EFactor/Software | HHTP | PTT | KAPP | EATD | ST | CINOUT | MINSH | LMS | LEAP | TM |
|---|---|---|---|---|---|---|---|---|---|---|
| E1 | 4x1.5 | 3x1.5 | 2x1.5 | 2x1.5 | 2x1.5 | 3x1.5 | 5x1.5 | 2x1.5 | 3x1.5 | 3x1.5 |
| E2 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 |
| E3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| E4 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 | 3x0.5 |
| E5 | 3 | 3 | 3 | 4 | 3 | 4 | 5 | 3 | 3 | 2 |
| E6 | 3x2 | 3x2 | 3x2 | 3x2 | 3x2 | 3x2 | 3x2 | 3x2 | 3x2 | 3x2 |
| E7 | 0(-1) | 3(-1) | 2x(-1) | 5(-1) | 5(-1) | 0(-1) | 5(-1) | 5x(-1) | 4(-1) | 0(-1) |
| E8 | 3x2 | 4x2 | 5x2 | 4x2 | 4x2 | 3x2 | 3x2 | 5x2 | 3x2 | 3x2 |
| TOTAL | 27 | 24.5 | 26 | 22 | 21 | 22.5 | 21 | 23 | 21.5 | 24.5 |

คำนวณหา ECF ได้จากสูตรดังนี้

$$\text{ECF} = 1.4 + (-0.03 \times \text{EFactor})$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ค่าประเมินค่าความซับซ้อนทางสิ่งแวดล้อม ของโครงการซอฟต์แวร์ทั้ง 10 โครงการแสดงในตารางที่ 111

ตารางที่ 111 ผลลัพธ์ของการคำนวณค่าความซับซ้อนทางสิ่งแวดล้อมทั้ง 8 ปัจจัย ของโครงการซอฟต์แวร์ ทั้ง 10 โครงการ

| Software | HHTP | PTT | KAPP | EATD | ST | CINOUT | MINSH | LMS | LEAP | TM |
|---|---|---|---|---|---|---|---|---|---|---|
| ECF | 0.59 | 0.665 | 0.62 | 0.74 | 0.77 | 0.725 | 0.77 | 0.71 | 0.755 | 0.665 |

6) การหาค่ายูสเคสพอยต์ที่ปรับค่าแล้ว (Adjusted Use Case Point หรือ UCP)

การคำนวณหาค่ายูสเคสพอยต์ที่ปรับค่าแล้วจะคำนวณได้จากค่า ยูสเคสพอยต์ที่ยังไม่ได้ปรับค่า (UUCP), ค่าความซับซ้อนทางเทคนิค (TCF) และค่าความซับซ้อนทางสิ่งแวดล้อม (ECF) ดังสูตรต่อไปนี้

UCP = UUCP x TCF x ECF

ค่าประเมินของยูสเคสพอยต์ที่ปรับค่าแล้วของโครงการซอฟต์แวร์ทั้ง 10 โครงการ แสดงในตารางที่ 112

ตารางที่ 112 ผลลัพธ์จากการประเมินค่ายูสเคสพอยต์ที่ปรับค่าแล้วของโครงการซอฟต์แวร์ ทั้ง 10 โครงการ

| ลำดับ | โครงการซอฟต์แวร์ | UUCP | TCF | ECF | UCP |
|---|---|---|---|---|---|
| 1 | HHTP | 37 | 0.735 | 0.59 | 16.04 |
| 2 | PTT | 39 | 0.735 | 0.665 | 19.06 |
| 3 | KAPP | 66 | 1.03 | 0.62 | 42.14 |
| 4 | EATD | 51 | 0.765 | 0.74 | 28.8 |
| 5 | ST | 93 | 1.045 | 0.77 | 74.83 |
| 6 | CINOUT | 31 | 0.835 | 0.725 | 18.76 |
| 7 | MINSH | 44 | 0.77 | 0.77 | 26.08 |
| 8 | LMS | 70 | 1.05 | 0.71 | 52.18 |
| 9 | LEAP | 41 | 0.77 | 0.755 | 23.83 |
| 10 | TM | 40 | 0.845 | 0.665 | 22.47 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

7) การคำนวณหาค่าความพยายามในการพัฒนาซอฟต์แวร์

ผู้วิจัยใช้ผลการศึกษาของ Karner สำหรับการประมาณค่า 1 ยูสเคสพอยต์ เท่ากับ 20 ชั่วโมง นั้นคือค่าความพยายามในการพัฒนาซอฟต์แวร์ในหน่วย man-hour จะมีค่าเท่ากับ UCP x 20 และผู้วิจัยจะแปลงค่าความพยายามพยายามในการพัฒนาซอฟต์แวร์ในหน่วย man-hour เป็นหน่วย man-month โดยการนำผลลัพธ์ที่ได้ หารด้วย 152 (1 man-month = 152 man-hour) การ

ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบยูสเคสพอยต์ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ แสดงในตารางที่ 113

ตารางที่ 113 ค่าประเมินจากการประเมินด้วยยูสเคสพอยต์ของโครงการซอฟต์แวร์ในการทดลองทั้ง 10 โครงการ

| ลำดับ | โครงการซอฟต์แวร์ | UCP | ค่าความพยายาม (man-hour) | ค่าความพยายาม (man-month) |
|---|---|---|---|---|
| 1 | HHTP | 16.04 | 320.8 | 2.11 |
| 2 | PTT | 19.06 | 381.2 | 2.5 |
| 3 | KAPP | 42.1 | 842 | 5.53 |
| 4 | EATD | 28.8 | 576 | 3.79 |
| 5 | ST | 74.83 | 1496.6 | 9.84 |
| 6 | CINOUT | 18.76 | 375.2 | 2.46 |
| 7 | MINSH | 26.08 | 521.6 | 3.43 |
| 8 | LMS | 52.18 | 1043.6 | 6.86 |
| 9 | LEAP | 23.83 | 476.6 | 3.13 |
| 10 | TM | 22.47 | 449.4 | 2.95 |

สรุปการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบขั้นตอนวิธีทั้ง 3 ตัวแบบของโครงการซอฟต์แวร์ทั้ง 10 โครงการ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยได้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีทั้ง 3 คือ ฟังก์ชันพอยต์ ยูสเคสพอยต์ และ โคโคโม่ ของโครงการซอฟต์แวร์ที่ใช้ในการทดลองทั้ง 10 โครงการ มีผลสรุปค่าความพยายามในการพัฒนาซอฟต์แวร์ดังตารางที่ 114

ตารางที่ 114 สรุปค่าความพยายามในการพัฒนาซอฟต์แวร์จากการประเมินด้วยวิธีฟังก์ชันพอยต์ ยูสเคสพอยต์ และ โคโคโม่ (หน่วย คน-เดือน)

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าความพยายามจากตัวแบบฟังก์ชันพอยต์ | ค่าความพยายามจากตัวแบบโคโคโม่ | ค่าความพยายามจากตัวแบบยูสเคสพอยต์ | ค่าความพยายามที่ใช้จริง |
|---|---|---|---|---|---|
| 1 | HHTP | 1.5 | 15.1 | 2.11 | 2.5 |
| 2 | PTT | 1 | 5.4 | 2.5 | 7.5 |
| 3 | KAPP | 2.5 | 25.5 | 5.53 | 8.5 |
| 4 | EATD | 1.5 | 19.8 | 3.79 | 5 |
| 5 | ST | 5.5 | 117.6 | 9.84 | 14 |
| 6 | CINOUT | 1 | 5.4 | 2.46 | 3 |
| 7 | MINSH | 3 | 30.4 | 3.43 | 3 |
| 8 | LMS | 4 | 41 | 6.86 | 3 |
| 9 | LEAP | 1.5 | 16.0 | 3.13 | 5.5 |
| 10 | TM | 1.5 | 11.1 | 2.95 | 4 |

**4.4 ความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากตัวแบบวิธีทั้งสาม**

ผู้วิจัยหาความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของตัวแบบขั้นตอนวิธีทั้ง 3 ตามสูตรการหาค่าความคลาดเคลื่อนสัมพัทธ์ดังนี้

$$\text{MRE} = \left| \frac{MMest - MMact}{MMact} \right|$$

โดย

MMact คือ ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง

MMest คือ ค่าประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์จากผู้ประเมิน

ค่าความคลาดเคลื่อนสัมพัทธ์ที่ได้จากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของตัวแบบขั้นตอนวิธีทั้งสาม แสดงในตารางที่ 115

ตารางที่ 115 ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของตัวแบบขั้นตอนวิธี ฟังก์ชันพอยต์ โคโคโม่ และ ยูสเคสพอยต์ ของโครงการซอฟต์แวร์ที่ใช้ในการทดลองทั้ง 10 โครงการ

| ลำดับที่ | โครงการซอฟต์แวร์ | ค่าความคลาดเคลื่อนของตัวแบบฟังก์ชันพอยต์ | ค่าความคลาดเคลื่อนของตัวแบบโคโคโม่ | ค่าความคลาดเคลื่อนของตัวแบบยูสเคสพอยต์ |
|---|---|---|---|---|
| 1 | HHTP | 0.40 | 5.04 | 0.156 |
| 2 | PTT | 0.80 | 0.28 | 0.660 |
| 3 | KAPP | 0.70 | 2.00 | 0.349 |
| 4 | EATD | 0.70 | 2.96 | 0.242 |
| 5 | ST | 13.60 | 7.40 | 0.297 |
| 6 | CINOUT | 0.66 | 0.80 | 0.180 |
| 7 | MINSH | 0 | 9.13 | 0.143 |
| 8 | LMS | 0.33 | 12.66 | 1.286 |
| 9 | LEAP | 0.73 | 1.91 | 0.430 |
| 10 | TM | 0.63 | 1.78 | 0.2625 |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี ทั้งสามตัวแบบคือ ฟังก์ชันพอยต์ โคโคโม่ และยูสเคสพอยต์ ได้ให้ค่าความแม่นยำของผลลัพธ์ที่แตกต่างกันจากการประเมินโครงการซอฟต์แวร์ทั้ง 10 โครงการ ตัวแบบการประเมินฟังก์ชันพอยต์ให้ค่าประเมินแม่นยำที่สุดสำหรับโครงการซอฟต์แวร์ MINSH และ LMS ตัวแบบการประเมินโคโคโม่ให้ค่าประเมินแม่นยำที่สุดสำหรับโครงการซอฟต์แวร์ PTT และตัวแบบการประเมินยูสเคสพอยต์ให้ค่า

ประเมินแม่นยำที่สุดสำหรับโครงการซอฟต์แวร์ HHTP KAPP EATD ST CINOUT LEAP และ TM

**4.5 การประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ**

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ผู้วิจัยได้ให้ผู้ประเมินทั้ง 5 คนร่วมกันประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ในห้องเดียวกัน ผู้วิจัยจะเป็นคนบรรยายความต้องการด้านหน้าที่และไม่ใช่ด้านหน้าที่ (Functional and Non Functional Requirement) ของโครงการซอฟต์แวร์ทั้ง 10 โครงการซอฟต์แวร์ โดยจะอธิบายและให้ผู้ประเมินทำการประเมินทีละโครงการในระหว่างที่มีการประเมิน ผู้ประเมินทั้ง 5 คนจะไม่สามารถคุยหรือปรึกษากันได้ หากมีคำถามเกี่ยวกับความต้องการด้านหน้าที่และไม่ใช่ด้านหน้าที่ของโครงการซอฟต์แวร์ทั้ง 10 โครงการ ผู้ประเมินสามารถถามผู้วิจัยได้ ก่อนเริ่มต้นการประเมินผู้วิจัยจะให้ผู้ประเมินตอบแบบสอบถามที่ 1 (ภาคผนวก ข) และภายหลังประเมินโครงการซอฟต์แวร์แต่ละโครงการเสร็จเรียบร้อยแล้ว ผู้ประเมินจะกรอกแบบสอบถามที่ 2 (ภาคผนวก ค) ภายหลังจากการประเมินครั้งแรก ผู้วิจัยจะแสดงค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี เพื่อให้ผู้ประเมินตัดสินใจอีกครั้งว่าจะเปลี่ยนแปลงค่าประเมินที่ตนประเมินได้หรือไม่ หากไม่เปลี่ยนแปลงค่าประเมินผู้ประเมินจะกรอกตัวเลขเดิมลงไปในแบบฟอร์มบันทึกการประเมิน ค่าประเมินครั้งที่ 1 (ก่อนการเห็นค่าประเมินของวิธีตัวแบบขั้นตอนวิธี) และครั้งที่ 2 (หลังการเห็นค่าประเมินจากวิธีตัวแบบขั้นตอนวิธี) ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ทั้ง 10 โครงการ แสดงในตารางที่ 116

Chulalongkorn University

ตารางที่ 116 ค่าประเมินจากผู้เชี่ยวชาญทั้ง 5 ในการประเมินโครงการซอฟต์แวร์ทั้ง 10 โครงการ

| ลำดับที่ | โครงการซอฟต์แวร์ | ผู้เชี่ยวชาญคนที่ 1 | | ผู้เชี่ยวชาญคนที่ 2 | | ผู้เชี่ยวชาญคนที่ 3 | | ผู้เชี่ยวชาญคนที่ 4 | | ผู้เชี่ยวชาญคนที่ 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 |
| 1 | HHTP | 2.5 | 2.5 | 3 | 4 | 3 | 3 | 2 | 2 | 1 | 1 |
| 2 | PTT | 1.5 | 1.5 | 3 | 2.5 | 4.5 | 4 | 3.5 | 3 | 6 | 5 |
| 3 | KAPP | 5 | 5 | 6 | 6 | 2.5 | 2.5 | 2.5 | 2.5 | 3.5 | 3 |
| 4 | EATD | 3 | 3 | 4 | 4 | 2.5 | 3 | 3 | 3 | 2.5 | 2.5 |
| 5 | ST | 7 | 8 | 10 | 10 | 5 | 6 | 4 | 6 | 4 | 6 |

| ลำดับที่ | โครงการซอฟต์แวร์ | ผู้เชี่ยวชาญคนที่ 1 | | ผู้เชี่ยวชาญคนที่ 2 | | ผู้เชี่ยวชาญคนที่ 3 | | ผู้เชี่ยวชาญคนที่ 4 | | ผู้เชี่ยวชาญคนที่ 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 1 | ครั้งที่ 2 |
| 6 | CINOUT | 2 | 2 | 1 | 1.5 | 2 | 2 | 1 | 1 | 0.7 | 1 |
| 7 | MINSH | 4.5 | 4 | 3.5 | 3.5 | 1.5 | 2 | 1.5 | 2 | 2 | 2 |
| 8 | LMS | 3.5 | 4 | 5.5 | 6 | 2 | 3 | 2.5 | 2.5 | 2 | 2 |
| 9 | LEAP | 2 | 2 | 2.5 | 3 | 1.5 | 1.5 | 1 | 1 | 1 | 1 |
| 10 | TM | 1.5 | 1.5 | 1.5 | 1.5 | 2.5 | 2.5 | 1.5 | 1.5 | 1.5 | 1.5 |

**4.6 ความคลาดเคลื่อนสัมพัทธ์ของค่าประเมินที่ได้จากวิธีการตัดสินของผู้เชี่ยวชาญ**

ค่าประเมินที่ได้จากผู้ประเมินในครั้งที่ 1 ของการประเมิน (ก่อนการเห็นค่าประเมินจากวิธีตัวแบบขั้นตอนวิธี) ผู้วิจัยนำมาหาค่าความคลาดเคลื่อนสัมพัทธ์ แสดงในตารางที่ 117

ตารางที่ 117 ค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ของผู้ประเมินทั้ง 5 คน (การประเมินในครั้งที่ 1)

| ลำดับที่ | โครงการซอฟต์แวร์ | ผู้ประเมินคนที่ 1 | ผู้ประเมินคนที่ 2 | ผู้ประเมินคนที่ 3 | ผู้ประเมินคนที่ 4 | ผู้ประเมินคนที่ 5 |
|---|---|---|---|---|---|---|
| 1 | HHTP | 0 | 0.6 | 0.2 | 0.2 | 0.6 |
| 2 | PTT | 0.8 | 0.67 | 0.47 | 0.6 | 0.33 |
| 3 | KAPP | 0.41 | 0.29 | 0.71 | 0.71 | 0.65 |
| 4 | EATD | 0.4 | 0.2 | 0.4 | 0.4 | 0.5 |
| 5 | ST | 0.5 | 0.29 | 0.57 | 0.64 | 0.57 |
| 6 | CINOUT | 0.33 | 0.5 | 0.33 | 0.67 | 0.67 |
| 7 | MINSH | 0.5 | 0.17 | 0.33 | 0.33 | 0.33 |
| 8 | LMS | 0.16 | 1 | 0 | 0.17 | 0.33 |
| 9 | LEAP | 0.7 | 0.73 | 0.73 | 0.73 | 0.82 |
| 10 | TM | 0.63 | 0.375 | 0.38 | 0.63 | 0.63 |
| | MMRE | 0.4430 | 0.4825 | 0.4120 | 0.5067 | 0.5126 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยจะหาค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ โดยหาค่าเฉลี่ยของค่าความคลาดเคลื่อนสัมพัทธ์ของผู้ประเมินทั้ง 5 คน และนำมาเปรียบเทียบกับค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของวิธีตัวแบบขั้นตอนวิธี ทั้ง 3 ตัวแบบ

การเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีตัวแบบขั้นตอนวิธี ซึ่งประกอบด้วย ฟังก์ชันพอยต์ โคโคโม่ และยูสเคสพอยต์ กับวิธีการตัดสินของผู้เชี่ยวชาญ สรุปได้ดังตารางที่ 118

ตารางที่ 118 สรุปผลค่าความคลาดเคลื่อนเฉลี่ยของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี ฟังก์ชันพอยต์ โคโคโม่ ยูสเคสพอยต์ และวิธีการตัดสินของผู้เชี่ยวชาญ

| วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ | จำนวนโครงการ | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย |
|---|---|---|
| ฟังก์ชันพอยต์ | 10 | 1.8550 |
| โคโคโม่ | 10 | 4.3960 |
| ยูสเคสพอยต์ | 10 | 0.4005 |
| การตัดสินของผู้เชี่ยวชาญ 5 คน | 10 x 5 = 50 | 0.4713 |

จากตารางพบว่าการประเมินด้วยวิธียูสเคสพอยต์ มีความแม่นยำมากที่สุด (ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย 0.4005) รองลงมาคือ วิธีการตัดสินของผู้เชี่ยวชาญ (ค่าความคาดเคลื่อนสัมพัทธ์เฉลี่ย 0.4713) ฟังก์ชันพอยต์ (ค่าความคลาดเคลื่อนเฉลี่ย 1.8550) และ โคโคโม่ (ค่าความคาดเคลื่อนเฉลี่ย 4.3960) ผู้วิจัยจึงต้องการทดสอบสมมติฐานที่ 1 เพื่อให้ทราบแน่ชัดว่าการผลลัพธ์ที่ได้มีความแตกต่างกันอย่างมีนัยสำคัญหรือไม่

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

**4.7 การทดสอบสมมติฐาน**

**การทดสอบสมมติฐานที่ 1**

สมมติฐานที่ 1

จากคำถามวิจัยในบทที่ผ่านมา เกี่ยวกับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ภายในสภาวะแวดล้อมที่ผู้เป็นคนไทยด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าหรือ

น้อยกว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี สามารถเขียนเป็นสมมติฐานได้ดังนี้

$H_0$ = ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีขั้นตอนวิธี **มากกว่าหรือเท่ากับ** ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

$H_1$ = ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีขั้นตอนวิธี **น้อยกว่า** ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

**การทดสอบความเป็นการกระจายแบบปกติ (Normality Test)**

กำหนดสมมติฐานดังนี้

$H_0$ : ข้อมูลมีการกระจายแบบปกติ

$H_1$ : ข้อมูลไม่มีการกระจายแบบปกติ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบความเป็นการกระจายแบบปกติ มีผลวิเคราะห์ดังตารางที่ 119-122

ตารางที่ 119 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบโคโคโม่ ก่อนการทดสอบสมมติฐาน

| Kolmogorov-Sminov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|
| Static | df | p-value | Static | df | p-value |
| 0.237 | 10 | 0.118 | 0.872 | 10 | 0.106 |

ตารางที่ 120 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบฟังก์ชันพอยต์ ก่อนการทดสอบสมมติฐาน

| Kolmogorov-Sminov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|
| Static | df | p-value | Static | df | p-value |
| 0.501 | 10 | 0 | 0.42 | 10 | 0 |

ตารางที่ 121 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบยูสเคสพอยต์ ก่อนการทดสอบสมมติฐาน

| Kolmogorov-Sminov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|
| Static | df | p-value | Static | df | p-value |
| 0.27 | 10 | 0.038 | 0.735 | 10 | 0.002 |

ตารางที่ 122 การทดสอบความเป็นการกระจายแบบปกติ ข้อมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ก่อนการทดสอบสมมติฐาน

| Kolmogorov-Sminov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|
| Static | df | p-value | Static | df | p-value |
| 0.110 | 50 | 0.178 | 0.974 | 50 | 0.346 |

จากการทดสอบความเป็นการกระจายแบบปกติพบว่า

จากตารางที่ 119 การทดสอบความเป็นการกระจายแบบปกติของข้องมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบโคโคโม่ ค่า p-value มีค่ามากกว่า 0.05 จึงทำให้ยอมรับ $H_0$

จากตารางที่ 120 การทดสอบความเป็นการกระจายแบบปกติของข้องมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบฟังก์ชันพอยต์ ค่า p-value มีค่าน้อยกว่า 0.05 จึงทำให้ปฏิเสธรับ $H_0$

จากตารางที่ 121 การทดสอบความเป็นการกระจายแบบปกติของข้องมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยตัวแบบยูสเคสพอยต์ ค่า p-value มีค่าน้อยกว่า 0.05 จึงทำให้ปฏิเสธ $H_0$

จากตารางที่ 122 การทดสอบความเป็นการกระจายแบบปกติของข้องมูลค่าความคลาดเคลื่อนเฉลี่ยของค่าประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ค่า p-value มีค่ามากกว่า 0.05 จึงทำให้ยอมรับ $H_0$

การทดสอบสมมติฐานที่ 1 ผู้วิจัยจะใช้ การทดสอบ Kruskal-Wallis เพื่อหาความแตกต่างของค่ากลางของความคลาดเคลื่อนสัมพัทธ์ระหว่างวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบขั้นตอนวิธีทั้ง 3 กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 = M_1 = M_2 = M_3 = M_4$

$H_1 = M_i \neq M_j$ อย่างน้อย 1 ค่า เมื่อ $i \neq j\ ;\ i,j = 1,2,3,4$

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

**การทดสอบ Kruskal-Wallis**

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 4.29

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 123 ผลการทดสอบสมมติฐานด้วยวิธี Kruskal-Wallis

| Test | p-value | Decision |
|---|---|---|
| Kruskal-Wallis Test | 0 | Reject the null hypothesis. |

จากข้อมูลสรุปได้ว่า ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีโคโคโม่ ยูสเคสพอยต์ และวิธีการตัดสินของผู้เชี่ยวชาญมี**ความแตกต่างกัน** อย่างน้อย 1 คู่ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ผู้วิจัยจึงได้ทดสอบความแตกต่างกันของค่าความคลาดเคลื่อนสัมพัทธ์ในแต่ละคู่ รวมทั้งหมด 6 คู่

ผู้วิจัยจะดำเนินการทดสอบทีละคู่ตามลำดับดังนี้

1) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบยูสเคสพอยต์

2) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบโคโคโม่และวิธีการตัดสินของผู้เชี่ยวชาญ

3) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบฟังก์ชันพอยต์และวิธีการตัดสินของผู้เชี่ยวชาญ

4) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบโคโคโม่และตัวแบบยูสเคสพอยต์

5) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบฟังก์ชันพอยต์และตัวแบบยูสเคสพอยต์

6) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบโคโคโม่และตัวแบบฟังก์ชันพอยต์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

การทดสอบสมมติฐานของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ทั้ง 6 คู่ เนื่องจากมีบางคู่ที่มีการกระจายของข้อมูลปกติ จึงไม่สามารถใช้การทดสอบที กับการทดลองทั้งหมดได้ ผู้วิจัยจะใช้การทดสอบสมมติฐานด้วยวิธี Mann-Whitney เพื่อให้การทดสอบสมมติฐานเป็นวิธีเดียวกันและสามารถเปรียบเทียบกันได้

**การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ทีละคู่**

1) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบยูสเคสพอยต์

**การทดสอบ Mann-Whitney**

กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 : M_1 \leq M_2$

$H_1 : M_1 > M_2$

เมื่อ $M_1$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

และ $M_2$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของการประเมินด้วยตัวแบบยูสเคสพอยต์

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 124

ตารางที่ 124 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญและตัวแบบยูสเคสพอยต์

| Mann-Whitney U | Z | p-value |
|---|---|---|
| 163 | -1.728 | 0.042 |

จากการทดสอบพบว่า ค่า p-value มีค่าน้อยกว่า 0.05 จึงปฏิเสธ $H_0$ และสรุปได้ว่า ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของวิธีการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญมากกว่าค่า

กลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของวิธีการประเมินด้วยตัวแบบยูสเคสพอยต์ ที่ระดับนัยสำคัญ 0.05

2) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบโคโคโม่และวิธีการตัดสินของผู้เชี่ยวชาญ

**การทดสอบ Mann-Whitney**

กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 : M_1 \leq M_2$

$H_1 : M_1 > M_2$

เมื่อ $M_1$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบโคโคโม่ และ $M_2$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 125

ตารางที่ 125 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบโคโคโม่และวิธีการตัดสินของผู้เชี่ยวชาญ

| Mann-Whitney U | Z | p-value |
|---|---|---|
| 44.5 | -4.082 | 0 |

จากการทดสอบพบว่า ค่า p-value มีค่าน้อยกว่า 0.05 จึงทำให้ปฏิเสธ $H_0$ นั่นคือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบโคโคโม่มากกว่าค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ที่ระดับนัยสำคัญ 0.05

3) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบฟังก์ชันพอยต์และวิธีการตัดสินของผู้เชี่ยวชาญ

**การทดสอบ Mann-Whitney**

กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 : M_1 \leq M_2$

$H_1 : M_1 > M_2$

เมื่อ $M_1$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบฟังก์ชันพอยต์ และ $M_2$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 126

ตารางที่ 126 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบฟังก์ชันพอยต์และวิธีการตัดสินของผู้เชี่ยวชาญ

| Mann-Whitney U | Z | p-value |
|---|---|---|
| 169 | -1.610 | 0.055 |

จากการทดสอบพบว่าค่า p-value มีค่ามากกว่า 0.05 จึงทำให้ยอมรับ $H_0$ นั่นคือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบฟังก์ชันพอยต์ไม่มากกว่าค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ ที่ระดับนัยสำคัญ 0.05

4) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบโคโคโม่และตัวแบบยูสเคสพอยต์

**การทดสอบ Mann-Whitney**

กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 : M_1 \leq M_2$

$H_1 : M_1 > M_2$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เมื่อ $M_1$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบโคโคโม่

และ $M_2$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบยูสเคสพอยต์

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 127

ตารางที่ 127 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัวแบบโคโคโม่และตัวแบบยูสเคสพอยต์

| Mann-Whitney U | Z | p-value |
|---|---|---|
| 6 | -3.326 | 0 |

จากการทดสอบพบว่าค่า p-value มีค่าน้อยกว่า 0.05 จึงทำให้ปฏิเสธรับ $H_0$ นั่นคือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบโคโคโม่มากกว่าค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบยูสเคสพอยต์ ที่ระดับนัยสำคัญ 0.05

5) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบฟังก์ชันพอยต์และตัวแบบยูสเคสพอยต์

**การทดสอบ Mann-Whitney**

กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 : M_1 \leq M_2$

$H_1 : M_1 > M_2$

เมื่อ $M_1$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบฟังก์ชันพอยต์ และ $M_2$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบยูสเคสพอยต์

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 128

ตารางที่ 128 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัวแบบฟังก์ชันพอยต์และตัวแบบยูสเคสพอยต์

| Mann-Whitney U | Z | p-value |
|---|---|---|
| 24.5 | -1.928 | 0.027 |

จากการทดสอบพบว่าค่า p-value มีค่าน้อยกว่า 0.05 จึงทำให้ปฏิเสธ $H_0$ นั่นคือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบฟังก์ชันพอยต์มากกว่าค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบยูสเคสพอยต์ ที่ระดับนัยสำคัญ 0.05

6) การทดสอบทางสถิติของการเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยตัวแบบโคโคโม่และตัวแบบฟังก์ชันพอยต์

กำหนดสมมติฐานทางสถิติ ดังนี้

$H_0 : M_1 \leq M_2$

$H_1 : M_1 > M_2$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เมื่อ $M_1$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบโคโคโม่

และ $M_2$ คือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยตัวแบบฟังก์ชันพอยต์

กำหนดระดับนัยสำคัญ

$$\alpha = 0.05$$

**การทดสอบ Mann-Whitney**

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 129

ตารางที่ 129 การทดสอบ Mann-Whitney ของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินด้วยวิธีการตัวแบบโคโคโม่และตัวแบบฟังก์ชันพอยต์

| Mann-Whitney U | Z | p-value |
|---|---|---|
| 18.5 | -2.382 | 0.008 |

จากการทดสอบพบว่าค่า p-value มีค่าน้อยกว่า 0.05 จึงทำให้ปฏิเสธ $H_0$ นั่นคือ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมินด้วยตัวแบบโคโคโม่มากกว่าค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของวิธีการประเมินด้วยตัวแบบฟังก์ชันพอยต์ ที่ระดับนัยสำคัญ 0.05

**สรุปการทดสอบสมมติฐานที่ 1**

การทดสอบสมมติฐานที่ 1 ผู้วิจัยได้เปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีตัวแบบขั้นตอนวิธี 3 ตัวแบบ ประกอบด้วย ตัวแบบยูสเคสพอยต์ ตัวแบบโคโคโม่ และตัวแบบฟังก์ชันพอยต์ กับวิธีการตัดสินของผู้เชี่ยวชาญ และทดสอบสมมติฐานโดยใช้วิธี Kruskal-Wallis ค่า p-value ที่ได้พบว่าค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์มีความแตกต่างกันอย่างมีนัยสำคัญ ผู้วิจัยจึงเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ทีละคู่ ด้วยวิธี Mann-Whitney ผลความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์สามารถสรุปได้ดังตารางที่ 130

ตารางที่ 130 ผลการทดสอบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์และค่า p-value ของวิธีการประเมิน 6 คู่

| การเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมิน 2 วิธี | p-value | การเปรียบเทียบค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ |
|---|---|---|
| วิธีการตัดสินของผู้เชี่ยวชาญ กับ ตัวแบบยูสเคสพอยต์ | 0.042 | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการตัดสินของผู้เชี่ยวชาญมากกว่าตัวแบบยูสเคส |

| การเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมิน 2 วิธี | p-value | การเปรียบเทียบค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ |
|---|---|---|
| | | พอยต์ |
| ตัวแบบโคโคโม่ กับ วิธีการตัดสินของผู้เชี่ยวชาญ | 0 | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการตัดสินของผู้เชี่ยวชาญไม่มากกว่าตัวแบบโคโคโม่ |
| ตัวแบบฟังก์ชันพอยต์ กับ วิธีการตัดสินของผู้เชี่ยวชาญ | 0.055 | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบฟังก์ชันพอยต์ ไม่มากกว่า วิธีการตัดสินของผู้เชี่ยวชาญ |
| ตัวแบบโคโคโม่ กับ ตัวแบบยูสเคสพอยต์ | 0 | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่มากกว่าตัวแบบยูสเคสพอยต์ |
| ตัวแบบยูสเคสพอยต์ กับ ตัวแบบฟังก์ชันพอยต์ | 0.027 | ค่ากลางของค่าความคลาดเคลื่อนตัวแบบยูสเคสพอยต์ไม่มากกว่าตัวแบบฟังก์ชันพอยต์ |
| ตัวแบบโคโคโม่ กับ ตัวแบบฟังก์ชันพอยต์ | 0.008 | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่ มากกว่าตัวแบบฟังก์ชันพอยต์ |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 130 พบว่าที่ระดับนัยสำคัญ 0.05 ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการตัดสินของผู้เชี่ยวชาญ มากกว่า ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบยูสเคสพอยต์ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่ มากกว่า วิธีการตัดสินของผู้เชี่ยวชาญ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบฟังก์ชันพอยต์ ไม่มากกว่า ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการตัดสินของผู้เชี่ยวชาญ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่มากกว่าตัวแบบยูสเคสพอยต์ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบยูสเคสพอยต์ไม่มากกว่าตัวแบบฟังก์ชันพอยต์ และ ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่มากกว่าตัวแบบฟังก์ชันพอยต์ โดยลำดับความแม่นยำสามารถเรียงลำดับได้ดังนี้

Usecase Point > Expert Judgment > Function Point > COCOMO

**การทดสอบสมมติฐานที่ 2**

จากคำถามวิจัย เกี่ยวกับค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญจะช่วยให้การประเมินแม่นยำขึ้นหรือไม่ และผู้วิจัยต้องการทราบเพิ่มเติมเกี่ยวกับจำนวนการทดลองที่ผู้ประเมินเปลี่ยนแปลงค่าประเมินเมื่อเห็นค่าประเมินจากวิธีตัวแบบขั้นตอนวิธี ว่าค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี ทำให้ผู้ประเมินเปลี่ยนแปลงค่าประเมินมากกว่าครึ่งหนึ่งหรือไม่ ด้วยเหตุนี้จึงสามารถกำหนดสมมติฐานได้ตาม สมมติฐานที่ 2.1 และ สมมติฐานที่ 2.2 ดังนี้

**สมมติฐานที่ 2.1**

$H_0$ : ค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี **ทำให้ผู้ประเมินเปลี่ยนแปลงค่าประเมินเท่ากับครึ่งหนึ่งของจำนวนการทดลองทั้งหมด** (**จำนวนการทดลองทั้งหมด 50 การทดลอง**) จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินใจของผู้เชี่ยวชาญ

$H_1$ : ค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี **ทำให้ผู้ประเมินเปลี่ยนแปลงค่าประเมินน้อยกว่าครึ่งหนึ่งของจำนวนการทดลองทั้งหมด** (**จำนวนการทดลองทั้งหมด 50 การทดลอง**) จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินของผู้เชี่ยวชาญ

สถิติที่ใช้ทดสอบ : z-test

สมมติฐานที่ใช้ทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$H_0 : P \geq 0.5$$

$$H_1 : P < 0.5$$

เมื่อ P คือ ค่าเฉลี่ยที่ผู้ประเมินเปลี่ยนแปลงค่าประเมิน

คำนวณค่าสถิติ

$$z = \frac{p_0 - 0.5}{\sqrt{\frac{0.5(q_0)}{n}}}$$

เมื่อ

$p_0$ คือ อัตราส่วนจำนวนครั้งที่ผู้ประเมินเปลี่ยนแปลงค่าประเมินต่อจำนวนครั้งทั้งหมด

$q_0$ คือ อัตราส่วนจำนวนครั้งที่ผู้ประเมินไม่เปลี่ยนแปลงค่าประเมินต่อจำนวนครั้งทั้งหมด

n คือ จำนวนครั้งที่ผู้ประเมิน 5 คน ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ 10 โครงการ นั่นคือ

$p_0 = 0.4$

$q_0 = 0.6$

n = 50

ผู้วิจัยคำนวณโดยใช้โปรแกรม Excel เป็นเครื่องมือ ได้ผลการทดสอบสมมติฐานดังตารางที่ 131

ตารางที่ 131 การทดสอบ z-test ของโอกาสเปลี่ยนแปลงค่าประเมินของผู้เชี่ยวชาญทั้ง 5 คน

| | N | Mean | Std. | p-value |
|---|---|---|---|---|
| ตัวแบบยูสเคสพอยต์ | 50 | 0.4 | 0.1 | 0.135 |

p-value = 0.135 ซึ่งมากกว่า 0.05 จึงยอมรับ $H_0$

ดังนั้น จึงกล่าวได้ว่า ค่าเฉลี่ยของจำนวนครั้งที่ผู้ประเมินตัดสินใจเปลี่ยนแปลงค่าประเมินมีค่าเท่ากับ 0.5 หรือ ครึ่งหนึ่งของจำนวนครั้งที่ประเมินทั้งหมด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**สมมติฐานที่ 2.2**

$H_0$ : หากผู้ประเมินเห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำไม่แตกต่างจาก** ค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

$H_1$ : หากผู้ประเมินได้เห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำแตกต่างจาก** ค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

การทดสอบสมมติฐานที่ 2.2 ผู้วิจัยจะทดสอบจากข้อมูลสองกลุ่ม กลุ่มแรกคือข้อมูลที่มีการเปลี่ยนแปลงค่าความคลาดเคลื่อน จำนวน 20 ข้อมูล กลุ่มที่สองคือข้อมูลทั้งหมดทั้งที่เปลี่ยนแปลงค่าความคลาดเคลื่อนและไม่เปลี่ยนแปลงค่าความคลาดเคลื่อน จำนวน 50 ข้อมูล ข้อมูลที่มีการเปลี่ยนแปลงค่าความคลาดเคลื่อนแสดงในตารางที่ 132

ตารางที่ 132 ค่าความคลาดเคลื่อนเฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินและหลังการเปลี่ยนแปลงค่าประเมินของผู้ประเมินจำนวน 5 คน

| ผู้ประเมิน | ชื่อโครงการซอฟต์แวร์ | MRE ก่อนการเปลี่ยนแปลงค่าประเมิน | MRE หลังการเปลี่ยนแปลงค่าประเมิน | ผลต่างระหว่าง MRE ($D_i$) |
|---|---|---|---|---|
| ผู้ประเมินคนที่ 1 | ST | 0.5 | 0.428 | 0.072 |
| ผู้ประเมินคนที่ 1 | MINSH | 0.5 | 0.33 | 0.17 |
| ผู้ประเมินคนที่ 1 | LMS | 0.16 | 0.33 | -0.17 |
| ผู้ประเมินคนที่ 2 | HHTP | 0.2 | 0.6 | -0.4 |
| ผู้ประเมินคนที่ 2 | PTT | 0.6 | 0.66 | -0.06 |
| ผู้ประเมินคนที่ 2 | CINCOUT | 0.66 | 0.5 | 0.16 |
| ผู้ประเมินคนที่ 2 | LMS | 0.83 | 1 | -0.17 |
| ผู้ประเมินคนที่ 2 | LEAP | 0.54 | 0.45 | 0.09 |
| ผู้ประเมินคนที่ 3 | PTT | 0.4 | 0.46 | -0.06 |
| ผู้ประเมินคนที่ 3 | EATD | 0.5 | 0.4 | 0.1 |
| ผู้ประเมินคนที่ 3 | ST | 0.64 | 0.57 | 0.07 |
| ผู้ประเมินคนที่ 3 | MINSH | 0.5 | 0.33 | 0.17 |
| ผู้ประเมินคนที่ 3 | LMS | 0.33 | 0 | 0.33 |
| ผู้ประเมินคนที่ 4 | PTT | 0.53 | 0.6 | -0.07 |
| ผู้ประเมินคนที่ 4 | ST | 0.71 | 0.57 | 0.14 |
| ผู้ประเมินคนที่ 4 | MINSH | 0.5 | 0.33 | 0.17 |
| ผู้ประเมินคนที่ 5 | PTT | 0.2 | 0.33 | -0.13 |
| ผู้ประเมินคนที่ 5 | KAPP | 0.58 | 0.65 | -0.07 |
| ผู้ประเมินคนที่ 5 | ST | 0.71 | 0.57 | 0.14 |
| ผู้ประเมินคนที่ 5 | CINOUT | 0.76 | 0.66 | 0.1 |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

| | | | | |
|---|---|---|---|---|
| MMRE | | 0.5175 | 0.4884 | |

**การทดสอบข้อมูลกลุ่มแรกคือข้อมูลที่มีการเปลี่ยนแปลงค่าความคลาดเคลื่อน จำนวน 20 ข้อมูล**

**การทดสอบความเป็นการกระจายแบบปกติ (Normality Test)**

กำหนดสมมติฐานดังนี้

$H_0$ : ข้อมูลมีการกระจายแบบปกติ

$H_1$ : ข้อมูลไม่มีการกระจายแบบปกติ

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบความเป็นการกระจายแบบปกติ มีผลวิเคราะห์ดังตารางที่ 133

ตารางที่ 133 การทดสอบความเป็นการกระจายแบบปกติ

| | Kolmogorov-Sminov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|---|
| | Static | df | p-value | Static | df | p-value |
| MRE ก่อนการเปลี่ยนแปลงค่าประเมิน | 0.213 | 20 | 0.018 | 0.943 | 20 | 0.278 |
| MRE หลังการเปลี่ยนแปลงค่าประเมิน | 0.166 | 20 | 0.153 | 0.928 | 20 | 0.143 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากการทดสอบความเป็นการกระจายแบบปกติ พบว่าค่า p-value มีค่ามากกว่า 0.05 จึงทำให้ยอมรับ $H_0$ ผู้วิจัยจึงทำการทดสอบสมมติฐานด้วยวิธี paired t-test

สถิติการทดสอบ :การทดสอบ paired t-test

กำหนดให้

$H_0$ คือ ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินมีค่า**ไม่แตกต่าง**จากค่าประเมินหลังเปลี่ยนแปลงการประเมิน

$H_1$ คือ ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินมีค่า**แตกต่าง**จากค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

สามารถเขียนสมมติฐานการทดลองได้ดังนี้

$$H_0: \mu_1 = \mu_2$$

$$H_1: \mu_1 \neq \mu_2$$

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 134

ตารางที่ 134 การทดสอบ paired t-test ของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินและหลังเปลี่ยนแปลงค่าประเมิน

| | N | Mean | Std. Deviation | t | p-value |
|---|---|---|---|---|---|
| ก่อนการเปลี่ยนแปลงค่าประเมิน | 20 | 0.5175 | 0.18598 | 0.777 | 0.447 |
| หลังจากเปลี่ยนแปลงค่าประเมิน | 20 | 0.4884 | 0.20125 | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 134 ค่า p-value มากกว่า 0.05 จึงยอมรับ $H_0$ นั่นคือ หากผู้ประเมินได้เห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี (Algorithmic Model) และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินหลังการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำไม่แตกต่าง** จากค่าประเมินก่อนการเปลี่ยนแปลงการประเมินที่ระดับนัยสำคัญ 0.05

**การทดสอบข้อมูลกลุ่มที่สองคือข้อมูลทั้งหมดทั้งที่เปลี่ยนแปลงค่าความคลาดเคลื่อนและไม่เปลี่ยนแปลงค่าความคลาดเคลื่อน จำนวน 50 ข้อมูล**

**การทดสอบความเป็นการกระจายแบบปกติ (Normality Test)**

กำหนดสมมติฐานดังนี้

$H_0$ : ข้อมูลมีการกระจายแบบปกติ

$H_1$ : ข้อมูลไม่มีการกระจายแบบปกติ

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบความเป็นการกระจายแบบปกติ มีผลวิเคราะห์ดังตารางที่ 135

ตารางที่ 135 การทดสอบความเป็นการกระจายแบบปกติ

| | Kolmogorov-Sminov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|---|
| | Static | df | p-value | Static | df | p-value |
| MRE ก่อนการเปลี่ยนแปลงค่าประเมิน | 0.102 | 50 | 0.2 | 0.976 | 50 | 0.404 |
| MRE หลังการเปลี่ยนแปลงค่าประเมิน | 0.102 | 50 | 0.2 | 0.959 | 50 | 0.084 |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จากการทดสอบความเป็นการกระจายแบบปกติ พบว่าค่า p-value มีค่ามากกว่า 0.05 จึงทำให้ยอมรับ $H_0$ ผู้วิจัยจึงทำการทดสอบสมมติฐานด้วยวิธี paired t-test

สถิติการทดสอบ :การทดสอบ paired t-test

กำหนดให้

$H_0$ คือ ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินมีค่า**ไม่แตกต่าง**จากค่าประเมินหลังเปลี่ยนแปลงการประเมิน

$H_1$ คือ ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินมีค่า**แตกต่าง**จากค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

สามารถเขียนสมมติฐานการทดลองได้ดังนี้

$$H_0: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

ผู้วิจัยใช้เครื่องมือ The IBM SPSS Software สำหรับการทดสอบสมมติฐานทางสถิติ มีผลการวิเคราะห์ดังตารางที่ 136

ตารางที่ 136 การทดสอบ paired t-test ของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินและหลังเปลี่ยนแปลงค่าประเมิน

| | N | Mean | Std. Deviation | t | p-value |
|---|---|---|---|---|---|
| ก่อนการเปลี่ยนแปลงค่าประเมิน | 50 | 0.4803 | 0.20994 | 0.782 | 0.438 |
| หลังจากเปลี่ยนแปลงค่าประเมิน | 50 | 0.4687 | 0.21375 | | |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 136 ค่า p-value มากกว่า 0.05 จึงยอมรับ $H_0$ นั่นคือ หากผู้ประเมินได้เห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี (Algorithmic Model) และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินหลังการเปลี่ยนแปลงการประเมินจะมี**ค่าความแม่นยำไม่แตกต่าง** จากค่าประเมินก่อนการเปลี่ยนแปลงการประเมินที่ระดับนัยสำคัญ 0.05

**สรุปผลการทดสอบสมมติฐานที่ 2**

ผู้วิจัยได้แสดงค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีภายหลังการประเมินครั้งแรกของผู้ประเมิน พบว่า 40% ของจำนวนครั้งที่ประเมินทั้งหมด ที่ผู้ประเมินเปลี่ยนแปลงค่าประเมิน

ผู้วิจัยจึงได้ทดสอบสมมติฐาน 2.1 ที่ว่าภายหลังจากการเห็นค่าประเมินด้วยวิธีตัวแบบขั้นตอนวิธีแล้ว ผู้ประเมินมีโอกาสเปลี่ยนแปลงค่าประเมินเท่ากับ 50% หรือ ครึ่งหนึ่งของการประเมินทั้งหมด การทดสอบสมมติฐานที่ 2.1 ผู้วิจัยใช้การทดสอบ z-test มีผลการทดสอบสมมติฐาน ยอมรับ $H_0$ ด้วยค่า p-value เท่ากับ 0.135

การทดสอบสมมติฐานที่ 2.2 ที่ว่าหากผู้ประเมินได้เห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินหลังการเปลี่ยนแปลงการประเมินจะมีค่าความแม่นยำไม่แตกต่าง จากค่าประเมินก่อนการเปลี่ยนแปลงค่าประเมิน การทดสอบสมมติฐานที่ 2.2 ผู้วิจัยใช้การpaired t-test ผลการทดสอบสรุปได้ว่า ยอมรับ $H_0$ ด้วยค่า p-value เท่ากับ 0.447 สำหรับ 20 ข้อมูล และ 0.438 สำหรับ 50 ข้อมูล

**การศึกษาเพิ่มเติมเกี่ยวกับคุณสมบัติของผู้ประเมิน**

ผู้วิจัยต้องการศึกษาลักษณะของผู้ประเมินที่จะส่งผลต่อความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ การสรุปผลการวิเคราะห์ข้อมูลลักษณะของผู้ประเมินที่ผู้วิจัยเลือกศึกษา มีดังต่อไปนี้

1. จำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์

ผู้วิจัยทำการเปรียบเทียบความแม่นยำของผู้ประเมินทั้ง 5 ท่าน ประกอบกับจำนวนปีประสบการณ์ในการพัฒนาหรือมีส่วนร่วมในโครงการซอฟต์แวร์ดังตารางที่ 137

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 137 ข้อมูลจำนวนปีประสบการณ์ของผู้ประเมิน ตำแหน่งงานล่าสุด ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยและอันดับความแม่นยำ

| ผู้ประเมิน | จำนวนปีประสบการณ์ทั้งหมด | ตำแหน่งงานล่าสุด | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย | อันดับความแม่นยำ |
|---|---|---|---|---|
| คนที่ 1 | 17 | Project Manager | 0.4430 | 2 |
| คนที่ 2 | 13 | Assistant Manager | 0.4825 | 3 |
| คนที่ 3 | 11 | Project Manager | 0.4120 | 1 |
| คนที่ 4 | 16 | Project Manager | 0.5068 | 4 |
| คนที่ 5 | 10 | System Analyst | 0.5127 | 5 |

จากตารางที่ 137 ผู้ประเมินที่มีจำนวนปีประสบการณ์มากที่สุดคือ ผู้ประเมินคนที่ 1 มีจำนวนปีประสบการณ์ 17 ปี แต่มีความแม่นยำในการประเมินเป็นอันดับ 2 รองจากผู้ประเมินคนที่ 3 ซึ่งมี จำนวนปีประสบการณ์ 11 ปี ในขณะเดียวกัน ผู้ประเมินคนที่ 4 มีจำนวนปีประสบการณ์รองลงมาเป็นอันดับ 2 แต่มีความแม่นยำในการประเมินเป็นอันดับ 4 และด้วยจำนวนข้อมูลในการทดลองมีไม่มากนัก ด้วยเหตุนี้ผู้วิจัยจึงไม่สามารถสรุปได้แน่ชัดว่าจำนวนปีประสบการณ์มีผลต่อความแม่นยำในการประเมินหรือไม่

2. จำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม

ผู้วิจัยทำการเปรียบเทียบความแม่นยำของผู้ประเมินทั้ง 5 คน ประกอบกับจำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม ดังตารางที่ 138

ตารางที่ 138 ข้อมูลจำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม และค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย

| ผู้ประเมิน | จำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย | อันดับความแม่นยำ |
|---|---|---|---|
| คนที่ 1 | 40 | 0.4430 | 2 |
| คนที่ 2 | 5 | 0.4825 | 3 |
| คนที่ 3 | 10 | 0.4120 | 1 |
| คนที่ 4 | 20 | 0.5068 | 4 |
| คนที่ 5 | 10 | 0.5127 | 5 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 138 ผู้ประเมินที่มีจำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วมมากที่สุดคือ ผู้ประเมินคนที่ 1 แต่มีความแม่นยำในการประเมินเป็นอันดับ 2 รองจากผู้ประเมินคนที่ 3 ส่วนผู้ประเมินที่มีจำนวนโครงการซอฟต์แวร์ที่น้อยที่สุดคือผู้ประเมินคนที่ 2 แต่มีอันดับความแม่นยำเป็นอันดับ 3 รองจากผู้ประเมินคนที่ 3 และ 1 และด้วยจำนวนข้อมูลที่มีไม่มากนักด้วยเหตุนี้ผู้วิจัยจึงไม่สามารถสรุปได้ว่าจำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วมมีผลต่อความแม่นยำในการประเมินหรือไม่

3. ตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์

ผู้วิจัยทำการเปรียบเทียบความแม่นยำของผู้ประเมินทั้ง 5 คน ประกอบกับตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์ ดังตารางที่ 139



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 139 ข้อมูลตำแหน่งงานของผู้ประเมิน ค่าความคาดเคลื่อนสัมพัทธ์เฉลี่ย และอันดับความแม่นยำ

| ผู้ประเมิน | ตำแหน่งงาน | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย | อันดับความแม่นยำ |
|---|---|---|---|
| คนที่ 1 | Developer, System Analyst, Project Manager | 0.4430 | 2 |
| คนที่ 2 | Programmer, Software Engineer, System Analyst, Assistant Manager | 0.4825 | 3 |
| คนที่ 3 | Software Manager, Project Manager | 0.4120 | 1 |
| คนที่ 4 | Programmer, System Analyst, Project Manager | 0.5068 | 4 |
| คนที่ 5 | Application Development, Analyst, IT Operation, System Analyst | 0.5127 | 5 |

จากตารางที่ 139 ข้อมูลตำแหน่งงานของผู้ประเมินมีความหลากหลาย 4 ใน 5 ของผู้ประเมินเคยทำงานเป็นผู้พัฒนาซอฟต์แวร์มาก่อน แต่มีความแม่นยำน้อยกว่าผู้ประเมินคนที่ 3 ที่ทำงานในตำแหน่งระดับ Manager และ 4 ใน 5 ของผู้ประเมินมีตำแหน่ง Project Manager รวมถึง Assistant Manager มีความสามารถในการประเมินมากกว่าผู้ประเมินคนที่ 5 ที่ไม่เคยทำงานในตำแหน่ง Manager จากข้อมูลดังกล่าวผู้วิจัยมีความเห็นว่า ตำแหน่ง Manager อาจมีผลต่อความแม่นยำในการประเมิน ทั้งนี้ข้อมูลจากตารางที่ 139 ยังไม่เพียงพอต่อการสรุปผลว่าตำแหน่ง Project Manager มีผลต่อความแม่นยำในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์หรือไม่

4. ภาษาการเขียนโปรแกรมที่ถนัด

ผู้วิจัยทำการเปรียบเทียบความแม่นยำของผู้ประเมินทั้ง 5 คน ประกอบกับภาษาการเขียนโปรแกรมที่ถนัด แสดงในตารางที่ 140

ตารางที่ 140 ข้อมูลภาษาการเขียนโปรแกรม ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย และอันดับความแม่นยำของผู้ประเมินทั้ง 5 คน

| ผู้ประเมิน | ภาษาการเขียนโปรแกรม | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย | อันดับความแม่นยำ |
|---|---|---|---|
| คนที่ 1 | C#, Java, Javascript | 0.4430 | 2 |
| คนที่ 2 | VB.Net, C#, VBA | 0.4825 | 3 |
| คนที่ 3 | Java, Javascript, C# | 0.4120 | 1 |
| คนที่ 4 | PHP, Java, Javascript | 0.5068 | 4 |
| คนที่ 5 | C#, Javascript, C | 0.5127 | 5 |

จากตารางที่ 140 ภาษาการเขียนโปรแกรมของผู้ประเมินทั้ง 5 คน ล้วนแต่มีความถนัดในการเขียนโปรแกรมเชิงวัตถุทั้งสิ้น ด้วยเหตุนี้ผู้วิจัยจึงไม่สามารถสรุปได้ว่าภาษาการเขียนโปรแกรมมีผลต่อความแม่นยำในการประเมินค่าความพยายามหรือไม่ เนื่องจากข้อมูลไม่เพียงพอที่จะสรุปผล

**4.8 สรุปการวิเคราะห์ข้อมูล**

**การเปรียบเทียบความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์**

**คำถามวิจัย** : การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ภายในสภาวะแวดล้อมที่ผู้เชี่ยวชาญเป็นคนไทยด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าหรือน้อยกว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี

**สมมติฐานที่ 1** : ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีขั้นตอนวิธี มากกว่าหรือเท่ากับ ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

**ผลการทดสอบสมมติฐาน** : ไม่สามารถสรุปได้ว่าวิธีการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญแม่นยำมากกว่า เท่ากับ หรือน้อยกว่าวิธีตัวแบบขั้นตอนวิธี

**ค่าประเมินที่ได้จากวิธีตัวแบบขั้นตอนวิธีมาประกอบการประเมินด้วยวิธีการตัดสินใจของผู้เชี่ยวชาญ**

Chulalongkorn University

**คำถามวิจัย** : ค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญจะช่วยให้การประเมินแม่นยำขึ้นหรือไม่

**สมมติฐานที่ 2.1** : ค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี ทำให้ผู้ประเมินเปลี่ยนแปลงค่าประเมินเท่ากับครึ่งหนึ่งของจำนวนการทดลองทั้งหมด (จำนวนการทดลองทั้งหมด 50 การทดลอง) จากการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วย วิธีการตัดสินใจของผู้เชี่ยวชาญ

**ผลการทดสอบสมมติฐาน 2.1** : ค่าเฉลี่ยของจำนวนครั้งที่ผู้ประเมินตัดสินใจเปลี่ยนแปลงค่าประเมินมีค่าเท่ากับ 0.5 หรือ ครึ่งหนึ่งของจำนวนครั้งที่ประเมินทั้งหมด

**สมมติฐานที่ 2.2 :** หากผู้ประเมินเห็นผลลัพธ์จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงการประเมินจะมีค่าความแม่นยำไม่แตกต่างจาก ค่าประเมินหลังการเปลี่ยนแปลงการประเมิน

**ผลการทดสอบสมมติฐาน 2.2** : หากผู้ประเมินได้เห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี และได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินหลังจากเปลี่ยนแปลงค่าประเมินจะมีค่าความแม่นยำไม่แตกต่าง จากค่าประเมินก่อนการเปลี่ยนแปลงค่าประเมิน

**การศึกษาเพิ่มเติมเกี่ยวกับคุณสมบัติของผู้ประเมิน**

1. จำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์

ผู้วิจัยไม่สามารถสรุปได้แน่ชัดว่าจำนวนปีประสบการณ์มีผลต่อความแม่นยำในการประเมินหรือไม่

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2. จำนวนโครงการที่เคยพัฒนาหรือมีส่วนร่วม

ผู้วิจัยไม่สามารถสรุปได้ว่าจำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วมมีผลต่อความแม่นยำในการประเมินหรือไม่

3. ตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์

ผู้วิจัยไม่สามารถสรุปได้ว่าตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์มีผลต่อความแม่นยำในการประเมินหรือไม่

4. ภาษาการเขียนโปรแกรม

ผู้วิจัยไม่สามารถสรุปได้ว่าภาษาการเขียนโปรแกรมมีผลต่อความแม่นยำในการประเมินหรือไม่

# บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ

## 5.1 สรุปผลการวิจัย

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ส่วนใหญ่ไม่แม่นยำ 31.1% ของโครงการซอฟต์แวร์ ถูกยกเลิกก่อนเสร็จโครงการและ 52.7% มีต้นทุนในการพัฒนาโครงการสูงกว่าที่ประมาณการไว้ถึง 189% (The Standish Group, 2014) ผู้วิจัยได้ทบทวนวรรณกรรมที่เกี่ยวข้อง พบว่า วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ที่ได้รับความนิยมมากมี 2 วิธี คือ วิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี ซึ่งตัวแบบขั้นตอนวิธีที่ได้รับความนิยมคือ ตัวแบบโคโคโม่ ตัวแบบฟังก์ชันพอยต์ และตัวแบบยูสเคสพอยต์ นอกจากนี้ผู้วิจัยยังศึกษางานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบความแม่นยำของสองวิธีการประเมินข้างต้น พบว่า ยังไม่สามารถสรุปได้แน่ชัดถึงวิธีการประเมินที่แม่นยำที่สุด เนื่องจากปัจจัยหลายอย่าง อาทิ เช่น ผลการเปรียบเทียบความแม่นยำ มี 4 งานวิจัยที่วิธีการตัดสินของผู้เชี่ยวชาญแม่นยำมากกว่าวิธีตัวแบบขั้นตอนวิธีและอีก 2 งานวิจัยที่วิธีตัวแบบขั้นตอนวิธีมีความแม่นยำมากกว่าวิธีการตัดสินของผู้เชี่ยวชาญ นอกจากนี้การทดลองเปรียบเทียบความแม่นยำระหว่างสองวิธีข้างต้นของงานวิจัยที่ผ่านมาใช้ข้อมูลทุติยภูมิของ Kemerer (1987) ซึ่งข้อมูลดังกล่าวไม่มีหน้าที่การทำงานของซอฟต์แวร์ รวมไปถึงภาษาที่ใช้ในการพัฒนาไม่ใช่ภาษาที่นิยมใช้ในปัจจุบัน ผู้วิจัยได้กำหนดคำถามวิจัยและสมมติฐานดังนี้

คำถามวิจัยที่ 1 การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ในสภาวะแวดล้อมที่ผู้เชี่ยวชาญเป็นคนไทยด้วยวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าหรือน้อยกว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีหรือไม่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้วิจัยได้กำหนดสมมติฐานจากคำถามวิจัยดังต่อไปนี้

สมมติฐานที่ 1 ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธี มากกว่าหรือเท่ากับ ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

คำถามวิจัยที่ 2 การนำค่าประเมินที่ได้จากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มาประกอบการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญจะช่วยให้การประเมินแม่นยำขึ้นหรือไม่

ผู้วิจัยได้กำหนดสมมติฐานจากคำถามวิจัยดังต่อไปนี้

สมมติฐานที่ 2.1 ค่าประเมินที่ได้จากการใช้วิธีตัวแบบขั้นตอนวิธี ทำให้ผู้เชี่ยวชาญเปลี่ยนแปลงค่าประเมินมากกว่าครึ่งหนึ่งของจำนวนการทดลองทั้งหมด (จำนวนครั้งของการประมาณค่า 50 ครั้ง) จากการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

สมมติฐานที่ 2.2 หากผู้เชี่ยวชาญได้เห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีและได้เปลี่ยนแปลงค่าประเมินแล้ว ค่าประเมินก่อนการเปลี่ยนแปลงค่าประเมินจะมีค่าความแม่นยำแตกต่างจากค่าประเมินหลังการเปลี่ยนแปลงค่าประเมิน

ผู้วิจัยได้ออกแบบและดำเนินการทดลองเพื่อตอบคำถามวิจัยและทดสอบสมมติฐานดังกล่าว โดยผู้วิจัยจะหาโครงการซอฟต์แวร์จำนวน 10 โครงการ โครงการซอฟต์แวร์เหล่านี้มาจากบริษัทที่เกี่ยวข้องกับธุรกิจการพัฒนาซอฟต์แวร์ และโครงการซอฟต์แวร์ที่ผู้วิจัยเคยพัฒนาหรือมีส่วนร่วม โครงการซอฟต์แวร์เหล่านี้เป็นโครงการที่เกิดขึ้นจริง ข้อมูลของโครงการซอฟต์แวร์มาจากการสัมภาษณ์และตรวจสอบความสอดคล้องกับระบบควบคุมเวอร์ชันซอฟต์แวร์ (Version Control) ทำให้ผู้วิจัยทราบระยะเวลาในการทำซอฟต์แวร์ของแต่ละโครงการ ผู้วิจัยประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีตัวแบบขั้นตอนวิธีและหาค่าความคลาดเคลื่อนสัมพัทธ์ ทั้ง 10 โครงการ ผู้วิจัยได้ติดต่อผู้เชี่ยวชาญจำนวน 5 คน มีประสบการณ์ทำงานในธุรกิจที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ไม่น้อยกว่า 10 ปี มีตำแหน่งงานในระดับ System Analyst หรือ Manager ขึ้นไป มีความคุ้นเคยกับการพัฒนาซอฟต์แวร์ด้วยภาษาการเขียนโปรแกรมที่นิยมในปัจจุบัน อาทิเช่น JAVA, C# ในวันทดลองผู้วิจัยได้อธิบายความต้องการด้านหน้าที่และไม่ใช่ด้านหน้าที่ของโครงการซอฟต์แวร์ (Functional and Non-Functional Requirement) ผู้ประเมินจะไม่พูดคุยกันในระหว่างที่ประเมิน เมื่อประเมินเสร็จผู้วิจัยจะแสดงค่าประเมินที่ได้จากวิธีตัวแบบขั้นตอนวิธีทั้ง 3 เพื่อให้ผู้ประเมินทำการประเมินใหม่อีกครั้งหนึ่ง ผู้วิจัยบันทึกค่าประเมินของผู้ประเมินทุกครั้งหลังจากที่ผู้ประเมินทำการประเมินจนเสร็จ เมื่อผู้ประเมินทั้ง 5 คนประเมินโครงการซอฟต์แวร์จนเสร็จ ผู้วิจัยจะหาค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย เพื่อหาลำดับของผู้ประเมินสำหรับการมอบของรางวัล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 141 สรุปผลค่าความคลาดเคลื่อนเฉลี่ยของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี ฟังก์ชันพอยต์ โคโคโม่ ยูสเคสพอยต์ และวิธีการตัดสินของผู้เชี่ยวชาญ

| วิธีการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ | จำนวนโครงการ | ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย |
|---|---|---|
| ฟังก์ชันพอยต์ | 10 | 1.85 |
| โคโคโม่ | 10 | 4.40 |
| ยูสเคสพอยต์ | 10 | 0.4 |
| การตัดสินของผู้เชี่ยวชาญ 5 คน | 10 x 5 = 50 | 0.47 |

การทดสอบสมมติฐานที่ 1 : ผลการทดลองแสดงในตารางที่ 141 พบว่าตัวแบบยูสเคสพอยต์มีความคลาดเคลื่อนสัมพัทธ์เฉลี่ยน้อยที่สุด รองลงมาคือ วิธีการตัดสินของผู้เชี่ยวชาญ ฟังก์ชันพอยต์ และ โคโคโม่ตามลำดับ ค่าความคลาดเคลื่อนสัมพัทธ์ระหว่างตัวแบบยูสเคสพอยต์และวิธีการตัดสินของผู้เชี่ยวชาญมีค่าใกล้เคียงกันมาก ผู้วิจัยจึงได้ทดสอบสมมติฐานทางสถิติด้วยการใช้การทอสอบ Kruskal-Wallis พบว่าค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างตัวแบบ โคโคโม่ ยูสเคสพอยต์ ฟังก์ชันพอยต์ และวิธีการตัดสินของผู้เชี่ยวชาญมีความแตกต่างกันอย่างมีนัยสำคัญ ผู้วิจัยจึงทำการทดสอบในแต่ละคู่ด้วยวิธี Mann-Whitney ได้ผลการทดสอบสมมติฐานดังตารางที่ 142

ตารางที่ 142 ความแตกต่างของค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยของวิธีการประเมิน 6 คู่

| การเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมิน 2 วิธี | การเปรียบเทียบค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ |
|---|---|
| วิธีการตัดสินของผู้เชี่ยวชาญ กับ ตัวแบบยูสเคสพอยต์ | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการตัดสินของผู้เชี่ยวชาญมากกว่าตัวแบบยูสเคสพอยต์ |
| ตัวแบบโคโคโม่ กับ วิธีการตัดสินของผู้เชี่ยวชาญ | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการตัดสินของผู้เชี่ยวชาญน้อยกว่าตัวแบบโคโคโม่ |
| ตัวแบบฟังก์ชันพอยต์ กับ วิธีการตัดสินของผู้เชี่ยวชาญ | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบฟังก์ชันพอยต์ ไม่มากกว่า วิธีการตัดสินของผู้เชี่ยวชาญ |
| ตัวแบบโคโคโม่ กับ ตัวแบบยูสเคสพอยต์ | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่มากกว่าตัวแบบยูสเคสพอยต์ |
| ตัวแบบยูสเคสพอยต์ กับ ตัวแบบฟังก์ชันพอยต์ | ค่ากลางของค่าความคลาดเคลื่อนตัวแบบยูสเคส |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

| การเปรียบเทียบความแตกต่างของค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของวิธีการประเมิน 2 วิธี | การเปรียบเทียบค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ |
|---|---|
| | พอยต์ไม่มากกว่าตัวแบบฟังก์ชันพอยต์ |
| ตัวแบบโคโคโม่ กับ ตัวแบบฟังก์ชันพอยต์ | ค่ากลางของค่าความคลาดเคลื่อนสัมพัทธ์ของตัวแบบโคโคโม่ มากกว่า ตัวแบบฟังก์ชันพอยต์ |

จากผลการทดสอบสมมติฐานแสดงให้เห็นว่า ความแม่นยำของตัวแบบยูสเคสพอยต์มีความแม่นยำมากกว่าวิธีการตัดสินของผู้เชี่ยวชาญ แต่ความแม่นยำของวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำมากกว่าตัวแบบฟังก์ชันพอยต์และตัวแบบโคโคโม่ ด้วยเหตุนี้ยังไม่สามารถสรุปได้ว่าการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธีตัวแบบขั้นตอนวิธีและวิธีการตัดสินของผู้เชี่ยวชาญ วิธีใดแม่นยำกว่ากัน บางคู่วิธีการตัดสินของผู้เชี่ยวชาญแม่นยำกว่าและบางคู่วิธีตัวแบบขั้นตอนวิธีแม่นยำกว่า ซึ่งสอดคล้องกับการโต้เถียงของสองผู้เชี่ยวชาญระหว่าง Boehm และ Jorgensen (Jorgensen & Boehm, 2009) นอกจากนี้สามารถอภิปรายผลการทดลองได้ว่า ในการประเมินค่าความพยายามสำหรับการพัฒนาซอฟต์แวร์ แม้ตัวแบบยูสเคสพอยต์จะมีความแม่นยำในการประเมินมากที่สุด เจ้าของโครงการสามารถใช้วิธีการตัดสินของผู้เชี่ยวชาญได้ เนื่องจากมีความแม่นยำน้อยกว่าตัวแบบยูสเคสพอยต์ เพียง 7%

การทดสอบสมมติฐานที่ 2.1 : ผู้วิจัยทำการทดสอบสมมติฐานทางสถิติด้วยวิธีการทดสอบ z-test พบว่าผู้ประเมินมีโอกาสตัดสินใจเปลี่ยนแปลงค่าประเมิน 50% ของจำนวนครั้งที่ประเมินทั้งหมด เมื่อผู้ประเมินเห็นค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี นอกจากนี้สามารถอธิปรายผลการทดลองได้ว่า ค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีมีผลต่อการตัดสินใจเปลี่ยนแปลงค่าประเมินของผู้ประเมิน 50% หรือครึ่งหนึ่งของจำนวนครั้งที่ประเมินทั้งหมด ซึ่งผู้ประเมินรู้จักวิธีตัวแบบขั้นตอนวิธี จึงกล่าวได้ว่า ค่าประเมินจากวิธีตัวแบบขั้นตอนวิธีมีผลกระทบต่อการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญ

การทดสอบสมมติฐานที่ 2.2 : ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยก่อนการเปลี่ยนแปลงค่าประเมินเมื่อเห็นค่าประเมินจากวิธีตัวแบบขั้นตอนวิธี ของผู้ประเมินทั้ง 5 คนมีค่าเท่ากับ 0.5175 ในขณะที่ ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยภายหลังการเปลี่ยนแปลงค่าประเมินอยู่ที่ 0.4884 จากตัวเลขดังกล่าวแสดงให้เห็นว่าความแม่นยำของการประเมินภายหลังจากเห็นค่าประเมินด้วยวิธีตัวแบบขั้นตอนวิธีมีความแม่นยำขึ้นเล็กน้อย ผู้วิจัยจึงได้ทำการทดสอบสมมติฐานทางสถิติโดยใช้วิธีการทดสอบ paired t-test พบว่าไม่มีความแตกต่างกันอย่างมีนัยสำคัญทางสถิติ ของผลต่างระหว่าง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ความแม่นยำก่อนการเปลี่ยนแปลงค่าประเมินและหลังการเปลี่ยนแปลงค่าประเมิน ผู้วิจัยสามารถอภิปรายผลการทดลองได้ว่า การนำค่าประเมินจากวิธีตัวแบบขั้นตอนวิธีมาประกอบการตัดสินใจในการประเมินของผู้เชี่ยวชาญ นอกจากจะเสียเวลาในการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีแล้ว ยังไม่ได้ช่วยให้การประเมินแม่นยำขึ้นอย่างมีนัยสำคัญ

นอกจากนี้ผู้วิจัยได้ศึกษาลักษณะเพิ่มเติมของผู้ประเมินที่จะส่งผลต่อความแม่นยำในการประเมิน ลักษณะของผู้ประเมินที่ผู้วิจัยเลือกศึกษามีดังต่อไปนี้

1. จำนวนปีประสบการณ์ในการพัฒนาซอฟต์แวร์
2. จำนวนโครงการซอฟต์แวร์ที่เคยพัฒนาหรือมีส่วนร่วม
3. ตำแหน่งหน้าที่ในบริษัทซอฟต์แวร์
4. ภาษาการเขียนโปรแกรมที่ถนัด

จากการศึกษาลักษณะของผู้ทั้ง 4 หัวข้อข้างต้น ไม่สามารถสรุปได้ว่าคุณสมบัติดังกล่าวมีผลต่อความแม่นยำในการพัฒนาซอฟต์แวร์หรือไม่ เนื่องจากข้อมูลในการทดลองมีไม่เพียงพอสำหรับการสรุปผล

## 5.2 การนำงานวิจัยไปใช้

องค์ความรู้จากงานวิจัยสามารถนำไปใช้ในการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ดังต่อไปนี้

1. การประเมินด้วยตัวแบบยูสเคสพอยต์มีความแม่นยำมากที่สุด สำหรับการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ แต่หากผู้บริการโครงการไม่สะดวกที่จะใช้ตัวแบบยูสเคสพอยต์ในการประมาณค่า สามารถใช้วิธีการตัดสินของผู้เชี่ยวชาญแทนได้ เนื่องจากวิธีการตัดสินของผู้เชี่ยวชาญมีความแม่นยำน้อยกว่าวิธีตัวแบบยูสเคสพอยต์ เพียง 7%
2. การนำค่าประเมินจากการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีมาประกอบการประเมินของผู้เชี่ยวชาญ มีผลให้ผู้เชี่ยวชาญตัดสินใจเปลี่ยนแปลงค่าประเมินแต่ไม่มีผลต่อความแม่นยำของการประเมินอย่างมีนัยสำคัญ ผู้บริหารโครงการจึงไม่มีความจำเป็นต้องนำค่าประเมินที่ได้จากวิธีตัวแบบขั้นตอนวิธีมาประกอบการประเมินของผู้เชี่ยวชาญ

## 5.3 ข้อจำกัดของงานวิจัยและข้อเสนอแนะ

งานวิจัยนี้จะสมบูรณ์ขึ้น ถ้าเพิ่มเติมหัวข้อต่างๆ ดังนี้

1. การเพิ่มตัวแบบขั้นตอนวิธีที่ใช้ในการทดลอง ในงานวิจัยนี้ ผู้วิจัยเลือกใช้ตัวแบบวิธีที่ได้รับความนิยมเพียง 3 ตัวแบบ คือ ยูสเคสพอยต์ โคโคโม่ และ ฟังก์ชันพอยต์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2. การเพิ่มข้อมูลนำเข้าก่อนการประเมิน ในงานวิจัยนี้ จำลองเหตุการณ์ที่มีข้อมูลเฉพาะช่วงเริ่มต้นของการพัฒนาโครงการซอฟต์แวร์ จึงมีเฉพาะความต้องการด้านหน้าที่และไม่ใช่ด้านหน้าที่ การเพิ่มเติมข้อมูลอื่นๆ นอกเหนือจากที่มีในช่วงเริ่มต้นของการพัฒนาโครงการซอฟต์แวร์ อาทิเช่น ยูสเคสไดอะแกรม คลาสไดอะแกรม สถาปัตยกรรมของระบบ เพื่อประกอบการประเมินของผู้เชี่ยวชาญ จะช่วยให้งานวิจัยสมบูรณ์มากยิ่งขึ้น

3. การเพิ่มวิธีการประเมิน ในงานวิจัยนี้ ผู้วิจัยเลือกใช้วิธีการประเมินที่ได้รับความนิยม 2 วิธี มาเปรียบเทียบความแม่นยำ คือ วิธีการตัดสินของผู้เชี่ยวชาญและวิธีตัวแบบขั้นตอนวิธี การเพิ่มวิธีการประเมินในการทดลอง อาทิเช่น วิธีพาร์กินสัน จะช่วยให้งานวิจัยสมบูรณ์มากยิ่งขึ้น

4. เพิ่มผู้ประเมินและการตรวจสอบสำหรับการประเมินด้วยวิธีตัวแบบขั้นตอนวิธี ในงานวิจัยนี้ผู้วิจัยเป็นผู้ประเมินด้วยวิธีตัวแบบขั้นตอนวิธี มีการตรวจสอบโดยอาจารย์ที่ปรึกษา งานวิจัยจะสมบูรณ์มากยิ่งขึ้นหากมีผู้ประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธีเพิ่มขึ้น และมีผู้ตรวจสอบความถูกต้องมากขึ้น

5. การจับเวลาในการดำเนินการทดลอง ผู้วิจัยไม่ได้จับเวลาที่ใช้ในการประเมินของผู้ประเมินและไม่ได้จับเวลาการประเมินค่าความพยายามด้วยวิธีตัวแบบขั้นตอนวิธีของผู้เชี่ยวชาญ จึงไม่สามารถให้รายละเอียดและหลักฐานสนับสนุนได้ชัดเจนว่า เวลาที่ใช้ในการประเมินของผู้วิจัยในการประเมินด้วยวิธีตัวแบบขั้นตอนวิธีและผู้เชี่ยวชาญในการประเมินด้วยวิธีการตัดสินของผู้เชี่ยวชาญแตกต่างกันมากน้อยเพียงใด

6. การเพิ่มผู้เชี่ยวชาญและโครงการซอฟต์แวร์ที่ไม่ได้อยู่ในสภาวะแวดล้อมในประเทศไทย ในงานวิจัยนี้มีขอบเขตการดำเนินงานภายใต้สภาวะแวดล้อมในประเทศไทย งานวิจัยจะสมบูรณ์มากขึ้นหากผู้ประเมินไม่ได้มีเฉพาะผู้ประเมินที่เป็นคนไทย และเพิ่มโครงการซอฟต์แวร์ที่มีการพัฒนาในต่างประเทศ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บรรณานุกรม



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# ประวัติผู้เขียน

| | |
|---|---|
| **ชื่อ-สกุล** | ชานน จรัสสุทธิกุล |
| **วัน เดือน ปี เกิด** | 14 พฤศจิกายน 2536 |
| **สถานที่เกิด** | ขอนแก่น |
| **วุฒิการศึกษา** | สถานบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง |
| **ที่อยู่ปัจจุบัน** | 190/59 Coolhouse รัชดาภิเษก เขตห้วยขวาง |



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บรรณานุกรม


Adrian, H. L., Ricardo, C. P., & Angel, G. C. . (2013). Software Engineering Job Productivity. *International Journal of Software Engineering and Knowledge Engineering, 23*(3), 387-406.

Albrecht, A. J., & Gaffney, J E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering., 9*(6), 639-648.

Arnuphaptrairong, T. (2018). The State of Practice pf Software Cost Estimation: Evidence From Thai Software Firms. *Proceedings of the International MultiConference of Engineers and Computer Scientists.* Retrieved from http://www.iaeng.org/publication/IMECS2018/IMECS2018_pp684-689.pdf.

Arnuphaptrirong, T. (2013). Early Stage Software Effort Estimation Using Function Point Analysis:Empirical Evidence. *International MultiConference of Engineers and Computer Scientists, 2*(1).

Atkinson, K., & Shepperd, M. (1994). The use of function points to find cost analogies. *5th European Software Cost Modelling Meeting*. Retrieved from https://bura.brunel.ac.uk/handle/2438/1553.

Banerjee, G. (2001). Use Case Points – An Estimation Approach. Retrieved from http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/reporty/use_case_points.pdf.

Behrens, C. A. (1983). Measuring the productivity of Computer Systems Development Activities with Function Points. *IEEEE Transactions on Software Engineering, 9*(6), 648-652.

Boehm, B. W. (1981). *Software Engineering Economics*: Prentice Hall. 1 edition.

Fan, W., Xiaohu, Y.., & Xiaochum, Z. (2009). Extended Use Case Points Method For Software Cost Estimation. *International Conference on Computational Intelligence and Software Engineering*.

Fenton, N. E., & Pfleeger, S L. (1997). *Software Metrics : Arigorous and Practical Approach*: International Thomson Computer Press.


จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Grimstad, S., & Jorgensen, M. (2007). Inconsistency of judgment-based estimates of software development effort. *The Journal of System and Software., 80*(11), 1770-1777.

Group, T. S. (2014). *Chaos Report*. Retrieved from https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Institutes., P. M. (2004). *A Guide to the Project Management Body of Knowledge*. PMBOK Guide. Project Management Institute, Inc.

Jeffery R., R. M., & Wieczorek, L. (2001). Using Public Domain Metric to Estimate Software Development Effort. *In Proceeding of international Software Metric Symposium*, 16-27.

Jong, A. D. (1996). Information about COCOMO Calculation. Retrieved from https://arthurdejong.org/cocomo/cocomo.html.

Jorgensen, M. (2002). A Review of studies on expert estimation of software development effort. *The journal of Systems and Software, 7*(1-2), 37-60.

Jorgensen, M. (2005). Practical Guidelines for Expert-Judgment-Based Software Effort Estimation. *IEEE Software., 12*(3), 57-63.

Jorgensen, M. (2007). Forecasting of software development work effort: Evidence on expert judgement and formal models. *International Journal of Forecasting, 23*(3), 449-462.

Jorgensen, M., & Boehm, B W. (2009). Software Development Effort Estimation: Formal Models or Expert Judgment? *IEEE Computer Society*, 14-19.

Kang, S., Choi, O., & Baik, J. . (2010). Model-based Dynamic Cost Estimation and Tracking Method for Agile Software Development. *IEEE/ACIS International Conference on Computer and Information Science.*, 743-748.

Kemerer, C. F. (1987). An Empirical Validation of Software Cost Estimation Models. Communication of the ACM., *30*(5), 416-429.

Kishore, S., & Naik, R. (2001). *Software Requirement and Estimation.*: 1st Edition. Tata McGrew-Hill Publishing.

Kuster, R. J., Gunuchten*, M J I M, van., & Heemstra, F J. (1990). Are software cost-estimation models accurate? *Information and Software Technology, 32*(3), 187-190.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Lokan, C. J. (2000). An empirical analysis of function point adjustment factors. *Information and Software Technology., 42*(9), 649-659.

Macdonell, S. G., & Shepperd, M J. (2003). Combining techniques to optimize effort predictions in software project management. *The journal of Systems and Software, 66*(2), 91-98.

Management., Q. S. (Producer). (2017). Function Point Languages Table. Retrieved from http://www.qsm.com/resources/function-point-languages-table

Maxwell, K. D., & Forselius, P. (2000). Benchmarking software development productivity. *IEEE Software., 17*(1), 80-88.

Menguy, T. (2001). Concrete Estimation (size, effort, schedule). Retrieved from http://users.polytech.unice.fr/~hugues/GL/Projet/estimation.html.

Molokken M., J. M. (2003). A review of software surveys on software effort estimation. *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*

Moser, S., & Nierstrasz, O. (1996). The Effect of Object-Oriented Frameworks on Devloper Productivity. *IEEE Computer Society, 29*(9), 45-51.

Mukhopadhyay, T., Vicinanza, S S., & Prietula, M J. (1992). Examining the Feasibility of a Case-

Based Reasoning Model for Software Effort Estimation. *Journal MIS Quarterly, 16*(2), 155-171.

Ourada, G. L. (1991). *Software Cost Estimating Models: A Calibration, Validation, and Comparison.*

Pengelly, A. (1995). Performance of effort estimating techniques in current development environments. *Software Engineering Journal, 10*(5), 162-170.

Popli, R., & Chauhan, N. (2014). Cost and Effort Estimation in Agile Software Development. *International Conference on Reliability, Optimization and Information Technology – ICROIT 2014.*, 57-61.

Shekhar, S. (2016). Review of Various Software Cost Estimation Techniques. *International Journal of Computer Application., 141*(11), 31-41.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Solution., S. M. (2004). Function Point Analysis Counting Function Points Reference Guide. *IFPUG CPM Version 4.1*. Retrieved from https://people.eecs.ku.edu/~hossein/811/Papers/fpa-ref.pdf.

Tayyab, M. R., Usman, M., & Ahmad, W. (2017). A Machine Learning Based Model for Software Cost Estimation. *SAI intelligent Systems Conference, 16*(1), 402-414.

Vicinanza, S. S., Mukhopadhyay, T., & Prietula, M J. (1991). Software-Effort Estimation: An Exploratory Study of Expert Performance. *Information Systems Research, 2*(4), 243-314.

Westfall, W. (2009). The Certified Software Quality Engineer Handbook.

เจิมบุญธนะพงศ์, ข. ธ. แ. ป. (2558). การประมาณการต้นทุนค่าใช้จ่ายในการพัฒนาเนื้อหาบทเรียนออนไลน์สำหรับเด็กผิดปกติ โดยวิธีแบบจำลอง COCOMO – II ร่วมกับวิธีการของ Web Model. วารสารวิชานวัตกรรมสื่อสารสังคม, *1*(5), 67-75.

กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร. (2557). การประเมินราคาและขนาดซอฟต์แวร์ด้วยมาตรฐาน *COCOMO*. โครงการพัฒนามาตรฐานราคากลางและเกณฑ์การประเมินราคาซอฟต์แวร์

อานุภาพไตรรงค์, ถ. (2558). การประมาณขนาดและต้นทุนซอฟต์แวร์. กรุงเทพฯ: เคทีพี.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# ภาคผนวก ก

ตารางที่ 143 ค่า Function Point Language Table

| Language | QSM FP (Line/Function Point) |
|---|---|
| ABAP (SAP) | 28 |
| ASP | 51 |
| Assembler | 119 |
| COBAL | 61 |
| C | 97 |
| C++ | 50 |
| HTML | 34 |
| J2EE | 46 |
| Java | 53 |
| JavaScript | 47 |
| .Net | 57 |
| Oracle | 37 |
| Perl | 24 |
| PL/SQL | 37 |
| SQL | 21 |
| VB.Net | 52 |
| Visual Basic | 42 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# ภาคผนวก ข.

**ตัวอย่าง ความต้องการซอฟต์แวร์ด้านหน้าที่ (Functional Requirement)**

1.ระบบลงทะเบียนเรียนวิชาสำหรับผู้ใช้ประเภทนักศึกษา เพื่อให้มีสิทธิ์ในการเข้าถึงวิดีโอ ไฟล์สื่อการเรียนการสอน และกระดานสนทนาในรายวิชา

2. ระบบจะสามารถแสดงรายการรายวิชาที่ผู้ใช้ประเภทนักศึกษาผู้นั้นลงทะเบียนเรียนไว้ได้

3. ระบบสามารถแสดงรายชื่อนักศึกษาที่ลงทะเบียนเรียนในวิชาให้แก่ผู้ใช้ประเภทอาจารย์ที่เป็นผู้สอนของรายวิชา และผู้สอนร่วมได้

4. ระบบจะต้องรองรับการค้นหารายวิชาได้

การค้นหาสามารถกรองได้ด้วย

- ชื่อวิชา
- ชื่ออาจารย์ผู้สอน และผู้สอนร่วม
- ปีการศึกษาของรายวิชา
- คณะของรายวิชา
- สาขาของรายวิชา

5. ระบบจะรองรับการดูไฟล์วิดีโอที่อาจารย์อัพโหลดได้ โดยวิดีโอจะแสดงความละเอียดได้เท่ากับไฟล์ที่อัพโหลดภายใน 24 ชั่วโมง

6. โปรแกรมประยุกต์สำหรับอุปกรณ์เคลื่อนที่จะต้องรองรับการสร้างหัวข้อสนทนาในกระดานสนทนาของรายวิชานั้นๆ และสามารถเพิ่มความคิดเห็นในหัวข้อสนทนาได้ นักศึกษาที่ลงทะเบียนเรียน ผู้สอนของรายวิชา และผู้สอนร่วมของรายวิชา สามารถสร้างหัวข้อสนทนา สามารถแก้ไขหัวข้อและลบหัวข้อสนทนาที่ตนเองสร้างได้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

**ตัวอย่างความต้องการซอฟต์แวร์ที่ไม่ใช่ด้านหน้าที่ (Non-Functional Requirement)**

7. โปรแกรมประยุกต์สำหรับอุปกรณ์เคลื่อนที่จะสามารถใช้งานได้ใน
อุปกรณ์เคลื่อนที่ที่เป็นระบบปฏิบัติการ iOS 10.3 ขึ้นไปที่มีประสิทธิภาพขั้นต่ำของอุปกรณ์ฮาร์ดแวร์เทียบเท่าประสิทธิภาพของ iPhone 6 ขึ้นไปที่ไม่ได้เติมแต่งระบบปฏิบัติการ (Jailbreak) และระบบปฏิบัติการ Android 6.0.1 ขึ้นไปที่ไม่ได้เติมแต่งระบบปฏิบัติการ (Root) ที่มีประสิทธิภาพขั้นต่ำดังนี้

a. มีหน่วยประมวลผลขั้นต่ำ 4 Cores 1.6 GHz (Cortex-A7)

b. หน่วยความจำภายใน (Ram) ขั้นต่ำ 1.5 Gigabytes

# ภาคผนวก ค.

**แบบสอบถามผู้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ ก่อนการประเมิน (แบบสอบถามที่ 1)**

**หลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิชาการพัฒนาซอฟต์แวร์ด้านธุรกิจ**

**คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย**

**<u>วัตถุประสงค์ของแบบสอบถาม</u>**

แบบสอบถามนี้มีวัตถุประสงค์ เพื่อทราบประสบการณ์การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เบื้องต้นหรือประสบการณ์ที่เกี่ยวข้องของผู้ประเมิน

ชื่อ-นามสกุล ของผู้เข้าร่วมการทดลอง _____

○ ชาย ○ หญิง อายุ (ปี) _____ วุฒิการศึกษาสูงสุด _____

1) เพศ
2) ท่านเคยทำงานอยู่ตำแหน่งใดบ้าง ระยะเวลาทำงานในแต่ละตำแหน่งนานเท่าไร

_____

_____

3) ท่านเคยพัฒนาหรือมีส่วนร่วมในโครงการซอฟต์แวร์ทั้งหมดกี่โครงการ แต่ละโครงการซอฟต์แวร์นั้นท่านมีส่วนร่วมกี่เดือน (ตัวอย่างคำตอบเช่น 3 โครงการ โดย โครงการที่1 – 2 เดือน โครงการที่ 2 – 3เดือน ...)

_____

4) เรียงลำดับภาษาการเขียนโปรแกรมที่ท่านถนัดจากความถนัดมาก -> น้อย จากภาษาที่กำหนดให้ต่อไปนี้ Javascript, PHP, C#, ClojureScript, JAVA, Python

_____

5) ท่านรู้จัก โคโคโม่ (COCOMO) มาก่อนหรือไม่ _____
6) ท่านรู้จัก ฟังก์ชันพอยต์ (Function Point) มาก่อนหรือไม่ _____
7) ท่านรู้จัก ยูสเคสพอยต์ (Use Case Point) มาก่อนหรือไม่ _____

# ภาคผนวก ง.

**แบบสอบถามผู้ประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ หลังการประเมิน**

**(แบบสอบถามที่ 2)**

**หลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิชาการพัฒนาซอฟต์แวร์ด้านธุรกิจ**

**คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย**

**<u>วัตถุประสงค์ของแบบสอบถาม</u>**

แบบสอบถามนี้มีวัตถุประสงค์ เพื่อทราบประสบการณ์การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์เบื้องต้นหรือประสบการณ์ที่เกี่ยวข้องของผู้ประเมิน

ชื่อ-นามสกุล ของผู้เข้าร่วมการทดลอง _____ โครงการซอฟต์แวร์ที่ _____

1.) ท่านเคยพัฒนาซอฟต์แวร์ที่มีฟังก์ชันการทำงานใกล้เคียง หรือคล้ายคลึงกันกับโครงการซอฟต์แวร์นี้หรือไม่ อย่างไร

_____

_____

_____

_____

2.) ในการประเมินโครงการนี้ ท่านมีวิธีการคิดในประเมินอย่างไร อธิบาย

_____

_____

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# ภาคผนวก จ.

## ตัวอย่างการทดสอบสมมติฐานของข้อมูลสมมติในสมมติฐานที่ 1

ใช้ข้อมูลสมมติจากตารางที่ 3.7 และ 3.8

ตารางที่ 144 การวิเคราะห์ความแปรปรวนของข้อมูลสมมติในสมมติฐานที่ 1

| แหล่งที่มา | ผลรวมของกำลังสอง (Sum of Square) | องศาอิสระ (Degree of Freedom) | ค่าประมาณความแปรปรวน (Variance Estimate) | อัตราส่วนเอฟ (F ratio) |
|---|---|---|---|---|
| ระหว่างกลุ่ม | $SS_B = 60.53$ | 4-1 = 3 | $\mathrm{MS}_B = 20.17$ | 48.02 |
| ภายในกลุ่ม | $SS_W = 31.97$ | 80-4 = 76 | $\mathrm{MS}_W = 0.42$ | |
| ทั้งหมด | $\mathrm{SS}_\mathrm{T} = 92.5$ | 79 | | |

$F_{คำนวณ}$ = 48.02

$F_{วิกฤต0.05(3,76)}$ =2.74

เนื่องจาก $F_{คำนวณ} > F_{วิกฤต}$ จึงปฏิเสธ $H_0$ และ ยอมรับ $H_1$

จากข้อมูลตัวอย่างสรุปได้ว่า ค่าความแม่นยำของการประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ระหว่างวิธี โคโคโม่, ฟังก์ชันพอยต์, ยูสเคสพอยต์ และวิธีการตัดสินโดยผู้เชี่ยวชาญ มี**ความแตกต่างกัน** อย่างน้อย 1 คู่ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ซึ่งผู้วิจัยจะทำการทดสอบสมติฐานต่อเพื่อหาว่าวิธีการใดบ้างที่มีค่าความแม่นยำที่แตกต่างกัน