การประมาณความพยายามการพัฒนาซอฟต์แวร์ด้วยแอคทิวิตีพอยท์

นายชารีฟ เด่นสุมิตร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Software Effort Estimation using Activity Point

Mr. Sharif Densumite

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2016

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประมาณความพยายามการพัฒนาซอฟต์แวร์ด้วยแอคทิวิตีพอยท์
โดย	นายชารีฟ เด่นสุมิตร
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี)

.....กรรมการ

(รองศาสตราจารย์ ดร. ทวีตีย์ เสนีวงศ์ ณ อยุธยา)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร. มฑุปายาส ทองมาก)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ชารีฟ เด่นสุมิตร : การประมาณความพยายามการพัฒนาซอฟต์แวร์ด้วยแอคทิวิตีพอยท์ (Software Effort Estimation using Activity Point) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. พรศิริ หมื่นไชยศรี, 138 หน้า.

วิทยานิพนธ์นี้ มีวัตถุประสงค์ในการนำเสนอแบบจำลองการประมาณขนาดของซอฟต์แวร์โดยใช้แผนภาพกิจกรรม (Activity Diagram) เพื่อให้ได้ผลลัพธ์ในรูปแบบของขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ (Activity Point) เพื่อใช้ในการเป็นข้อมูลนำเข้าในการประมาณความพยายามในการพัฒนาซอฟต์แวร์ พร้อมทั้งเสนอเครื่องมือที่ใช้ในการประมาณขนาดของซอฟต์แวร์จากการนำแผนภาพกิจกรรมที่ออกแบบมาใช้ในการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ เพื่อให้ผู้ที่วางแผนโครงการหรือองค์กรสามารถนำแบบจำลองไปใช้เพื่อการบริหารจัดการทรัพยากรที่เหมาะสมในการพัฒนาซอฟต์แวร์นั้น ๆ ได้อย่างมีประสิทธิภาพยิ่งขึ้น

การพัฒนาแบบจำลองการประมาณขนาดของซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์ เริ่มด้วยการนำปัจจัยต่าง ๆ ที่ส่งผลโดยตรงกับขนาดซอฟต์แวร์ ประกอบด้วย ทรานแซคชัน (Transaction) เส้นทาง (Path) และความสัมพันธ์กันระหว่างยูสเคส (Use Case Relations) มาใช้ในการคำนวณหาขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ก่อนการปรับปรุง (Unadjusted Activity Point) เพื่อใช้ในการปรับด้วยค่าน้ำหนักเพื่อให้ได้ขนาดของการพัฒนาซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์ต่อไป

งานวิจัยนี้ใช้โครงการพัฒนาซอฟต์แวร์ ในการสร้างแบบจำลองและทดสอบคุณภาพของแบบจำลอง ประกอบด้วย 10 โครงการ โดยแบ่งออกเป็น 2 กลุ่ม ดังนี้ กลุ่มที่หนึ่งประกอบด้วย 7 โครงการสำหรับใช้ในการสร้างแบบจำลองการประมาณขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ และกลุ่มที่สองประกอบด้วย 3 โครงการสำหรับการทดสอบคุณภาพของแบบจำลองที่ได้สร้างขึ้น โดยใช้มาตรวัด ในการประเมินคุณภาพของแบบจำลองนี้ ซึ่งประกอบด้วย ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (Average Magnitude Relative Error) และคุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.25 (Prediction Quality: PRED(25))

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2559

5670167621 : MAJOR SOFTWARE ENGINEERING
KEYWORDS: TRANSACTION; SCENARIO; UNADJUSTED USE CASE POINT; USE CASE POINT; TECHNICAL CONTROL FACTOR; ENVIRONMENT COMPLEXITY FACTOR; UNADJUSTED ACTIVITY POINT; ACTIVITY POINT.

SHARIF DENSUMITE: Software Effort Estimation using Activity Point. ADVISOR: ASSOC. PROF. PORNSIRI MUENCHAISRI, Ph.D., 138 pp.

This thesis aims to propose a model for estimating software size in activity point unit from activity diagram for software effort estimation. The research provides an estimating software size tool which estimates software size and software effort from Activity Diagram. The tool is developed for project manager to estimate with the Activity Point approach.

Developing of Activity Point method, starting from defining factors which are directly impact to software size. The factors include Transactions, Paths, and Use Case Relations of each use case to represent software size in Unadjusted Activity Point (UAP). UAP will be adjusted with external weight factors to get result of software size in activity point unit.

The research uses 10 software development projects to create and evaluate quality of the model which are separated into 2 groups. First group includes 7 projects to create an estimation model. Second group includes 3 projects to measure a quality of the model. Average Magnitude Relative Error and PRED(25) are used to evaluate quality of Activity Point approach.

Department: Computer Engineering Student's Signature .....
Field of Study: Software Engineering Advisor's Signature .....
Academic Year: 2016

กิตติกรรมประกาศ

งานวิจัยฉบับนี้จะไม่สำเร็จลุล่วงอย่างสมบูรณ์ได้เลย หากขาดความอนุเคราะห์เป็นอย่างดียิ่งจาก รองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษา ในการแนะนำ ชี้แนะ และวิเคราะห์ผลในผลงานการวิจัยเรื่อง การประมาณความพยายามการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์ ที่ใช้กิจกรรมใด ๆ ของการพัฒนาซอฟต์แวร์เป็นปัจจัยหลักในการประมาณ ซึ่งท่านได้กรุณาสละเวลาในการให้คำปรึกษา จนทำให้งานวิจัยนี้มีความถูกต้องและสมบูรณ์ยิ่งขึ้น ผู้ทำวิจัยจักขอบพระคุณในความกรุณาอย่างสูง ณ โอกาสนี้

อีกทั้งขอกราบขอบพระคุณ คณาจารย์คณะวิศวคอมพิวเตอร์ทุกท่าน ที่ได้บ่มเพาะ องค์ความรู้ที่สำคัญในสาขาวิชาการต่าง ๆ ซึ่งทำให้ผู้ทำงานวิจัยสามารนำความรู้ ความสามารถเหล่านั้น มาวิเคราะห์และเชื่อมโยงเพื่อนำมาใช้ในการแก้ปัญหาในงานวิจัยชิ้นนี้ ให้สำเร็จลุล่วงตามวัตถุประสงค์ที่มุ่งหมายไว้

สุดท้ายนี้ ขอขอบพระคุณครอบครัว และกัลยามิตรทุกท่าน ที่คอยให้กำลังใจและสนับสนุน พร้อมร่วมมือในการให้คำชี้แนะต่าง ๆ อันเป็นประโยชน์อย่างยิ่งต่อการทำงานวิจัยชิ้นนี้ ซึ่งผลที่ได้จะช่วยให้อุตสาหกรรมพัฒนาซอฟต์แวร์สามารถนำไปใช้ในการวางแผนการพัฒนาซอฟต์แวร์สืบต่อไป

ชารีฟ เด่นสุมิตร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตของการวิจัย 4

1.4 ขั้นตอนการดำเนินงาน 4

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย 5

1.5.1 ประโยชน์ทางตรง 5

1.5.2 ประโยชน์ทางอ้อม 5

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 5

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 6

2.1 แบบจำลองเชิงฟังก์ชันในการออกแบบกระบวนการทางธุรกิจด้วย UML 6

2.1.1 แผนภาพยูสเคส (Use Case Diagram) 6

2.1.2 แผนภาพกิจกรรม (Activity Diagram) 10

2.2 กราฟควบคุมกระแส (Control Flow Graph) 12

2.2.1 แผนภาพเชิงลำดับ 12

2.2.2 แผนภาพการวนซ้ำ 12

2.2.3 แผนภาพการตัดสินใจ 12

2.3 การประมาณขนาดของซอฟต์แวร์เชิงฟังก์ชัน 13

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.3.1 การประมาณขนาดด้วยวิธี Line of Code (LOC) 13
2.3.2 การประมาณขนาดด้วยวิธีฟังก์ชันพอยท์ (Function Point) 14
2.3.3 การประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐาน IFPUG 14
2.3.4 การประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐานอื่น ๆ 16
2.4 การประมาณขนาดของซอฟต์แวร์เชิงวัตถุ 16
2.4.1 การประมาณขนาดด้วยวิธียูสเคสพอยท์ 16
2.4.2 การประมาณขนาดด้วยวิธีซีเควนซ์พอยท์ 19
2.4.3 การประมาณขนาดด้วยวิธีคลาสพอยท์ 19
2.5 การประมาณความพยายามในการพัฒนาซอฟต์แวร์ 19
2.5.1 Algorithmic Models 20
2.5.2 Expert Judgment 20
2.5.3 Machine Learning Methods 21
2.5.4 Fuzzy Logic and Evolutionary Algorithms, Genetic Programming 21
บทที่ 3 งานวิจัยที่เกี่ยวข้อง 22
3.1 งานวิจัย “Metrics for Objectory” [9] 22
3.2 งานวิจัย "Class point: an approach for the size estimation of object-oriented systems," [11] 23
3.3 งานวิจัย "Transactions and paths: Two use case based metrics which improve the early effort estimation," [21] 24
3.4 งานวิจัย “A Complexity Measure” [22] 25
3.5 งานวิจัย "The Impact of Non-Technical Factors on Software Architecture," [23] 25
3.6 งานวิจัย "An Alternative Method Employing Uses Cases for Early Effort Estimation," [24] 26

หน้า

บทที่ 4 การออกแบบและวิธีการประมาณความพยายามด้วยแอคทิวิตีพอยท์ 27
4.1 คำนิยามและมุมมองแผนภาพกิจกรรม 27
4.2 การออกแบบแผนภาพกิจกรรมในระดับบนจากแผนภาพยูสเคส 29
4.2.1 จุดเริ่มต้นของแผนภาพกิจกรรม 29
4.2.2 กรณียูสเคสไม่ได้มีการปฏิสัมพันธ์ 29
4.2.3 กรณียูสเคสมีการปฏิสัมพันธ์ 29
4.3 การแปลงแผนภาพกิจกรรมเป็นกราฟควบคุมกระแส 32
4.3.1 เส้นทางเชิงลำดับของแผนภาพกิจกรรม 32
4.3.2 เส้นทางของการตัดสินใจ 33
4.3.3 เส้นทางของการวนซ้ำ 34
4.3.4 เส้นทางของ Fork/Join เป็นแผนภาพเชิงลำดับ 34
4.3.5 เส้นทางของ Fork/Join ประกอบด้วยเส้นทางการตัดสินใจ (Decision) หรือ เส้นทางวนซ้ำ (Looping) ภายใน 35
4.3.6 เส้นทางของ Fork/Join เป็นเส้นทางมีการเชื่อมประสานกับเส้นทางอื่น ๆ (Synchronization link) 37
4.4 ภาพรวมของวิธีการประมาณขนาดและความพยายามด้วยแอคทิวิตีพอยท์ 38
4.5 วิธีการประมาณด้วยแอคทิวิตีพอยท์ 39
4.5.1 ผลรวมของแอคทิวิตีพอยท์ (Total Activity Point) 39
4.5.2 ผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม (Total Activity Complexity Weight) 41
4.5.3 ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (Unadjusted Activity Point) 42
4.5.4 แอคทิวิตีพอยท์ (Activity Point) 42
4.5.5 ความพยายาม (Effort) 45

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.6 การกำหนดค่าน้ำหนักของระดับสัญลักษณ์กิจกรรม ระดับเส้นทางที่เป็นไปได้ และระดับความซับซ้อนของกิจกรรม 45
บทที่ 5 การวิเคราะห์และออกแบบการพัฒนาเครื่องมือวิจัย 50
5.1 การวิเคราะห์ความต้องการของเครื่องมืองานวิจัย 50
5.2 การออกแบบเครื่องมืองานวิจัย 51
5.2.1 แผนภาพยูสเคสของเครื่องมืองานวิจัย 51
5.2.2 แผนภาพกิจกรรมของเครื่องมืองานวิจัย 52
5.2.3 แผนภาพคลาสของเครื่องมืองานวิจัย 53
บทที่ 6 การประเมินผล 55
6.1 ผลการตรวจสอบวิธีการประมาณด้วยวิธีแอคทิวิตีพอยท์ 55
6.2 ผลการเปรียบเทียบวิธีการประมาณด้วยวิธียูสเคสพอยท์และวิธีแอคทิวิตีพอยท์ 64
บทที่ 7 บทสรุปงานวิจัยและงานวิจัยในอนาคต 66
7.1 สรุปผลงานวิจัย 66
7.2 งานวิจัยในอนาคต 66
รายการอ้างอิง 67
ภาคผนวก ก รายละเอียดยูสเคสเครื่องมืองานวิจัย 70
ภาคผนวก ข วิธีการติดตั้งเครื่องมืองานวิจัยสำหรับการพัฒนา 82
ภาคผนวก ค โครงสร้างชุดรหัสข้อมูลเครื่องมืองานวิจัย 103
ภาคผนวก ง การใช้งานเครื่องมืองานวิจัย 106
ภาคผนวก จ ตัวอย่างการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์ 118
ประวัติผู้เขียนวิทยานิพนธ์ 138

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

ตารางที่ 1 องค์ประกอบของแผนภาพยูสเคส [12] 7
ตารางที่ 2 ตัวอย่างข้อมูลของรายละเอียดของแผนภาพยูสเคส 8
ตารางที่ 3 องค์ประกอบของแผนภาพกิจกรรม [12] 11
ตารางที่ 4 ประเภทการเคลื่อนที่ของข้อมูลด้วยมาตรฐานด้วยวิธี COSMIC 16
ตารางที่ 5 เกณฑ์การแบ่งระดับสัญลักษณ์กิจกรรมหรือการกระทำ 40
ตารางที่ 6 เกณฑ์การแบ่งระดับของเส้นทางที่เป็นไปได้ 41
ตารางที่ 7 เกณฑ์การแบ่งระดับระดับของความซับซ้อนของกิจกรรม 42
ตารางที่ 8 ปัจจัยความซับซ้อนทางเทคนิค [9] 43
ตารางที่ 9 ปัจจัยความซับซ้อนทางสภาพแวดล้อม [9] 43
ตารางที่ 10 ปัจจัยทางด้านความเสี่ยงของคน 44
ตารางที่ 11 สัดส่วนของปัจจัยที่มีความสัมพันธ์กับความพยายามในการพัฒนาซอฟต์แวร์จริง 46
ตารางที่ 12 จำนวนระดับของเส้นทาง ทรานแซคชัน และค่าความซับซ้อนของกิจกรรม 47
ตารางที่ 13 ค่าน้ำหนักที่ทำให้ค่าประมาณมีความใกล้เคียงความพยายามในการพัฒนาซอฟต์แวร์จริงสูงสุด 47
ตารางที่ 14 น้ำหนักเกณฑ์การแบ่งระดับสัญลักษณ์กิจกรรมหรือการกระทำ 49
ตารางที่ 15 น้ำหนักเกณฑ์การแบ่งระดับของเส้นทางที่เป็นไปได้ 49
ตารางที่ 16 น้ำหนักเกณฑ์การแบ่งระดับระดับของความซับซ้อนของกิจกรรม 49
ตารางที่ 17 ผลรวมค่าน้ำหนักของกิจกรรมของแต่ละโครงการ 56
ตารางที่ 18 ผลรวมค่าน้ำหนักของกิจกรรมของแต่ละโครงการ 57
ตารางที่ 19 ผลรวมของแอคทิวิตีของแต่ละโครงการ 58
ตารางที่ 20 ผลรวมค่าน้ำหนักความซับซ้อนกิจกรรมของแต่ละโครงการ 59
ตารางที่ 21 ขนาดของแอคทิวีตีพอยท์ก่อนการปรับปรุงของแต่ละโครงการ 60
ตารางที่ 22 ค่าขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ของแต่ละโครงการ 61

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 23 ข้อมูลผลสรุปการประมาณขนาดของซอฟต์แวร์และความคลาดเคลื่อน 63

ตารางที่ 24 ตารางเปรียบเทียบการประมาณความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยท์และแอคทิวิตีพอยท์ 64

ตารางที่ 25 ตารางเปรียบเทียบความคลาดเคลื่อนสัมพัทธ์ของค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยการประมาณด้วยวิธียูสเคสพอยท์และวิธีแอคทิวิตีพอยท์ 65

ตารางที่ 26 รายละเอียดยูสเคสของระบบสมัครเรียน 122

ตารางที่ 27 ตารางสรุปจำนวนของระดับทรานแซคชัน เส้นทาง และจำนวนความซับซ้อนของกิจกรรมของระบบสมัครเรียน 135

ตารางที่ 28 จำนวนของระดับปัจจัยเส้นทางของระบบสมัครเรียน 135

ตารางที่ 29 จำนวนของระดับปัจจัยกิจกรรมของระบบสมัครเรียน 135

ตารางที่ 30 ค่าผลรวมของแอคทิวิตีพอยท์ของระบบสมัครเรียน 136

ตารางที่ 31 ค่าผลรวมของค่าน้ำหนักความซับซ้อนกิจกรรมของระบบสมัครเรียน 136

ตารางที่ 32 ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงของระบบสมัครเรียน 136

ตารางที่ 33 ขนาดของแอคทิวิตีพอยท์ของระบบสมัครเรียน 136

ตารางที่ 34 ความพยายามในการพัฒนาซอฟต์แวร์ของระบบสมัครเรียน 137

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญรูปภาพ

รูปที่ 1 ตัวอย่างแผนภาพยูสเคสระบบสมัครเรียน 9
รูปที่ 2 รูปแบบของกราฟควบคุมกระแส 12
รูปที่ 3 แบบฟอร์มการประมาณขนาดด้วยวิธีฟังก์ชันพอยท์ [2, 3] 15
รูปที่ 4 แผนภาพยูสเคสระบบ Online Shopping [15] 18
รูปที่ 5 กระบวนการประมาณขนาดด้วยยูสเคสพอยท์ [9] 23
รูปที่ 6 ขั้นตอนการประมาณขนาดซอฟต์แวร์ด้วยวิธีคลาสพอยท์ 24
รูปที่ 7 ปัจจัยที่ไม่ใช่ทางเทคนิคที่ส่งผลต่อการพัฒนาซอฟต์แวร์ [23] 25
รูปที่ 8 มุมมองแผนภาพกิจกรรม 28
รูปที่ 9 ตัวอย่างแผนภาพยูสเคสระบบสมัครเรียน 30
รูปที่ 10 ตัวอย่างแผนภาพกิจกรรมระดับบนของระบบสมัครเรียน 31
รูปที่ 11 เส้นทางเชิงลำดับของแผนภาพกิจกรรม 32
รูปที่ 12 เส้นทางของแผนภาพกิจกรรมในรูปแบบเส้นทางตัดสินใจ 33
รูปที่ 13 เส้นทางของแผนภาพกิจกรรมที่ประกอบด้วยเส้นทางวนซ้ำ 34
รูปที่ 14 เส้นทางของ Fork/Join เป็นแผนภาพเชิงลำดับ 35
รูปที่ 15 เส้นทางของ Fork/Join ประกอบด้วยเส้นทางตัดสินใจหรือเส้นทางวนซ้ำ 36
รูปที่ 16 เส้นทางของ Fork/Join ประกอบด้วยการเชื่อมประสานกับเส้นทางอื่น ๆ 37
รูปที่ 17 ภาพรวมของวิธีการประมาณขนาดและความพยายามด้วยแอคทิวิตีพอยท์ 38
รูปที่ 18 ค่าสหสัมพันธ์ที่มีผลกับความพยายามจริงในการพัฒนาซอฟต์แวร์ 45
รูปที่ 19 ลำดับรหัสเทียมการหาค่าน้ำหนักของวิธีแอคทิวิตีพอยท์ 48
รูปที่ 20 แผนภาพยูสเคสเครื่องมืองานวิจัย 51
รูปที่ 21 แผนภาพกิจกรรมของเครื่องมืองานวิจัย 52
รูปที่ 22 แผนภาพคลาสเครื่องมืองานวิจัย 54
รูปที่ 23 ข้อมูลสถิติการสร้างแบบจำลองความพยายามด้วยแอคทิวิตีพอยท์ 62

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 24 ลิงค์ขั้นตอนการติดตั้ง JDK ในการพัฒนาด้วยภาษา Java 82
รูปที่ 25 ลิงค์ขั้นตอนการติดตั้งเครื่องมือการพัฒนาด้วย Eclipse 83
รูปที่ 26 ลิงค์ขั้นตอนการดาวน์โหลด Server Tomcat version 8 84
รูปที่ 27 ขั้นตอนการตั้งค่า Server Tomcat ใน Eclipse 85
รูปที่ 28 ขั้นตอนการเลือก Server Tomcat ใน Eclipse 86
รูปที่ 29 ขั้นตอนการเลือกตำแหน่งของ Server Tomcat ใน Eclipse 87
รูปที่ 30 ขั้นตอนการ Start Server Tomcat ใน Eclipse 88
รูปที่ 31 ลิงค์การติดตั้ง Vaadin Plugin for Eclipse 90
รูปที่ 32 ขั้นตอนการเลือกติดตั้ง Plugin ใน Eclipse 91
รูปที่ 33 ขั้นตอนการติดตั้ง Vaadin Plugin จากลิงค์ Plugin ที่เลือก 92
รูปที่ 34 การติดตั้ง Vaadin Plugin ใน Eclipse เสร็จสิ้น 93
รูปที่ 35 www.bitbucket.org 94
รูปที่ 36 ลิงค์การติดตั้งโปรแกรม SourceTree โปรแกรมจัดการเวอร์ชันชุดรหัสข้อมูล 95
รูปที่ 37 ขั้นตอนการ Clone ชุดรหัสเครื่องมือวิจัยด้วยโปรแกรม SourceTree 96
รูปที่ 38 ขั้นตอนการ Import ชุดรหัสโครงการมายังโปรแกรม Eclipse 98
รูปที่ 39 ขั้นตอนการเลือกประเภทชุดรหัสข้อมูลของโครงการที่มีอยู่ในเครื่องผู้พัฒนา 99
รูปที่ 40 ขั้นตอนการเลือกตำแหน่งของชุดรหัสโครงการเพื่อนำเข้าในโปรแกรม Eclipse 99
รูปที่ 41 ขั้นตอนการเปิดไฟล์หลักของโครงการวิจัยเพื่อทำการรัน 100
รูปที่ 42 ขั้นตอนการรันโครงการงานวิจัย 101
รูปที่ 43 ขั้นตอนผลลัพธ์การรันเครื่องมืองานวิจัย 102
รูปที่ 44 โครงสร้างไฟล์ของชุดรหัสเครื่องมือวิจัย 105
รูปที่ 45 การนำข้อมูลแผนภาพกิจกรรมออกในโปรแกรม Visual Paradigm 110
รูปที่ 46 หน้าแรกของเครื่องมือการประมาณขนาดด้วยวิธีแอคทิวิตีพอยท์ 111
รูปที่ 47 ขั้นตอนการใช้งานในส่วนของการจำแนกแผนภาพกิจกรรม 112

รูปที่ 48 ขั้นตอนหน้าการคำนวณค่า TAW TPW และ TAP 113
รูปที่ 49 ขั้นตอนการทำงานหน้าการคำนวณค่า TACW 114
รูปที่ 50 ขั้นตอนการใช้งานหน้าการคำนวณค่า UAP 115
รูปที่ 51 ขั้นตอนการทำงานหน้าการกำหนดค่าของปัจจัยในการปรับค่า 116
รูปที่ 52 ขั้นตอนการคำนวณค่าขนาดแอคทิวิตีพอยท์ และความพยายามในการพัฒนาซอฟต์แวร์ 117
รูปที่ 53 แผนภาพยูสเคสของระบบสมัครเรียน 119
รูปที่ 54 แผนภาพกิจกรรมในมุมมองระดับบนของระบบสมัครเรียน 120
รูปที่ 55 แผนภาพควบคุมกระแสมุมมองระดับบนของระบบสมัครเรียน 121

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การประมาณความพยายามในการพัฒนาซอฟต์แวร์ เป็นหนึ่งในกระบวนการที่ใช้ในการบริหารจัดการโครงการเพื่อใช้ในการพิจารณาจัดสรรทรัพยากรและงบประมาณ ให้ครอบคลุมและเหมาะสมต่อค่าใช้จ่าย ระยะเวลา จำนวนผู้พัฒนา

รูปแบบในการประมาณความพยายามในการพัฒนาซอฟต์แวร์ ถูกจำแนกออกเป็น 3 รูปแบบ [1] คือ การประมาณความพยายามจากมุมมองระดับบนลงล่าง (Top-down Effort Estimation) ซึ่งเป็นการประมาณจากทั้งโครงการและแตกออกเป็นหน่วยย่อย ๆ ซึ่งเป็นวิธีที่เหมาะกับโครงการที่มีความต้องการของระบบที่มีความคล้ายเคียงกับโครงการก่อนหน้าที่พัฒนาเสร็จสิ้นแล้ว การประมาณความพยายามจากมุมมองระดับล่างขึ้นบน (Bottom-up Effort Estimation) เป็นการประมาณจากงานเล็ก ๆ แล้วรวมกันเพื่อให้อยู่ในระดับความพยายามของทั้งโครงการ ซึ่งเป็นวิธีที่เหมาะกับโครงการที่ไม่มีข้อมูลโครงการก่อนหน้ามาใช้เปรียบเทียบกันในการประมาณ และการประมาณความพยายามด้วยแบบจำลองทางคณิตศาสตร์ (Parametric Model) ซึ่งเป็นรูปแบบการประมาณโดยใช้สมการคณิตศาสตร์ที่ในรูปของแบบจำลองมาใช้ ซึ่งเหมาะกับการประมาณที่เป็นมาตรฐานกับโครงการที่เข้ากับบริบทของแบบจำลองนั้น ๆ ซึ่งการประมาณในแต่ละรูปแบบนั้นต่างก็ใช้ปัจจัยในการประเมินที่แตกต่างกันตามเงื่อนไขและข้อกำหนดที่ถูกระบุในแต่ละวิธี เพื่อให้ได้มาซึ่งผลลัพธ์ค่าประมาณความพยายามในการพัฒนาซอฟต์แวร์ ดังนั้นวิธีการที่ใช้ในการประมาณความพยายามที่ได้ค่าประมาณที่มีความคลาดเคลื่อนน้อย ก็จะช่วยให้การวางแผนโครงการมีความเหมาะสมต่อการจัดสรรทรัพยากรที่จำเป็นในการพัฒนาซอฟต์แวร์ได้อย่างมีประสิทธิภาพมากขึ้น

ปัจจุบันขนาดของซอฟต์แวร์เป็นหนึ่งในปัจจัยหลักที่นิยมใช้ในการประมาณความพยายามในการพัฒนาซอฟต์แวร์ อาทิเช่น จำนวนบรรทัดของซอฟต์แวร์ (Software Line of Code) เป็นวิธีหนึ่งในปัจจัยที่ถูกใช้แทนขนาดของซอฟต์แวร์ โดยประมาณได้จากการนับจำนวนบรรทัดของชุดรหัสซอฟต์แวร์ที่เสร็จสิ้นในขั้นตอนการพัฒนา อีกทั้งยังมีแบบจำลองฟังก์ชันพอยท์ (Function Point) IFPUG (The International Function Point User Group) เป็นต้น ที่นำเสนอถึงวิธีการประมาณขนาดของซอฟต์แวร์ให้สามารถประมาณขนาดได้ตั้งแต่ในขั้นตอนต้นของกระบวนการพัฒนาซอฟต์แวร์ เพื่อให้สามารถจัดสรรทรัพยากรที่จำเป็นต้องใช้ในการพัฒนาซอฟต์แวร์ได้ตั้งแต่การวางแผนโครงการ

ในปี 1979 Allan Albrecht [2, 3] เสนอถึงการประมาณขนาดด้วยวิธีการวิเคราะห์ฟังก์ชันพอยท์ (Function Point Analysis) เพื่อช่วยให้สามารถประมาณขนาดซอฟต์แวร์ได้ตั้งแต่ขั้นตอนต้น ๆ ของกระบวนการการพัฒนาซอฟต์แวร์ โดยมีวิธีในการประมาณจากการนับปัจจัยทั้ง 4 ประเภท ประกอบด้วย ข้อมูลนำเข้า (Input), ข้อมูลส่งออก (Output), การสอบถามข้อมูล (Inquiries) และไฟล์ใช้งานหลัก (Master File) เพื่อนำจำนวนของแต่ประเภทมาถ่วงน้ำหนักและคำนวณหาผลรวมของขนาดซอฟต์แวร์ฟังก์ชันพอยท์ก่อนปรับปรุง (Unadjusted Function Point) ไปสู่การปรับค่าในการปรับปรุงด้วยค่าปัจจัยความซับซ้อน (Complexity Adjustment) เพื่อให้ได้ผลลัพธ์ขนาดของซอฟต์แวร์ในหน่วยฟังก์ชันพอยท์ (Function Point Delivered or Designed)

ต่อมาในปี 1984 – 1986 วิธีการวิเคราะห์ฟังก์ชันพอยท์ได้ถูกพัฒนาต่อยอด ภายใต้การสนับสนุนของลุ่มผู้ใช้ฟังก์ชันพอยต์สากล (The International Function Point User Group: IFPUG) ที่ทำหน้าที่ในการระบุมาตรฐานในแต่ละเวอร์ชันของวิธีการวิเคราะห์ฟังก์ชันพอยท์ โดยการปรับปรุงปัจจัยที่ใช้ในการประมาณเป็น 5 ปัจจัย โดยรายละเอียดเพิ่มเติมจะถูกอธิบายในบทที่ 2 จึงทำให้วิธีการวิเคราะห์ฟังก์ชันพอยท์จึงถูกเรียกชื่อแทนด้วย IFPUG ต่อมา [4] และนำไปสู่การพัฒนาต่อยอดไปสู่วิธีอื่น ๆ อีกมากมายจากมาตรฐานนี้ เพื่อให้สอดคล้องกับบริบทต่าง ๆ ของประเภทซอฟต์แวร์ที่ถูกพัฒนาในช่วงเวลาดังกล่าว ยกตัวอย่างเช่น การประมาณขนาดซอฟต์แวร์ด้วยมาตรฐานของสมาคมผู้ใช้ตัววัดซอฟต์แวร์แห่งเนเธอร์แลนด์ (Netherlands Software Metrics users Association: NESMA) [5] การประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐานขององค์กร The Common Software Measurement International Consortium – Full Function Point (COSMIC-FFP) [6, 7] และการประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐาน Mark II (Mk II) [8]

การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object Oriented Programming) ได้รับความนิยมมากขึ้น จากการนำ Unified Modeling Language (UML) มาใช้ในการนำเสนอถึงการอธิบายการออกแบบระบบเชิงวัตถุ ในขั้นตอนก่อนการพัฒนาซอฟต์แวร์ และถูกนำมาใช้เป็นเกณฑ์ในการประมาณขนาดของซอฟต์แวร์ในแต่ละบริบท อาทิเช่น ยูสเคสพอยท์ (Use Case Point) [9] ซีเควนส์พอยท์ (Sequence Point) [10] และ คลาสพอยท์ (Class Point) [11]

วิธีการประมาณขนาดซอฟต์แวร์ด้วยยูสเคสพอยท์ที่ได้กล่าวมาข้างต้นนั้น ถูกนำเสนอโดย Gustav Karner ในปี 1993 ซึ่งเป็นหนึ่งในวิธีที่ใช้ในการประมาณขนาดของซอฟต์แวร์ในขั้นตอนเริ่มต้นของการพัฒนาซอฟต์แวร์ โดยมีการนำแผนภาพยูสเคส (Use Case Diagram) มาใช้ในการรองรับการประมาณระบบที่ออกแบบในเชิงวัตถุ ซึ่งประกอบด้วยปัจจัยหลักที่นำมาใช้เป็นพิจารณาคือ

- ทรานแซ็คชัน (Transaction) แสดงถึงลำดับขั้นตอนย่อยของฟังก์ชันหนึ่ง ๆ ตั้งแต่เริ่มต้นจนถึงจุดสิ้นสุด ที่แสดงอยู่ภายในรายละเอียดของยูสเคส (Use Case Description) นั้น ๆ
- ผู้มีปฏิสัมพันธ์กับระบบ (Actor) แสดงถึงผู้ใช้งานที่เกี่ยวข้องกับระบบ ที่มีปฏิสัมพันธ์ระหว่างยูสเคสเหล่านั้น

การประมาณขนาดด้วยวิธียูสเคสพอยท์จะนำปัจจัยหลักข้างต้น มาประมาณหาผลรวมของปัจจัยในแต่ละระดับเพื่อนำมาปรับค่าด้วยปัจจัยภายนอกที่ส่งผลกระทบต่อขนาดของซอฟต์แวร์ เพื่อให้ได้มาซึ่งขนาดของซอฟต์แวร์ในหน่วยของยูสเคสพอยท์

จากวิธีการของยูสเคสพอยท์ที่นำ ทรานแซ็คชัน และ ผู้มีปฏิสัมพันธ์กับระบบ มาใช้ในการประมาณขนาดด้วยนั้น ไม่ได้มีการพิจารณาถึงซีเนริโอ (Scenario) ที่เป็นส่วนของเหตุการณ์ที่เป็นไปได้ของฟังก์ชันหนึ่ง ๆ ตั้งแต่เริ่มต้นจนถึงสิ้นสุด เนื่องจากหากจำนวนของซีเนริโอมีจำนวนที่มากขึ้น ความซับซ้อนของระบบที่พัฒนาก็ยิ่งซับซ้อนขึ้นด้วย อีกทั้งความสัมพันธ์กันระหว่างยูสเคส (Use Case Relations) ที่แสดงถึงการเชื่อมต่อประสานของแต่ละยูสเคส เปรียบได้กับส่วนเชื่อมต่อประสาน (Glue Codes) ในแต่ละคอมโพเนนท์ในขั้นตอนการพัฒนาซอฟต์แวร์ ที่ส่งผลต่อโดยตรงกับขนาดของซอฟต์แวร์และความพยายามที่ในการนำมาใช้ในการพัฒนาการทำงานในส่วนนั้น ๆ เช่นเดียวกัน

ดังนั้นในงานวิจัยนี้ จึงได้นำเสนอวิธีการประมาณขนาดของซอฟต์แวร์ด้วยวิธีแอคทิวิตี พอยท์ (Activity Point) โดยนำแผนภาพกิจกรรม (Activity Diagram) มาใช้ในการประมาณขนาดของซอฟต์แวร์ ที่มีการนำปัจจัยของทรานแซ็คชัน ซีเนริโอ และความสัมพันธ์กันระหว่างยูสเคสมาใช้เป็นข้อมูลนำเข้าหลัก ในการประมาณขนาดของซอฟต์แวร์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อเสนอถึงวิธีการประมาณขนาดและความพยายามจากแผนภาพกิจกรรม ให้สามารถประเมินได้ในขั้นตอนต้น ที่ลดข้อจำกัดของการประมาณความพยายามด้วยแผนภาพยูสเคสที่ไม่ได้พิจารณาถึงการปฏิสัมพันธ์กันระหว่างยูสเคส ซึ่งมีผลต่อขนาดของซอฟต์แวร์เช่นเดียวกับปัจจัยอื่น ๆ มาใช้

1.3 ขอบเขตของการวิจัย

1.3.1 งานวิจัยนี้เสนอวิธีการประมาณขนาดและความพยายามโดยใช้แผนภาพกิจกรรมของแผนภาพยูเอ็มแอล เวอร์ชัน 2.0 ในการประมาณค่า

1.3.2 งานวิจัยนี้ ใช้กรณีศึกษาทั้งหมด 10 โครงการ โดยใช้ในการสร้างแบบจำลอง 7 โครงการ และ การทดสอบคุณภาพของแบบจำลอง 3 โครงการ

1.3.3 ความพยายามในแต่ละโครงการ มีหน่วยของความพยายามเป็น คนต่อชั่วโมง

1.3.4 พัฒนาเครื่องมือในการประมาณความพยายามด้วยแอคทิวิตีพอยท์ที่นำเสนอในงานวิจัยนี้ โดยมีรูปแบบข้อมูลนำเข้าด้วยไฟล์เอกซ์เอ็มแอล

1.3.5 การประเมินแบบจำลองการประมาณขนาดด้วยแอคทิวิตีพอยท์ใช้ ค่าความคลาดเคลื่อนสัมพัทธ์ของแต่ละโครงการ มาหาค่าเฉลี่ยค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ของทุกโครงในการสรุปถึงความคลาดเคลื่อนของผลลัพธ์ และนำการวัดคุณภาพของการประมาณมาหาจำนวนร้อยละของโครงการว่ามีจำนวนโครงการร้อยละเท่าใดที่มีผลลัพธ์อยู่ในช่วงที่ยอมรับ

1.3.6 วิเคราะห์สมการที่นำเสนอในการประมาณขนาดและความพยายามด้วยแอคทิวิตีพอยท์ถึงที่มาของสมการและตัวแปรต่าง ๆ ภายใน

1.4 ขั้นตอนการดำเนินงาน

ขั้นตอนและวิธีการ การดำเนินงานมีรายละเอียดดังต่อไปนี้

1.4.1 ศึกษาข้อมูลการนิยามและกระบวนการในการกำหนดมาตรวัดของขนาดซอฟต์แวร์ ที่ถูกเสนอในอดีตและวิธีที่เป็นที่นิยมใช้ในปัจจุบัน

1.4.2 ศึกษาถึงขั้นตอนการประมาณขนาดด้วยวิธีต่าง ๆ ในการประมาณขนาดของซอฟต์แวร์

1.4.3 กำหนดคุณลักษณะต่าง ๆ ของวิธีการประมาณขนาดของซอฟต์แวร์ ที่สามารถนำมาใช้ในการประมาณขนาดจากการนำกิจกรรมเป็นปัจจัยหลักในการพิจารณาที่ถูกใช้ในการพัฒนาซอฟต์แวร์

1.4.4 ออกแบบขั้นตอนและวิธีการประมาณจากคุณลักษณะที่ได้กำหนดไว้

1.4.5 ประเมินคุณภาพของแบบจำลองการประมาณขนาดที่ได้ออกแบบไว้ เพื่อหาคุณภาพของแบบจำลองว่ามีอัตราความถูกต้องเพียงใด จากข้อมูลทดสอบ 10 โครงการซอฟต์แวร์

1.4.6 ศึกษาเครื่องมือที่ช่วยในการประมาณขนาดของซอฟต์แวร์ที่ใช้อยู่ ณ ปัจจุบัน

1.4.7 ออกแบบและพัฒนาเครื่องมือในการช่วยการประมาณขนาดด้วยแบบจำลองที่ได้ออกแบบไว้

1.4.8 จัดทำบทความวิชาการและนำเสนอ

1.4.9 สรุปผลแนวทางการวิจัยและข้อเสนอแนะ

1.4.10 จัดทำรูปเล่มวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1.5.1 ประโยชน์ทางตรง

- เสนอรูปแบบการประมาณความพยายามจากแผนภาพกิจกรรมที่ตอบสนองต่อบริบทของประเภทระบบเพื่อให้ตรงต่อค่าประมาณที่เหมาะสม
- ลดความเสี่ยงในการพัฒนา ไม่ให้ใช้เวลา ทรัพยากร หรือพัฒนาการทำงานที่เกินความจำเป็นและไม่สอดคล้องกับเป้าหมายทางธุรกิจ

1.5.2 ประโยชน์ทางอ้อม

- ทำให้ผู้พัฒนามีการกำหนดและวางแผนกิจกรรมที่ใช้ในการพัฒนา ทำให้ลดความเสี่ยงต่อการพัฒนาหรือดำเนินกิจกรรมที่นอกเหนือจากที่วางแผนจนก่อให้เกิดผลกระทบต่อภาพรวมของการพัฒนา
- ทำให้การพัฒนา ใช้ข้อมูลตามบริบทจริงของทีมผู้พัฒนา ตามกิจกรรมที่ทุกฝ่ายต้องพึงปฏิบัติ ทำให้รองรับในทุก ๆ บริบทของประเภทการพัฒนาซอฟต์แวร์
- รูปแบบของการประมาณไม่ขึ้นอยู่กับ ภาษาที่ใช้ในการพัฒนา และ ประเภทของระบบที่ถูกพัฒนา

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

บทความวิชาการในหัวข้อเรื่อง “Software Size Estimation Using Activity Point” by Sharif Densumite, and Pornsiri Muenchaisri, The International Conference on Information Technology and Digital Applications (ICITDA), 14 – 16 Nov, 2016, Yogyakarta, Indonesia

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้กล่าวถึงแผนภาพที่ใช้ในการออกแบบกระบวนการทางธุรกิจด้วย UML กราฟควบคุมกระแส (Control Flow Graph) และภาพรวมของการประมาณขนาดซอฟต์แวร์ด้วยวิธีการต่าง ๆ ที่แบ่งออกตามรูปแบบการประมาณ ประกอบด้วย การประมาณขนาดของซอฟต์แวร์เชิงฟังก์ชัน การประมาณขนาดของซอฟต์แวร์เชิงวัตถุ รวมถึงการประมาณความพยายามในการพัฒนาซอฟต์แวร์ ที่เป็นที่นิยมใช้ในการประมาณในแต่ละยุคสมัย และสอดคล้องกับบริบทของการพัฒนาซอฟต์แวร์ในรูปแบบต่าง ๆ

2.1 แบบจำลองเชิงฟังก์ชันในการออกแบบกระบวนการทางธุรกิจด้วย UML

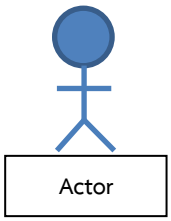
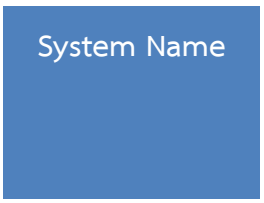
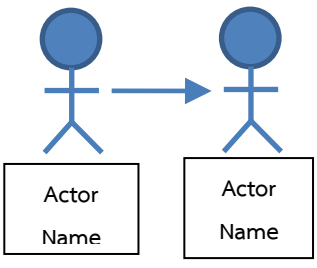
2.1.1 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคสเป็นแผนภาพที่ใช้ในการอธิบายการทำงานของผู้ใช้ระบบและความสัมพันธ์ย่อยของระบบ เพื่ออธิบายความต้องการของผู้ใช้ว่ามีความต้องการใดบ้าง เพื่อเป็นจุดเริ่มต้นในการวิเคราะห์และออกแบบระบบ โดยประกอบด้วย แผนภาพ และรายละเอียดของกิจรรมที่ผู้ใช้มีปฏิสัมพันธ์กับระบบ ทำให้ผู้ที่เกี่ยวข้องทั้งหมดมีความเข้าใจภาพรวมของระบบในทิศทางเดียวกันและง่ายในการสื่อสารกันภายในทีมผู้พัฒนาให้มีประสิทธิภาพยิ่งขึ้น

2.1.1.1 องค์ประกอบของแผนภาพยูสเคส

การออกแบบแผนภาพยูสเคส ถูกออกแบบด้วยการนำความต้องการของผู้ใช้ที่ถูกวิเคราะห์จากขั้นตอนการได้มาซึ่งความต้องการของผู้ใช้ (User Requirement Elicitation) มาออกแบบในรูปแบบของแผนภาพสัญลักษณ์ ในการอธิบายระหว่างทีมผู้พัฒนาและผู้ใช้ โดยมีวัตถุประสงค์ในการทวนสอบความของความต้องการที่ได้มาว่ามีความครบถ้วนและตรวจสอบความถูกต้องในรูปแบบสัญลักษณ์ที่เข้าใจง่ายขึ้น โดยองค์ประกอบต่าง ๆ ที่ใช้ในการออกแบบแผนภาพยูสเคสดังถูกอธิบายและการนำไปใช้ในตารางที่ 1

ตารางที่ 1 องค์ประกอบของแผนภาพยูสเคส [12]

องค์ประกอบ	สัญลักษณ์
ผู้มีปฏิสัมพันธ์ (Actor) แสดงถึงผู้ที่มีส่วนร่วมในการปฏิสัมพันธ์ทั้งภายในและภายนอกกับระบบ โดยประกอบด้วยชื่อระบุอยู่ล่างสัญลักษณ์เพื่อแสดงถึงบทบาทของผู้มีปฏิสัมพันธ์ เช่น เจ้าหน้าที่ฝ่ายจัดซื้อ ผู้ดูแลระบบ ผู้ใช้ทั่วไป เป็นต้น	
ยูสเคส (Use case) แสดงถึงหน้าที่หรือการทำงานต่าง ๆ ของระบบที่แสดงถึงความต้องการของผู้ใช้งาน เช่น การสั่งซื้อ การเช็คสต๊อก	
ขอบเขตระบบ (System Boundaries) แสดงถึงกลุ่มกรอบที่แบ่งขอบเขตระหว่างระบบกับผู้มีปฏิสัมพันธ์	
การสัมพันธ์ (Association) แสดงถึงเส้นเชื่อมความสัมพันธ์ระหว่างผู้มีปฏิสัมพันธ์กับยูสเคส โดยมีลักษณะเป็นลูกศรเส้นทึบที่ชี้ไปยังยูสเคสที่ระบุไว้	
ส่วนขยาย (Extension) แสดงถึงเส้นเชื่อมของยูสเคสหนึ่ง ๆ ที่เป็นทางเลือกของพฤติกรรมยูสเคส โดยจะชี้ไปยังยูสเคสหลัก	
ส่วนประกอบ (Inclusion) แสดงถึงเส้นเชื่อมของยูสเคสหนึ่ง ๆ ที่เป็นทางเลือกบังคับของพฤติกรรมยูสเคส โดยจะชี้ออกจากยูสเคสหลักไปยังยูสเคสบังคับนั้น ๆ	
ความสัมพันธ์แบบกล่าวสรุป (Generalization) เป็นเส้นเชื่อมที่แสดงถึงการสืบทอดของ ผู้มีปฏิสัมพันธ์ หรือ ยูสเคสหนึ่ง ๆ โดยส่วนหลักที่ถูกสืบทอดจะแสดงถึงพฤติกรรมที่ถูกส่งไปยังส่วนที่สืบทอดนั้น ๆ โดยมีสัญลักษณ์เป็น ลูกศรหัวทึบ ชี้ไปยังส่วนหลักที่ต้องการสืบทอด (parents)	

2.1.1.2 รายละเอียดของแผนภาพยูสเคส (Use case description)

รายละเอียดยูสเคส เป็นส่วนที่ใช้ในการอธิบายถึงรายละเอียดของยูสเคสที่ถูกระบุอยู่ในแผนภาพยูสเคส ประกอบด้วย ชื่อของยูสเคส (Use case name), รหัสของยูสเคส (ID), ระดับความสำคัญ (Important Level), ผู้มีปฏิสัมพันธ์หลัก (Primary Actor), ประเภทของยูสเคส (Use case type), ผู้ที่เกี่ยวข้อง (Stakeholders and Interests), รายละเอียดโดยย่อ (Brief description), ความสัมพันธ์(Association), การไหลของเหตุการณ์ (Basic Flows) ซึ่งจะประกอบด้วย ปฏิสัมพันธ์จากผู้ใช้ (Actor Action) และ การตอบกลับของระบบ (System Response) , การไหลของข้อมูลทางเลือก (Alternative/Exceptional Flow) ดังแสดงตัวอย่างรายละเอียดทั้งหมดในตารางที่ 2 และ รูปที่ 1

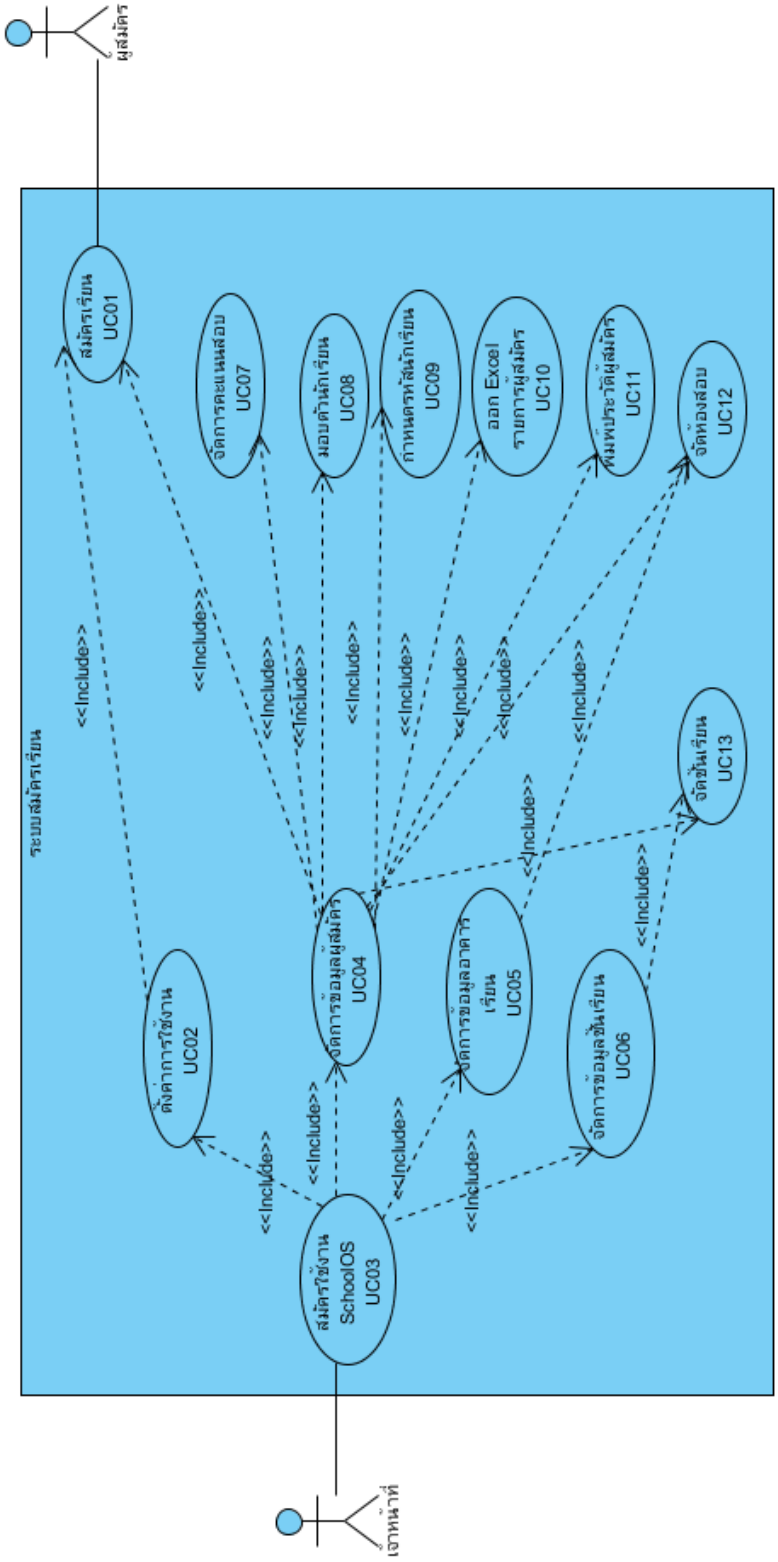
ขั้นตอนภายในส่วนของ Basic Flow และ Alternate/Exeptional Flows แสดงถึงทรานแซคชันของยูสเคส ซึ่งในตารางที่ 2 จำนวนทรานแซคชันรวมกันทั้งหมด 5 ทรานแซคชัน (ขั้นตอน 1 – 4 ของ Basic Flow และ 3.a1 ของ Alternate/Exeptional Flows)

อีกทั้งจำนวนของซีเนริโอของตารางที่ 2 ประกอบด้วย 3 ซีเนริโอ ซึ่งมาจาก 1 เส้นทางหลักใน Basic Flow และ 2 เส้นทางเลือกของ Alternate/Exeptional Flows

ตารางที่ 2 ตัวอย่างข้อมูลของรายละเอียดของแผนภาพยูสเคส

<table>
<tr><td colspan="2">Use case name: ตั้งค่าการใช้งาน</td><td>ID: UC02</td><td>Important Level : High</td></tr>
<tr><td colspan="2">Primary Actor : เจ้าหน้าที่</td><td colspan="2">Use case Type : Detail , Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests : ผู้สมัคร</td></tr>
<tr><td colspan="4">Brief Description : ส่วนที่เจ้าหน้าที่ใช้ในการตั้งค่าข้อมูลการใช้งานระบบในส่วนต่าง ๆ</td></tr>
<tr><td colspan="4">Association : ตั้งค่า</td></tr>
<tr><td colspan="4">Basic Flow of Events</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td colspan="2">1. เลือกเมนูตั้งค่าการใช้งาน</td><td colspan="2">2. ระบบแสดงแบบฟอร์มการตั้งค่า</td></tr>
<tr><td colspan="2">3. กรอกรายละเอียดการสมัคร พร้อมบันทึก</td><td colspan="2">4. ระบบบันทึกข้อมูลการสมัคร</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
3.a กรณี Url โรงเรียนซ้ำ
3.a1 ระบบขึ้นเตือนให้กรอกใหม่อีกครั้ง
3.b ผู้ใช้ยกเลิกการสมัคร</td></tr>
</table>

รูปที่ 1 ตัวอย่างแผนภาพยูสเคสระบบสมัครเรียน



2.1.2 แผนภาพกิจกรรม (Activity Diagram)

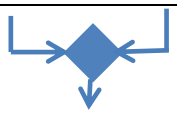
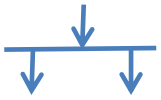
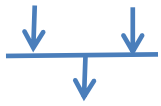
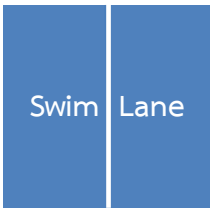
แผนภาพกิจกรรมเป็นแผนภาพที่ใช้ในการอธิบายกิจกรรมที่เกิดขึ้นในลักษณะกระแสการไหลของการทำงาน โดยมีลักษณะเดียวกับแผนภาพผังงาน (Flow Chart Diagram) โดยขั้นตอนในการทำงานทำงานแต่ละขั้นจะเรียกว่ากิจกรรม (Activity) หรือ เหตุการณ์ (Action) ที่ถูกระบุระหว่างจุดเริ่มต้น (Initial) และจุดสิ้นสุดของแผนภาพกิจกรรม ซึ่งแผนภาพกิจกรรมนี้สามารถใช้ในการอธิบายในหลาย ๆ มุมมองของระบบ ตั้งแต่ใช้เพื่ออธิบายการขั้นตอนการคำนวณ ใช้ในการแสดงการเปลี่ยนสถานะของระบบหรือฟังก์ชันการทำงาน เป็นต้น

2.1.2.1 องค์ประกอบของแผนภาพกิจกรรม

การนำแผนภาพกิจกรรมมาใช้ในการอธิบาย จำเป็นต้องใช้สัญลักษณ์ที่ถูกระบุในมาตรฐาน ซึ่งองค์ประกอบต่าง ๆ ได้ถูกกำหนดขึ้น เพื่อแสดงถึงมุมมองต่าง ๆ ของระบบที่ออกแบบไว้เพื่อให้ผู้ออกแบบและผู้ที่ทำความเข้าใจง่ายในการใช้ในการอธิบายถึงภาพรวมของกิจกรรมเหล่านั้นได้มีประสิทธิภาพยิ่งขึ้น ที่ช่วยลดความเข้าใจที่ผิดพลาดของผู้ที่เกี่ยวข้องกับระบบได้มากขึ้น โดยองค์ประกอบของแผนภาพกิจกรรมถูกอธิบายดังตารางที่ 3

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 3 องค์ประกอบของแผนภาพกิจกรรม [12]

องค์ประกอบ	สัญลักษณ์
เหตุการณ์ (Action) แสดงถึงขั้นตอนหน่วยย่อยสุดที่ไม่สามารถแตกย่อยได้	
กิจกรรม (Activity) แสดงถึงกลุ่มของเหตุการณ์	
วัตถุ (Object Node) แสดงถึงกลุ่มของวัตถุที่เกิดขึ้นของแผนภาพกิจกรรม	
กระแสควบคุม (Control Flow) แสดงถึงลำดับของการประมวลผลกิจกรรมหรือเหตุการณ์	
กระแสวัตถุ (Object Flow) แสดงถึงกระแสการไหลของวัตถุที่เกิดจากกิจกรรมหรือเหตุการณ์	
จุดเริ่มต้น (Initial Node) แสดงถึงสถานะจุดเริ่มต้นของแผนภาพกิจกรรม	
จุดสิ้นสุดกิจกรรม (Final Activity Node) แสดงถึงการหยุดกระแสควบคุมหรือกระแสวัตถุทั้งหมด	
จุดสิ้นสุดกระแส (Final Flow Node) แสดงถึงการหยุดกระแสควบคุมหรือกระแสวัตถุหนึ่ง ๆ	
จุดตัดสินใจ (Decision Node) แสดงถึงเงื่อนไขที่ใหลไปในทิศทางใด ทิศทางหนึ่งที่ตรงตามเงื่อนไข	
จุดรวม (Merge Node) แสดงถึงการรวมของเส้นทางที่แยกจากจุดตัดสินใจเพื่อไปยังเส้นทางหลัก	
จุดแยก (Fork Node) แสดงถึงจุดที่แยกเส้นทางคู่ขนานของกลุ่มกิจกรรมหรือเหตุการณ์	
จุดประสาน (Join Node) แสดงถึงการรวมของเส้นทางคู่ขนานของกลุ่มกิจกรรมหรือเหตุการณ์	
สวิมเลน (Swim Lane) แสดงถึงเส้นทางอิสระหลากหลายเส้นทางของแผนภาพกิจกรรม	

2.2 กราฟควบคุมกระแส (Control Flow Graph)

Frances E. Allen [13] ได้เสนอถึงแผนภาพที่ใช้เป็นสัญลักษณ์ในการอธิบายถึงเส้นทางการประมวลผลของโปรแกรมที่ถูกประมวลผล โดยกราฟควบคุมกระแส (Control Flow Graph) ที่ประกอบด้วย 2 องค์ประกอบคือ โหนด (Node) ที่แสดงถึงหน่วยการประมวลผลของชุดรหัสข้อมูลที่ถูกระบุในชุดรหัสข้อมูล (Source Code) และ เอดจ์ (Edge) ที่แสดงถึงทิศทางของของการไหลของโหนดหนึ่งไปยังโหนดหนึ่ง โดยที่ในแต่ละโหนดจะมีจำนวนของเส้นทางเข้า (Indegree) หรือ จำนวนของเส้นทางออก (Outdegree) ซึ่งถูกระบุด้วยจำนวนของเอดจ์ที่เข้าออกของโหนดต่าง ๆ ในแผนภาพกระแส โดยรูปแบบของกราฟควบคุมกระแสมีอยู่ 3 รูปแบบ ดังแสดงในรูปที่ 2 คือ

2.2.1 แผนภาพเชิงลำดับ

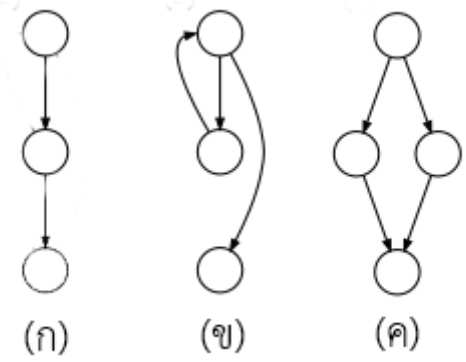
ในรูปที่ 2.ก แสดงถึงแผนภาพที่แสดงถึงการไหลของกระแสในลักษณะเชิงลำดับ โดยในแต่ละโหนด จะมีเส้นทางเข้าหรือออกในทิศทางไปข้างหน้าโดยไม่มีการย้อนกลับไปยังทิศทางเดิมที่ผ่านมาเหล่านั้น

2.2.2 แผนภาพการวนซ้ำ

ในรูปที่ 2.ข แสดงถึงแผนภาพที่แสดงถึงการไหลของกระแสในลักษณะการวนซ้ำ (While Loop) โดยโหนดจะมีเส้นทางออกไปยังโหนดก่อนหน้า และการวนซ้ำจนถึงจุดสิ้นสุดไปยังโหนดต่อไปเมื่อเข้าเงื่อนไขตามที่ชุดรหัสข้อมูลกำหนดไว้

2.2.3 แผนภาพการตัดสินใจ

ในรูปที่ 2.ค แสดงถึงแผนภาพแผนภาพที่แสดงถึงการไหลของกระแสในลักษณะเชิงการตัดสินใจ โดยเส้นทางจะถูกแยกไปยังเส้นทางอื่น ๆ ที่ตรงตามเงื่อนไขที่ถูกระบุในชุดรหัสข้อมูลกำหนดไว้



รูปที่ 2 รูปแบบของกราฟควบคุมกระแส

2.3 การประมาณขนาดของซอฟต์แวร์เชิงฟังก์ชัน

2.3.1 การประมาณขนาดด้วยวิธี Line of Code (LOC)

จำนวนบรรทัดของชุดรหัสซอฟต์แวร์เป็นหนึ่งในมาตรวัดที่นิยมใช้ในการแทนที่ด้วยขนาดของซอฟต์แวร์ โดยการนับจำนวนบรรทัดนั้นมีหลากหลายวิธีในการนับ โดย Vu Nguyen และคณะ [14] ได้เสนอถึงรูปแบบของการนับขนาดจากชุดรหัสซอฟต์แวร์ออกเป็น 3 รูปแบบ ดังนี้

- **Physical Lines of Code (LOC)** เป็นวิธีที่นับจากจำนวนบรรทัดจริงที่ผู้พัฒนาใช้ในการจัดรูปแบบของชุดรหัสซอฟต์แวร์เป็นเกณฑ์ในการกำหนดจำนวนของบรรทัดของชุดรหัสซอฟต์แวร์
- **Logical Lines of Code (LLOC)** เป็นวิธีที่นับจากการทำงานที่ประมวลผลเป็นหลัก โดยไม่สนใจถึงจำนวนบรรทัดที่ผู้พัฒนาจัดเรียงรูปแบบของชุดรหัสจริง
- **Comment Line** เป็นวิธีที่นับจากบรรทัดที่ผู้พัฒนาคอมเมนท์ในการอธิบาย

โดยวิธีที่เป็นที่นิยมคือ การใช้วิธี LOC และ LLOC มาใช้แทนด้วยจำนวนของบรรทัดของชุดรหัสซอฟต์แวร์ โดยขนาดดังกล่าวที่ได้สามารถนำมาใช้ในการประมาณความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธี COCOMO Model เป็นต้น ซึ่งตัวอย่างข้างล่างนี้สามารถนับขนาดด้วย 3 รูปแบบข้างต้นได้ดังนี้ การนับด้วยวิธี LOC จะมีค่าจำนวนบรรทัดเท่ากับ 1, การนับจำนวนบรรทัดด้วยวิธี LLOC มีค่าจำนวนบรรทัดเท่ากับ 2 (For loop และ printf) และการนับด้วยวิธี Comment Line มีค่าเท่ากับ 1 (How many lines of code is this?)

```
for (i = 0; i < 100; i++) printf("hello"); /* How many lines of code is this? */
```

แต่หากเขียนในอีกรูปแบบหนึ่งจำนวนบรรทัดในส่วนของ LOC จะเปลี่ยนไปจากรูปแบบของชุดรหัสที่เปลี่ยนแปลง โดยจำนวนบรรทัดด้วยการนับแบบ LOC จะมีค่าเท่ากับ 5 แต่ LLOC และ Comment Line ยังมีค่าคงเดิม

```
/* How how many lines of code is this? */
for (i = 0; i < 100; i++)
{
    printf("hello");
}
```

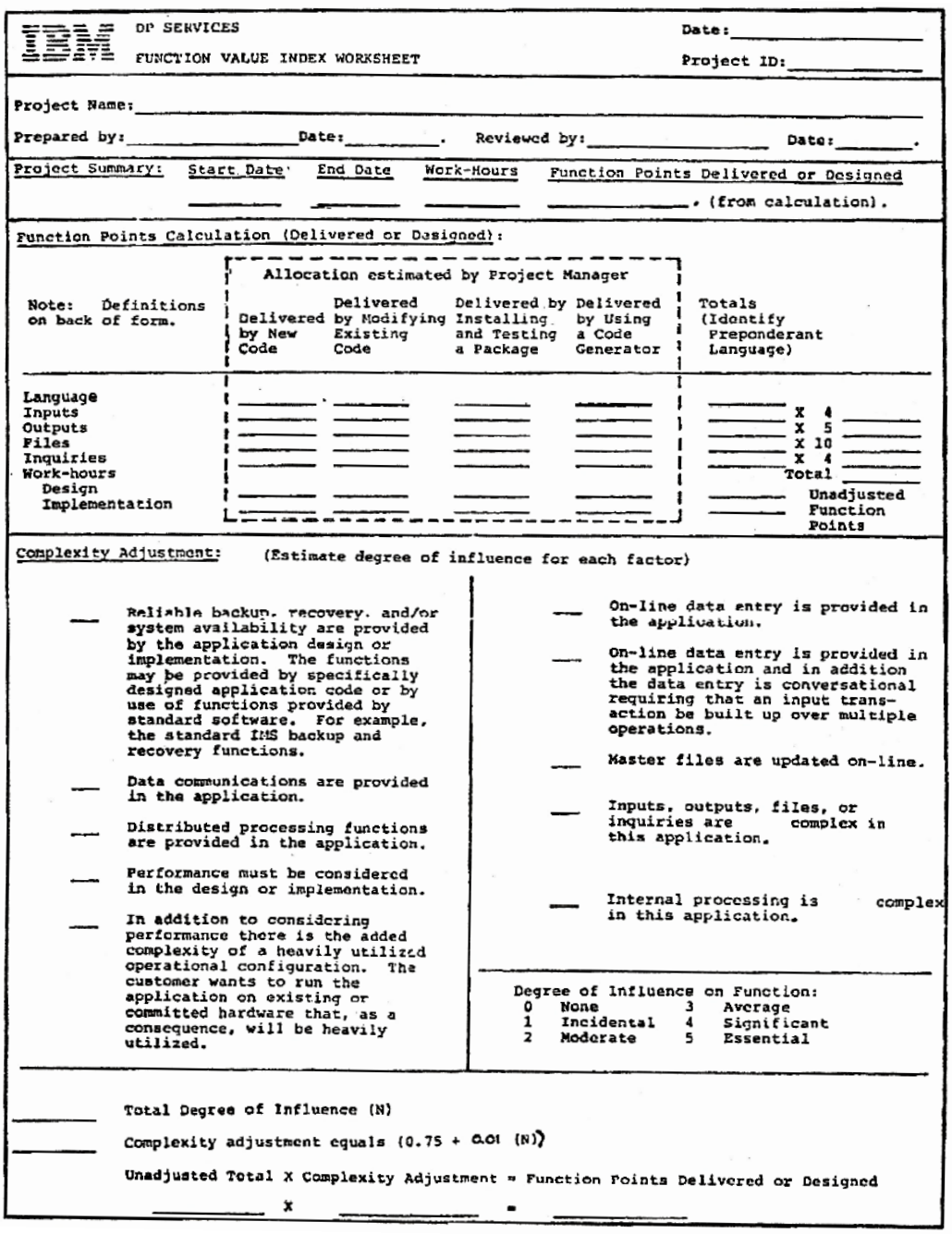
2.3.2 การประมาณขนาดด้วยวิธีฟังก์ชันพอยท์ (Function Point)

ในปี 1979 Allan Albrecht [2, 3] ได้เสนอการประมาณขนาดของซอฟต์แวร์ด้วยวิธีการวิเคราะห์ฟังก์ชันพอยท์ โดยมีวัตถุประสงค์ให้สามารถประมาณขนาดของซอฟต์แวร์ในขั้นตอนแรกของการพัฒนาซอฟต์แวร์ ยืดหยุ่นและไม่ขึ้นอยู่กับภาษาในการพัฒนา โดยแบ่งประเภทความต้องการเชิงฟังก์ชันออกเป็น 4 ประเภท คือ ข้อมูลนำเข้า (Input), ข้อมูลส่งออก (Output), การสอบถามข้อมูล (Inquiries) และ ไฟล์ใช้งานหลัก (Master File) โดยนำจำนวนของแต่ละเภทเหล่านั้นถ่วงน้ำหนักด้วยค่าที่เสนอไว้ และนำผลรวมของแต่ละประเภทฟังก์ชันมารวมกันเป็นขนาดเชิงฟังก์ชันพอยท์ก่อนมีการปรับปรุง (Unadjusted Function Point) ไปสู่การปรับค่าด้วยค่าการปรับปรุงความซับซ้อน (Complexity Adjustment) ที่มีการกำหนดคะแนนของผลกระทบอยู่ 6 ระดับ (ระดับ 0 คือลักษณะจำเพาะนี้ไม่มีผลกระทบกับระบบ จนถึงระดับ 5 แสดงถึงลักษณะจำเพาะนี้มีผลกระทบกับระบบมาก) เพื่อให้ได้มาซึ่งขนาดของซอฟต์แวร์ในหน่วยฟังก์ชันพอยท์ ดังแสดงรายละเอียดในรูปที่ 3 ที่แสดงถึงแบบฟอร์มที่ถูกใช้ในการประมาณด้วยวิธีฟังก์ชันพอยท์

2.3.3 การประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐาน IFPUG

ต่อมาวิธีการประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐาน IFPUG [4] เป็นวิธีที่ได้จากการพัฒนาต่อยอดของวิธีการประมาณขนาดด้วยการวิเคราะห์ฟังก์ชันพอยท์ เพื่อให้มีความเหมาะสมกับโครงการของซอฟต์แวร์ที่ถูกพัฒนาในช่วงยุค ค.ศ. 1980 - 2000 โดยมีการแบ่งประเภทของความต้องการเชิงฟังก์ชันออกเป็น 5 ประเภท คือ แฟ้มข้อมูลเชิงตรรกะภายใน (Internal Logical Files) แฟ้มข้อมูลต่อประสานภายนอก (External Interface Files) ข้อมูลนำเข้าภายนอก (External Inputs) ข้อมูลส่งออกภายนอก (External Outputs) และ การสอบถามภายนอก (External Inquiry) โดยจำนวนของแต่ละประเภทจะถูกนำมาถ่วงน้ำหนัก เพื่อนำค่าที่ถ่วงน้ำหนักของฟังก์ชันทั้ง 5 ประเภทมารวมกันเป็นขนาดเชิงฟังก์ชันพอยท์ก่อนการปรับปรุง และทำการปรับค่าด้วยค่าปัจจัยในการปรับปรุง (Value Adjustment Factor) ที่ประกอบด้วยระดับ 6 ระดับ เช่นเดียวกับวิธีการวิเคราะห์ฟังก์ชันพอยท์ เพื่อให้ได้ผลลัพธ์ขนาดของซอฟต์แวร์ในหน่วยฟังก์ชันพอยท์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



DP SERVICES

FUNCTION VALUE INDEX WORKSHEET

Date: _____

Project ID: _____

Project Name: _____

Prepared by: _____ Date: _____. Reviewed by: _____ Date: _____.

Project Summary: Start Date | End Date | Work-Hours | Function Points Delivered or Designed

_____ | _____ | _____ | _____. (from calculation).

Function Points Calculation (Delivered or Designed):

Note: Definitions on back of form.	Allocation estimated by Project Manager: Delivered by New Code	Delivered by Modifying Existing Code	Delivered by Installing and Testing a Package	Delivered by Using a Code Generator	Totals (Identify Preponderant Language)	
Language	_____	_____	_____	_____	_____	
Inputs	_____	_____	_____	_____	_____	X 4 _____
Outputs	_____	_____	_____	_____	_____	X 5 _____
Files	_____	_____	_____	_____	_____	X 10 _____
Inquiries	_____	_____	_____	_____	_____	X 4 _____
Work-hours						Total _____
Design	_____	_____	_____	_____	_____	Unadjusted Function Points
Implementation	_____	_____	_____	_____	_____	

Complexity Adjustment: (Estimate degree of influence for each factor)

___ Reliable backup, recovery, and/or system availability are provided by the application design or implementation. The functions may be provided by specifically designed application code or by use of functions provided by standard software. For example, the standard IMS backup and recovery functions.

___ Data communications are provided in the application.

___ Distributed processing functions are provided in the application.

___ Performance must be considered in the design or implementation.

___ In addition to considering performance there is the added complexity of a heavily utilized operational configuration. The customer wants to run the application on existing or committed hardware that, as a consequence, will be heavily utilized.

___ On-line data entry is provided in the application.

___ On-line data entry is provided in the application and in addition the data entry is conversational requiring that an input transaction be built up over multiple operations.

___ Master files are updated on-line.

___ Inputs, outputs, files, or inquiries are complex in this application.

___ Internal processing is complex in this application.

Degree of Influence on Function:

0	None	3	Average
1	Incidental	4	Significant
2	Moderate	5	Essential

_____ Total Degree of Influence (N)

_____ Complexity adjustment equals (0.75 + 0.01 (N))

Unadjusted Total X Complexity Adjustment = Function Points Delivered or Designed

_____ X _____ = _____

รูปที่ 3 แบบฟอร์มการประมาณขนาดด้วยวิธีฟังก์ชันพอยท์ [2, 3]

2.3.4 การประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐานอื่น ๆ

ในปี 1998 การประมาณขนาดของซอฟต์แวร์ด้วยมาตรฐาน COSMIC [6, 7] ถูกเสนอขึ้นโดยนำจุดเด่นของวิธีการประมาณขนาดด้วยมาตรฐาน IFPUG [4] NESMA [5] และ Mark II [8] มาใช้ในการนำเสนอวิธีการประมาณขนาดของซอฟต์แวร์ ที่เน้นการเคลื่อนที่ของข้อมูล (Data Movement) เป็นเกณฑ์หลักในการประมาณขนาด ซึ่งมีการกำหนดประเภทของการเคลื่อนที่ของข้อมูลออกเป็น 4 ประเภท ดังอธิบายรายละเอียดภายในตารางที่ 4 เพื่อทำการนับจำนวนของแต่ละประเภทมารวมกันเป็นขนาดของซอฟต์แวร์ในหน่วย COSMIC ดังรูปแบบสมการคำนวณของการประมาณขนาดในหน่วย COSMIC ดังข้างล่างนี้ โดย Size หาได้จากจำนวนของประเภทการไหลข้อมูลนั้น ๆ เช่น Size ($Entry_i$) หาก N = 5 ดังนั้นขนาดในส่วนของข้อมูลประเภท Entry จะเท่ากับ 5

$$COSMIC\ SIZE = \sum_{N}^{i=1} Size(Entry_i) + \sum_{N}^{i=1} Size(Exit_i) + \sum_{N}^{i=1} Size(Read_i) + \sum_{N}^{i=1} Size(Write_i)$$

ตารางที่ 4 ประเภทการเคลื่อนที่ของข้อมูลด้วยมาตรฐานด้วยวิธี COSMIC

COSMIC V.3.0.1	รายละเอียด
การเข้าถึง (Entry)	แสดงถึงผู้ใช้ทำการเข้าถึงระบบ
การสิ้นสุดการใช้ (Exit)	แสดงถึงระบบทำการยกเลิกการใช้ไปยังผู้ใช้
การอ่านข้อมูล (Read)	แสดงถึงการนำข้อมูลมาใช้งานเพื่อทำการประมวลผล และการเขียน
การเขียนข้อมูล (Write)	แสดงถึงการปรับเปลี่ยนแก้ไขข้อมูล

2.4 การประมาณขนาดของซอฟต์แวร์เชิงวัตถุ

2.4.1 การประมาณขนาดด้วยวิธียูสเคสพอยท์

G Karner [9] เสนอถึงการหาขนาดและความพยายามจากแผนภาพยูสเคสด้วยวิธี ยูสเคสพอยท์ โดยแบ่งขั้นตอนการประมาณขนาดของซอฟต์แวร์ออกเป็น 2 ขั้นตอนดังต่อไปนี้

2.4.1.1 จำแนกผู้มีปฏิสัมพันธ์กับระบบ (Classify Actors) เป็นส่วนที่ใช้จำแนกประเภทของผู้ใช้ที่มีปฏิสัมพันธ์กับระบบ มาใช้ในการประมาณขนาดของซอฟต์แวร์ ซึ่งจำนวนของผู้มีปฏิสัมพันธ์จะถูกกำหนดระดับของการปฏิสัมพันธ์ที่แบ่งระดับของการปฏิสัมพันธ์ออกเป็น 3 ระดับ ซึ่งได้จากจำนวนของผู้มีปฏิสัมพันธ์กับระบบที่อยู่ในช่วงของระดับเหล่านั้น ประกอบด้วย เรียบง่าย (Simple) ปานกลาง (Average) และซับซ้อน (Complex) มาใช้ในการคำนวณหาผลรวมของระดับการปฏิสัมพันธ์ของผู้ใช้งานเพื่อนำมารวมเป็นค่าน้ำหนักของผู้ใช้งานที่ยังไม่ได้ปรับปรุง (Unadjusted Actor Weight: UAW) (คำนวณได้จาก จำนวนของผู้ใช้ระดับเรียบง่าย x 1 + จำนวนของผู้ใช้ระดับปานกลาง x 2 + จำนวนของผู้ใช้ระดับซับซ้อน x 3)

2.4.1.2 จำแนกยูสเคส (Classify Use cases) เป็นส่วนที่ใช้ในการจำแนกประเภทของยูสเคส โดยใช้ช่วงของจำนวนของทรานแซ็คชันเป็นเกณฑ์ในการกำหนดระดับ โดยแบ่งออกเป็น 3 ระดับ ซึ่งได้จากจำนวนของยูสเคสที่มีจำนวนทรานแซคชันอยู่ในช่วงของระดับเหล่านั้น ประกอบด้วย เรียบง่าย (Simple), ปานกลาง (Average) และ ซับซ้อน (Complex) เพื่อทำการคำนวณหาผลรวมของระดับยูสเคสมาเป็นค่าน้ำหนักยูสเคสที่ยังไม่ได้ปรับปรุง (Unadjusted Use Case Weight: UUCW) (คำนวณได้จาก จำนวนของยูสเคสระดับเรียบง่าย x 5 + จำนวนของยูสเคสระดับปานกลาง x 10 + จำนวนของยูสเคสระดับซับซ้อน x 15)

ผลรวมของ UAW และ UUCW แสดงถึงขนาดของยูสเคสพอยท์ก่อนได้รับการปรับปรุง (Unadjusted Use Case Point: UUCP) เพื่อนำมาถ่วงน้ำหนักด้วยปัจจัยความซับซ้อนทางเทคนิค (Technical Complexity Factor: TCF) และปัจจัยความซับซ้อนทางสภาพแวดล้อม (Environmental Complexity Factor: ECF) ที่มีการกำหนดคะแนนของผลกระทบต่อขนาดของซอฟต์แวร์อยู่ 6 ระดับ โดยที่ ระดับ 0 คือลักษณะจำเพาะนี้ไม่มีผลกระทบกับระบบ จนถึงระดับ 5 ที่แสดงถึงลักษณะจำเพาะนี้มีนัยที่มีผลต่อระบบมาก เพื่อแทนค่าผลลัพธ์ขนาดของซอฟต์แวร์ในหน่วยยูสเคสพอยท์ โดยรูปแบบของการประมาณขนาดและความพยายามในหน่วยของ Use case ในรูปของสมการข้างล่าง โดยการคำนวนความพยายามในการพัฒนาซอฟต์แวร์ Karner ได้แนะนำให้ใช้จำนวนชั่วโมง 20 ชั่วโมงการ UCP

$$\mathrm{UCP} = \mathrm{UUCP} * \mathrm{TCF} * \mathrm{EF}$$

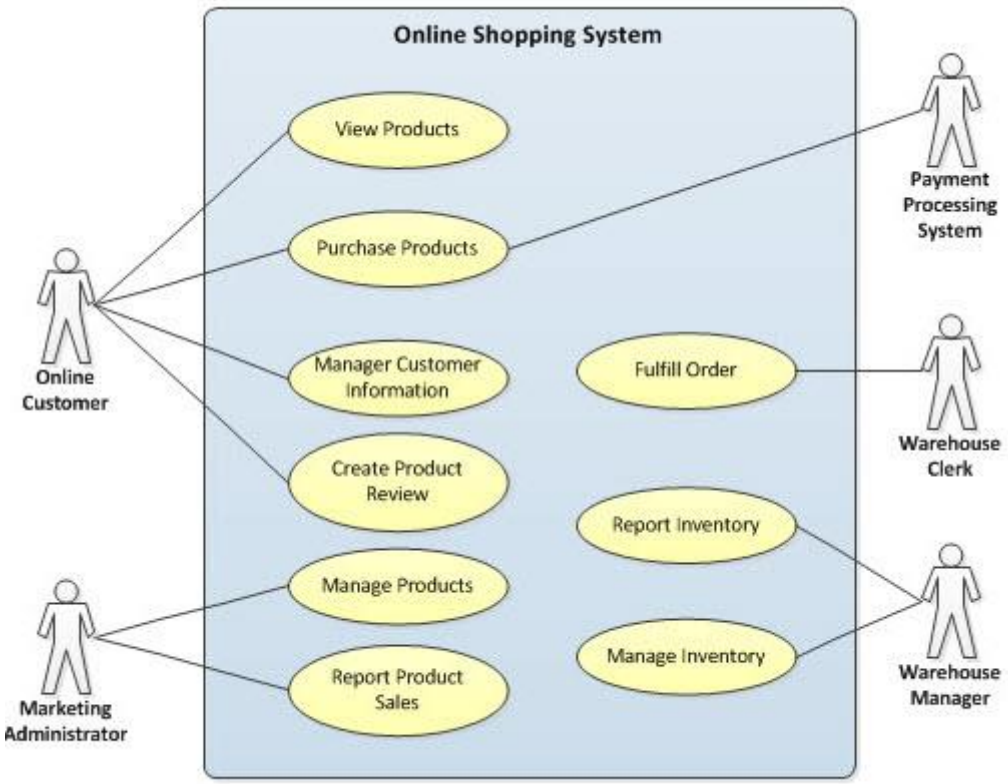
$$EFFORT = UCP * Hours\ per\ UCP$$

ตัวอย่างการประมาณขนาดด้วยวิธียูสเคสพอยท์

จากรูปที่ 4 แสดงถึงแผนภาพยูสเคสของระบบ Online Shopping ประกอบด้วย 9 ยูสเคส สมมติว่าเมื่อทำการแยกประเภทของแต่ละยูสเคสโดยใช้จำนวนของทรานแซ็คชันเป็นเกณฑ์ สามารถแบ่งได้ดังนี้ จำนวนยูสเคส 2 ยูสเคสในระดับเรียบง่าย, จำนวน 3 ยูสเคสในระดับปานกลาง, และจำนวน 4 ยูสเคสในระดับซับซ้อน ดังนั้น ค่า UUCW มีค่าเท่ากับ 2x5 + 3x10 + 4x15 ซึ่งมีค่าเท่ากับ 100

หากพิจารณาในส่วนของผู้มีปฏิสัมพันธ์กับระบบ สามารถจำแนกได้ดังนี้ 1 จำนวนผู้ใช้ระดับเรียบง่าย และ 4 จำนวนผู้ใช้ระดับซับซ้อน ดังนั้น ค่า UAW มีค่าเท่ากับ 1x1 + 0x2 + 4x3 ซึ่งมีค่าเท่ากับ 13

สมมติว่าค่าปัจจัยความซับซ้อนทางเทคนิคมีค่าเท่ากับ 1.02 และ ค่าปัจจัยความซับซ้อนทางสภาพแวดล้อมเท่ากับ 1.085 ดังนั้น ขนาดของซองซอฟต์แวร์ในหน่วยยูสเคสพอยท์มีค่าเท่ากับ (UUCW + UAW) x TCF x ECF หรือ (100 + 13) x 1.02 x 1.085 = 125.06



รูปที่ 4 แผนภาพยูสเคสระบบ Online Shopping [15]

2.4.2 การประมาณขนาดด้วยวิธีซีเควนซ์พอยท์

A. Sellami และคณะ [10] เสนอถึงวิธีการประมาณขนาดเชิงโครงสร้าง (Structural Size) ด้วยแผนภาพเชิงลำดับ (Sequence Diagram) เป็นเกณฑ์ในการประมาณขนาดของซอฟต์แวร์ โดยจำแนกประเภทของขนาดซอฟต์แวร์เป็น 2 ประเภทคือ ขนาดเชิงฟังก์ชัน (Functional Size) ที่หาได้จากจำแนกประเภทของการเคลื่อนที่ของข้อมูลภายในแผนภาพเชิงลำดับโดยใช้วิธี COSMIC [6, 7] มาใช้ในการแยกประเภทของการไหลของข้อมูล และขนาดเชิงโครงสร้างที่หาได้จากจำนวนของเส้นทางเลือก (Alternative) จำนวนของเส้นทางวนซ้ำ (Looping) เพื่อที่จะนำขนาดซอฟต์แวร์ของประเภททั้ง 2 นั้นมาคำนวณหาขนาดของซอฟต์แวร์ในหน่วยของ Control Structure Manipulation (CSM)

2.4.3 การประมาณขนาดด้วยวิธีคลาสพอยท์

G. Costagliola และคณะ [11] เสนอถึงการประมาณขนาดของซอฟต์แวร์ด้วยคลาสพอยท์ โดยใช้แผนภาพคลาส (Class Diagram) เป็นเกณฑ์ในการประมาณขนาด โดยแบ่งลำดับการประมาณออกเป็น 3 ขั้นตอนคือ

2.4.3.1 จำแนกประเภทของคลาส ออกเป็น 4 ประเภท คือ ขอบเขตการแก้ปัญหา (Problem Domain) การปฏิสัมพันธ์ของผู้ใช้ (Human Interaction) การจัดการข้อมูล (Data Management) และ การจัดการลำดับงาน (Task Management Type)

2.4.3.2 กำหนดระดับความซับซ้อนของคลาสในแต่ละประเภทข้างต้น เพื่อนำมาถ่วงน้ำหนักตามระดับความซับซ้อน สำหรับผลรวมทั้งหมดแทนถึงขนาดของคลาสพอยท์ก่อนปรับค่า (Unadjusted Class Point)

2.4.3.3 ปรับปรุงค่าขนาดของซอฟต์แวร์ก่อนการปรับปรุง ด้วยปัจจัยในการปรับค่าซึ่งประกอบด้วย ปัจจัยความซับซ้อนทางกระบวนการ (Processing Complexity Factor) และ ปัจจัยความซับซ้อนทางเทคนิค (Technical Complexity Factor) เพื่อให้ได้มาซึ่งขนาดของซอฟต์แวร์ในหน่วยคลาสพอยท์

2.5 การประมาณความพยายามในการพัฒนาซอฟต์แวร์

ในงานวิจัย [16, 17] ได้กล่าวถึงการประมาณความพยายาม เป็นหนึ่งในขั้นตอนการพัฒนาซอฟต์แวร์ เพื่อใช้ในการวางแผนจัดการโครงการทางด้านระยะเวลา และค่าใช้จ่าย โดยแบ่งรูปแบบในการประมาณดังต่อไปนี้

2.5.1 Algorithmic Models

เป็นรูปแบบการประมาณโดยใช้สมการคณิตศาสตร์ จากการสร้างสมการด้วยข้อมูลทางสถิติ เพื่อให้ได้แบบจำลองในการทำนายค่าประมาณความพยายาม ซึ่งมีวิธีในการประมาณที่เสนอหลากหลายอาทิเช่น

2.5.1.1 การประมาณแบบพื้นฐาน (Basic Effort Estimation) เป็นรูปแบบการประมาณความพยายามในรูปแบบของ $E = aS^B$ โดยที่ E แทนด้วยความพยายาม a แทนด้วยค่าคงที่ที่โดยขึ้นอยู่กับประเภทของซอฟต์แวร์ที่พัฒนาและวิธีที่ใช้ในการสร้างแบบจำลอง S แทนด้วยขนาดของซอฟต์แวร์ที่หาได้จากวิธีการฟังก์ชันพอยท์หรือออบเจ็คพอยท์ และ B แทนด้วยค่าคงที่ของขนาด

2.5.1.2 แบบจำลองโคโคโม่ (COCOMO Model) เป็นรูปแบบที่ได้รับความนิยมในการประมาณความพยายาม ด้วยการใช้ขนาดของซอฟต์แวร์ทั้งในรูปของจำนวนบรรทัด หรือจำนวนการทำงาน มาใช้ในการประมาณในช่วงเริ่มต้นของการพัฒนา โดยมีรูปแบบสมการในการประมาณคือ $pm = A(size)^{(sf)} \times(em_1) \times(em_2) \times(em_3)$.... โดยที่ pm แทนด้วย ความพยายามที่ทำได้ในระยะหนึ่งเดือน A แทนด้วยค่าคงที่ซึ่งมีค่าเท่ากับ 2.94 size แทนด้วยจำนวนบรรทัดหรือจำนวนฟังก์ชันที่ถูกพัฒนา sf (Scale Factor) แทนด้วย ปัจจัยค่าคงที่ 5 ปัจจัยที่มีผลต่อขนาดซอฟต์แวร์ และ Em (Effort) แทนด้วย ปัจจัยความพยายาม 17 ปัจจัย

2.5.1.3 แบบจำลอง Putnam's SLIM [18] เสนอโดย Putnam เป็นหนึ่งในวิธีในการประมาณความพยายาม ที่ได้จากการทำการทดลองและทดสอบด้วยโครงการภายในกองทัพ ซึ่งได้ผลลัพธ์ของแบบจำลองการประมาณที่อยู่ในรูปแบบของ $Effort = [Size / Productivity \times Time^{4/3}]^3 \times B$ โดยที่ Size แทนด้วย ขนาดของซอฟต์แวร์, Productivity แทนด้วย ความสามารถขององค์กรในการพัฒนาซอฟต์แวร์ Time แทนด้วย ระยะเวลาที่ต้องใช้ในการพัฒนาในหน่วยปี

2.5.2 Expert Judgment

เป็นรูปแบบการประมาณจากผู้เชี่ยวชาญ ด้วยประสบการณ์จากการจัดการงานก่อนหน้าผนวกกับปัจจัยอื่น ๆ ที่ได้ทำจากการเก็บข้อมูลมาใช้ในการประมาณความพยายามของระบบที่พัฒนา โดยเป็นเทคนิคที่ใช้ผู้เชี่ยวชาญที่เป็นที่ปรึกษาโครงการ ที่ผู้เชี่ยวชาญแต่ละคนทำการประเมินต้นทุนและนำเข้าที่ประชุมเพื่อทำการเปรียบเทียบ เพื่อทำการตกลงและกำหนดเป็นต้นทุนของการพัฒนาระบบนั้น ๆ ต่อไป

2.5.3 Machine Learning Methods

เป็นรูปแบบการสร้างวิธีการประมาณจากการเรียนรู้ข้อมูลในอดีตมาประมวลผลด้วยวิธีทางสถิติ ให้ได้แบบจำลองการประมาณจากกลุ่มข้อมูลที่ใช้ในการสร้างแบบจำลอง (Training Set) ซึ่งประกอบด้วยวิธีในการประมาณต่าง ๆ อาทิเช่น

2.5.3.1 แบบจำลองโครงข่ายประสาทเทียม (Neural Network Model) [19] เป็นรูปแบบการหาแบบจำลองทางคณิตศาสตร์จากการจำลองสมองมนุษย์ ซึ่งประกอบด้วย เซลล์ประสาท (Neuron) และ จุดประสานประสาท (Synapses) โดยที่แต่ละเซลล์จะประกอบด้วยปลายเส้นประสาท เรียกว่า เดนไดรท์ (Dendrite) ที่เป็นช่องทางนำเข้าของข้อมูล และส่งออกไปยังปลายทาง เรียกว่า แอคซอน (Axon) ทางคอมพิวเตอร์ก็เช่นเดียวกันที่ข้อมูลนำเข้าประกอบด้วยน้ำหนักในแต่ละชุดข้อมูล ซึ่งข้อมูลนำเข้าที่ถ่วงด้วยค่าน้ำหนักจะถูกนำมารวมกัน และเปรียบเทียบกับค่าเทรสโฮลด์ (Threshold) เพื่อคำนวณหาค่าของข้อมูลนำเข้าที่น้อยที่สุดที่ยอมรับ ในการส่งไปเป็นข้อมูลนำเข้าของเซลล์ประสาทอื่น ๆ ต่อไป

2.5.4 Fuzzy Logic and Evolutionary Algorithms, Genetic Programming

เป็นรูปแบบการประมาณโดยใช้เหตุผลประมาณ โดยผลลัพธ์ที่ได้จะอยู่ในรูปการอธิบายในเชิงของภาษาจากการแบ่งกลุ่มตามค่าใกล้เคียง เช่น ดีมาก ดี ปานกลาง แย่, ต่ำ ค่อนข้างต่ำ ปานกลาง และสูงมาก เป็นต้น ซึ่งการอธิบายผลลัพธ์ลักษณะนี้จะทำให้ง่ายต่อการอธิบายให้แก่ผู้ใช้ตามค่าที่มีความสัมพันธ์กับเซ็ตวิภัชนัย (Fuzzy Set) โดยเริ่มต้นจะนำชุดข้อมูลทดสอบมากำหนดคุณลักษณะ (Attributes) ของซอฟต์แวร์นั้น ๆ เช่น จำนวนของทีมผู้พัฒนา ประสบการณ์ของทีมพัฒนา เป็นต้น เพื่อมาทำการแบ่งกลุ่มออกเป็นเซ็ตวิภัชนัย ด้วยวิธี Fuzzy C-Means เพื่อนำกลุ่มของเซ็ตวิภัชนัยมาสร้างฟังก์ชันการประมาณทางสถิติด้วยวิธี Real Coded Genetic Algorithm ในการทำการวิเคราะห์ ความคลาดเคลื่อนสัมพัทธ์ และ คุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.20 ของคุณลักษณะของข้อมูลทดสอบนั้น ๆ [20]

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

รายละเอียดในบทนี้ กล่าวถึงงานวิจัยที่เกี่ยวข้อง ที่มีข้อมูลที่จะถูกนำมาประยุกต์ใช้ในงานวิจัย เพื่อให้เกิดคุณภาพในการสร้างแบบจำลองให้สมบูรณ์และมีประสิทธิภาพมากยิ่งขึ้น

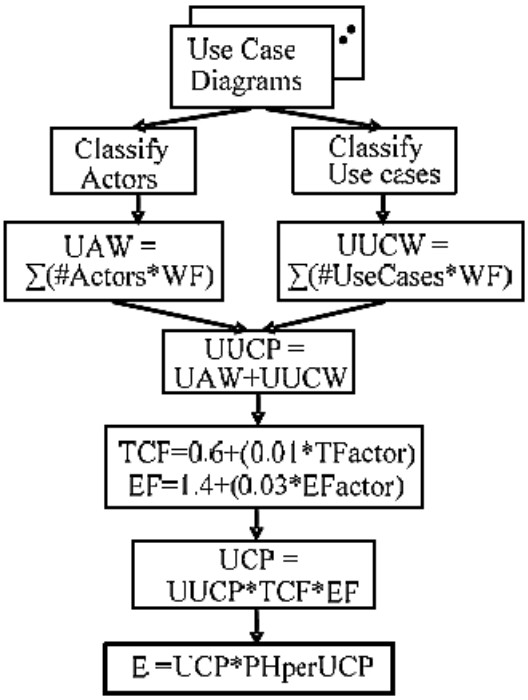
3.1 งานวิจัย “Metrics for Objectory” [9]

งานวิจัยของ G Karner เสนอถึงการหาขนาดและความพยายามจากแผนภาพยูสเคส ด้วยวิธียูสเคสพอยท์ โดยนำแผนภาพยูสเคส มาจำแนกประเภทออกเป็น 2 ประเภทคือ จำแนกผู้มีปฏิสัมพันธ์กับระบบ (Classify Actors) จำแนกยูสเคส (Classify Use cases) เพื่อใช้ในการหาขนาดของซอฟต์แวร์จากปัจจัยเหล่านั้น

ในส่วนของปัจจัยที่จำแนกผู้มีปฏิสัมพันธ์กับระบบ ใช้เกณฑ์รูปแบบการเข้าถึงของผู้ใช้งานในการกำหนดถึงผลกระทบที่ส่งผลต่อขนาดของซอฟต์แวร์ที่อยู่ในลักษณะของค่าน้ำหนักของผู้ใช้งาน (Actor Weight) อาทิเช่น การเข้าถึงระบบด้วย API การเข้าถึงระบบด้วยโปรโตคอล หรือ การเข้าถึงระบบด้วยหน้าจอการใช้งาน ซึ่งผลลัพธ์ที่ได้จากการวัดขนาดด้วยประเภทของผู้มีปฏิสัมพันธ์กับระบบจะแทนด้วย ค่าน้ำหนักของผู้มีปฏิสัมพันธ์กับระบบก่อนการปรับปรุงค่า (Unadjusted Actor Weight: UAW)

สำหรับการจำแนกยูสเคส จะถูกระบุประเภทโดยใช้เกณฑ์ของจำนวนทรานแซ็คชันในการจำแนกถึงผลกระทบที่ส่งผลต่อขนาดของซอฟต์แวร์ที่อยู่ในลักษณะของค่าน้ำหนักยูสเคสก่อนการปรับปรุง (Unadjusted Use Case Weight: UUCW)

ผลรวมของ UAW และ UUCW จะแสดงถึงขนาดของยูสเคสพอยท์ก่อนได้รับการปรับปรุง (Unadjusted Use case Point) เพื่อนำมาถ่วงน้ำหนักด้วยปัจจัยความซับซ้อนทางเทคนิคและปัจจัยความซับซ้อนทางสภาพแวดล้อมเพื่อให้ได้ผลลัพธ์ขนาดของซอฟต์แวร์ในหน่วยยูสเคสพอยท์ โดยลำดับขั้นตอนในการประมาณถูกอธิบายดังแสดงในรูปที่ 5

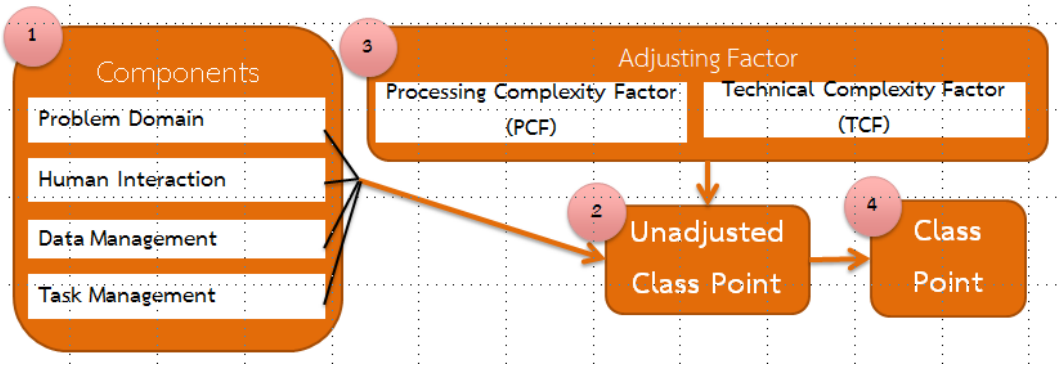


รูปที่ 5 กระบวนการประมาณขนาดด้วยยูสเคสพอยท์ [9]

3.2 งานวิจัย "Class point: an approach for the size estimation of object-oriented systems," [11]

G. Costagliola และคณะ เสนอถึงการหาขนาดของซอฟต์แวร์ด้วยวิธีคลาสพอยท์ โดยใช้แผนภาพคลาส (Class Diagram) เป็นเกณฑ์ในการประมาณขนาด โดยเริ่มจากการออกแบบแผนภาพคลาสของระบบขึ้น และทำการจำแนกประเภทของคลาสทั้งหมดออกเป็นหมวดหมู่ ซึ่งแบ่งออกเป็น 4 ประเภทคือ Problem Domain, Human interaction, Data Management, และ Task Management ซึ่งแต่ละประเภทของคลาสเหล่านั้นจะถูกกำหนดความซับซ้อนและนำมาปรับค่าด้วยน้ำหนักของระดับความซับซ้อนเหล่านั้นต่อไป

ค่าของคลาสทั้ง 4 ประเภทที่ถูกถ่วงน้ำหนัก จะถูกนำมารวมกันที่แสดงถึงขนาดของคลาสที่ยังไม่ได้แก้ไข (Unadjusted Class Point: UCP) เพื่อนำมาถ่วงค่าน้ำหนักด้วยปัจจัยความซับซ้อนด้านกระบวนการ (Processing Complexity Factor: PCF) และ ปัจจัยความซับซ้อนทางด้านเทคนิค (Technical Complexity Factor) เพื่อให้ได้ผลลัพธ์ขนาดของซอฟต์แวร์ในหน่วยคลาสพอยท์ โดยมีรูปแบบขั้นตอนในการประมาณดังแสดงในรูปที่ 6



รูปที่ 6 ขั้นตอนการประมาณขนาดซอฟต์แวร์ด้วยวิธีคลาสพอยท์

3.3 งานวิจัย "Transactions and paths: Two use case based metrics which improve the early effort estimation," [21]

Robiolo และคณะ เสนอถึงการปรับปรุงวิธีการประมาณขนาดด้วยวิธียูสเคสพอยท์โดยการใช้ข้อมูลนำเข้าเป็น เส้นทาง (Path) และ ทรานแซ็คชัน (Transaction) ที่ระบุภายในรายละเอียดของยูสเคสเป็นเกณฑ์ข้อมูลนำเข้าหลัก แทนที่การใช้เพียงทรานแซ็คชัน และจำนวนผู้ใช้งานเท่านั้น โดยทำการทดสอบผลกระทบของปัจจัยต่าง ๆ ว่าปัจจัยใดมีส่งผลต่อขนาดของซอฟต์แวร์มากกว่ากัน ด้วยการนำโครงการขนาดเล็ก 13 โครงการในบริษัทที่อยู่ระดับมาตรฐานกระบวนการ CMMI ระดับ 4 มาทดสอบ ซึ่งปัจจัยที่นำมาใช้เปรียบเทียบประกอบด้วย จำนวนของทรานแซ็คชัน จำนวนของเส้นทาง ค่าขนาดฟังก์ชันพอยท์ก่อนการปรับปรุง และ ค่าขนาดยูสเคสพอยท์ก่อนการปรับปรุง

ปัจจัยทั้ง 4 ที่นำมาใช้ จะถูกพิจารณาด้วยการประมาณขนาดของความพยายามที่คาดหวังเพื่อนำมาเปรียบเทียบกับความพยายามในการพัฒนาซอฟต์แวร์จริงของแต่ละโครงการด้วยเทคนิควิธีการทางสถิติ ซึ่งประกอบด้วย Z-mean, Z Standard Deviation, Root Mean Square Error (RMSE) และ Spearman Correlation ซึ่งผลลัพธ์ค่าสหสัมพันธ์ที่มีความสัมพันธ์ที่มีผลผลกระทบสูงสุดเมื่อเทียบกับค่าความพยายามจริงคือ จำนวนของทรานแซ็คชัน และ จำนวนของเส้นทาง ที่ให้ค่าเท่ากับ 0.82 และ 0.86 ตามลำดับ โดยที่จำนวนของเส้นทางมีค่าสหสัมพันธ์ที่มีผลต่อความพยายามจริงสูงที่สุด ตามด้วยจำนวนทรานแซ็คชัน

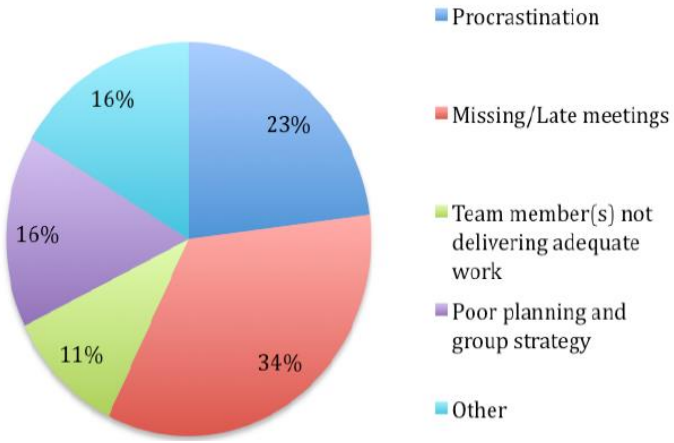
3.4 งานวิจัย “A Complexity Measure” [22]

Thomas J. McCabe, Sr เสนอถึงการประมาณเชิงปริมาณในการหาความซับซ้อนของทิศทางชุดรหัสข้อมูลด้วยวิธี Cyclomatic complexity เพื่อใช้ในการแทนถึงหน่วยวัดของซอฟต์แวร์ที่ระบุถึงความซับซ้อนของโปรแกรมที่พัฒนาขึ้น โดยใช้แผนภาพกระแส (Control Flow Graph) ในการประมาณความซับซ้อน โดยใช้โหนดและเอดจ์ในการคำนวณ ซึ่งคำนวณถูกแสดงดังสมการข้างล่างนี้ โดยที่ E แทนด้วยจำนวนของเอดจ์, N แทนด้วยจำนวนของโหนด และ P แทนด้วยจำนวนของคอมโพเนนท์ที่เชื่อมต่อกันในแผนภาพ ซึ่งโดยปกติแล้วค่า P จะเท่ากับหนึ่งเสมอ

$$v(G) = E - N + 2P$$

3.5 งานวิจัย "The Impact of Non-Technical Factors on Software Architecture," [23]

H. Madhavji และคณะ เสนอถึงผลกระทบที่ส่งผลต่อการพัฒนาซอฟต์แวร์ที่ไม่ใช่เพียงแค่ปัญหาทางด้านเทคนิคเท่านั้น แต่ปัจจัยที่ไม่ใช่ทางเทคนิคก็มีผลต่อการพัฒนาซอฟต์แวร์เช่นเดียวกัน โดยทำการทดสอบจากกลุ่มนักศึกษา 15 ทีม ที่แต่ละทีมประกอบด้วยสมาชิก 4 คน เพื่อให้ทำการพัฒนาสถาปัตยกรรมระบบในรายวิชาการวิเคราะห์ระบบ (System Analysis) ของทางมหาวิทยาลัยที่ผู้จัดทำงานวิจัยสังกัดอยู่ โดยรายละเอียดของระบบดังกล่าวมีความต้องการระบบที่เหมือนกัน เพื่อทำการระบุถึงปัจจัยทางคุณภาพ (Quality factors) ที่ส่งผลโดยตรงต่อโครงสร้างทางสถาปัตยกรรมซอฟต์แวร์ในมุมมองของปัญหาของคน (People Issues) ซึ่งผลลัพธ์ที่ได้ของงานวิจัยนี้ที่ระบุถึงปัจจัยที่ส่งผลกระทบต่อการพัฒนาซอฟต์แวร์ ประกอบด้วย การขาดประชุมหรือมาสายถึง 34% การผลัดวันประกันพรุ่ง 23% การวางแผนและการกำหนดกลยุทธ์ที่ไม่มีประสิทธิภาพ 16% การส่งงานที่ไม่ครบตามความต้องการ 11% และอื่น ๆ 16% ดังอธิบายในรูปที่ 7



รูปที่ 7 ปัจจัยที่ไม่ใช่ทางเทคนิคที่ส่งผลต่อการพัฒนาซอฟต์แวร์ [23]

3.6 งานวิจัย "An Alternative Method Employing Uses Cases for Early Effort Estimation," [24]

G Robiolo และคณะ เสนอถึงการเปรียบเทียบปัจจัยของแผนภาพยูสเคส เพื่อทำการระบุปัจจัยที่ส่งผลต่อขนาดและนำไปใช้ในการประมาณความพยายามในการพัฒนาซอฟต์แวร์ เพื่อให้ผู้ประมาณสามารถทราบถึงความพยายาม ที่จะถูกใช้ในการพัฒนาซอฟต์แวร์เหล่านั้น ได้ตั้งแต่ในช่วงต้น ๆ ของกระบวนการพัฒนาซอฟต์แวร์ โดยปัจจัยที่ถูกนำมาใช้ในการเปรียบเทียบถึงผลกระทบต่อขนาดของซอฟต์แวร์ ประกอบด้วย จำนวนทรานแซ็คชัน และเอนทิตีออบเจ็คของแผนภาพยูสเคส โดยใช้ค่าความคลาดเคลื่อนสัมพัทธ์ (Magnitude of Relative Error) และ ค่าคุณภาพของการประมาณที่อัตราความผิดพลาดต่ำกว่า 25% (Predictive Quality: PRED(25)) ซึ่งหมายถึง คุณภาพของแบบจำลองนี้จะต้องมีอัตราความถูกต้องที่สูงหรือเท่ากับ 75% ของค่าความพยายามในการพัฒนาซอฟต์แวร์จริงที่ถูกใช้ในการเปรียบเทียบ โดยผลลัพธ์ที่ได้ จำนวนของทรานแซ็คชันส่งผลกระทบต่อการเพิ่มของขนาดของซอฟต์แวร์และความพยายามในการพัฒนาที่สูงสุด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
การออกแบบและวิธีการประมาณความพยายามด้วยแอคทิวิตีพอยท์

ในบทนี้กล่าวถึงการออกแบบวิธีในการประมาณความพยายามด้วยแอคทิวิตีพอยท์ โดยใช้แผนภาพกิจกรรมเป็นเกณฑ์ในการระบุปัจจัยที่ส่งผลกระทบต่อขนาดของซอฟต์แวร์ โดยแบ่งออกเป็นคำนิยามและมุมมองของแผนภาพกิจกรรม การแปลงแผนภาพกิจกรรมเป็นกราฟควบคุมกระแสภาพรวมการประมาณด้วยแอคทิวิตีพอยท์ วิธีการประมาณด้วยแอคทิวิตีพอยท์ และการกำหนดค่าน้ำหนักที่ถูกใช้ในงานวิจัยนี้

4.1 คำนิยามและมุมมองแผนภาพกิจกรรม

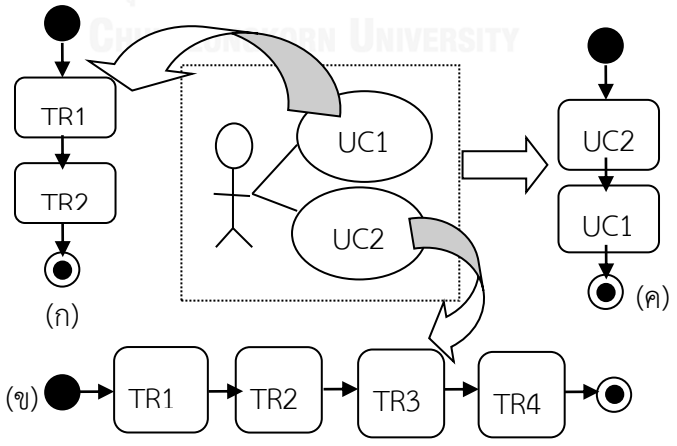
งานวิจัยนี้ได้เสนอถึงการออกแบบการประมาณขนาดจากแผนภาพกิจกรรมในหน่วยแอคทิวิตีพอยท์ จากการนำแผนภาพยูสเคสมาออกแบบ โดยมีวัตถุประสงค์เพื่อใช้ในการประมาณขนาดของซอฟต์แวร์เพื่อใช้ในการเป็นข้อมูลนำเข้าในการประมาณความพยายาม ซึ่งแผนภาพกิจกรรมจะถูกจำแนกออกเป็น 2 มุมมอง ประกอบด้วย มุมมองระดับบน (High Level) ที่แสดงถึงขนาดในระดับของการปฏิสัมพันธ์ระหว่างยูสเคส และมุมมองระดับล่าง (Low Level) ที่แสดงถึงขนาดในระดับของการปฏิสัมพันธ์ภายในของแต่ละยูสเคสในส่วนของทรานแซคชันและซีเนริโอ โดยแต่ละมุมมองจะถูกแยกในการคำนวณขนาดออกจากกัน เพื่อไม่เป็นการประมาณที่ทับซ้อนกันและแสดงถึงขนาดของซอฟต์แวร์ที่ต้องใช้ในการพัฒนาที่แตกต่างกันอีกด้วย

จากรูปที่ 8 แสดงถึงมุมมองของแผนภาพกิจกรรมที่ออกแบบโดยใช้แผนภาพยูสเคสเป็นเกณฑ์ ในการออกแบบแผนภาพกิจกรรมในแต่ละระดับ ซึ่งมุมมองในแต่ละระดับนั้นจะมีวิธีการประมาณขนาดที่แตกต่างกัน โดยใช้ปัจจัยที่เหมาะสมกับการประมาณขนาดในระดับนั้น ๆ เพื่อลดการประมาณที่ซ้ำซ้อน ที่ส่งผลสู่การประมาณที่คลาดเคลื่อนจากขนาดที่ได้จริงหลังจากซอฟต์แวร์ได้ถูกพัฒนาขึ้น โดยมีมุมมองทั้ง 2 ระดับดังต่อไปนี้

4.1.1 มุมมองระดับล่าง (Low Level) ดังรูปที่ 8.ก และ 8.ข แสดงถึงแผนภาพกิจกรรมในมุมมองระดับล่าง ที่แสดงถึงลำดับขั้นตอนการทำงานของแต่ละยูสเคสในระดับทรานแซ็คชัน โดยการพัฒนาซอฟต์แวร์ในระดับนี้เป็นการพัฒนาในส่วนของรายละเอียดของฟังก์ชัน ซึ่งอาจจะอยู่ในระดับของคลาสหรือเมทอดที่ทำงานเฉพาะในความต้องการที่กำหนดไว้โดยเฉพาะ ซึ่งแต่ละส่วนของขนาดในแต่ละฟังก์ชันนั้นจะแทนด้วยขนาดในแต่ละยูสเคสภายในแผนภาพยูสเคสนั่นเอง โดยที่สัญลักษณ์ในส่วนของกิจกรรม (Activity) หรือการกระทำ (Action) จะถูกแทนด้วยทรานแซ็คชันที่แสดงภายในรายละเอียดของ

ยูสเคสในส่วนของลำดับการทำงานต่าง ๆ ของยูสเคสนั้น ๆ และอีกปัจจัยหลักที่ถูกใช้ระบุถึงเส้นทางที่เป็นไปได้ของยูสเคสนั้น ๆ ว่ามีเส้นทางที่เป็นไปได้ในทิศทางใดบ้าง ดังที่ถูกแสดงในส่วนของรายละเอียดยูสเคสที่ถูกแทนด้วยซีเนริโอของยูสเคสนั้น ๆ

4.1.2 มุมมองระดับบน (High Level) ดังรูปที่ 8.ค แสดงถึงแผนภาพกิจกรรมในมุมมองระดับบนที่แสดงถึงการปฏิสัมพันธ์กันระหว่างยูสเคสที่ถูกใช้ในการพัฒนาในส่วนของการทำงานเชื่อมประสานในแต่ละฟังก์ชันที่มีผลต่อขนาดของซอฟต์แวร์ เนื่องจากผู้พัฒนาจำเป็นต้องพัฒนาส่วนเชื่อมต่อประสาน (Glue code) จึงทำให้การประมาณขนาดในส่วนนี้ จึงเป็นส่วนหนึ่งในการประมาณการขนาดที่ต้องใช้เพิ่มเติมเมื่อการทำงานในแต่ละฟังก์ชันถูกพัฒนาเสร็จสิ้น เพื่อให้การทำงานต่าง ๆ สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพตามความต้องการที่ได้กำหนดไว้ โดยที่สัญลักษณ์ในส่วนของกิจกรรม (Activity) หรือการกระทำ (Action) ในแผนภาพกิจกรรม จะถูกแทนด้วยยูสเคสที่ถูกระบุภายในแผนภาพยูสเคส ดังนั้นจำนวนของกิจกรรมหรือการกระทำจะเท่ากับจำนวนของยูสเคสในแผนภาพยูสเคสนั่นเอง ซึ่งการประมาณขนาดในมุมมองระดับบนนั้นจะเป็นส่วนที่ผู้พัฒนาซอฟต์แวร์ใช้ในการพัฒนาส่วนเชื่อมต่อประสานกันระหว่างยูสเคสซึ่งเป็นส่วนที่ถูกนำมาใช้พิจารณาประมาณขนาดเพิ่มเติม จึงไม่ได้เป็นการประมาณขนาดของซอฟต์แวร์ที่ซ้ำซ้อนกัน โดยความสัมพันธ์ของยูสเคสไม่ได้แสดงถึงลำดับของการออกแบบแผนภาพกิจกรรม แต่เป็นการแสดงถึง UC2 ต้องมีการปฏิสัมพันธ์กับ UC1



รูปที่ 8 มุมมองแผนภาพกิจกรรม

4.2 การออกแบบแผนภาพกิจกรรมในระดับบนจากแผนภาพยูสเคส

เนื่องจากการออกแบบแผนภาพกิจกรรมในระดับบนนั้นสามารถออกแบบได้หลากหลาย ดังนั้นงานวิจัยนี้จึงได้กำหนดกฎในการสร้างรูปแบบการออกแบบแผนภาพกิจกรรม เพื่อให้สามารถนำมาใช้ในการประมาณของงานวิจัยนี้ โดยมีรายละเอียดของตัวอย่างการแปลงแผนภาพยูสเคสไปยังแผนภาพแอคทิวิตีดังรูปที่ 9 - 10 ซึ่งมีรายละเอียดดังต่อไปนี้

4.2.1 จุดเริ่มต้นของแผนภาพกิจกรรม

จุดเริ่มต้นของแผนภาพกิจกรรมจะถูกชี้ไปยังสัญลักษณ์ Folk เพื่อทำการแยกเส้นทางของยูสเคสที่ไม่ได้มีการปฏิสัมพันธ์ออกจากกัน เพื่อเข้าสู่การออกแบบเส้นทางต่าง ๆ ที่มียูสเคสที่ต้องปฏิสัมพันธ์ ก็จะอยู่ภายใน Folk ที่เชื่อมมาจากจุดเริ่มต้น

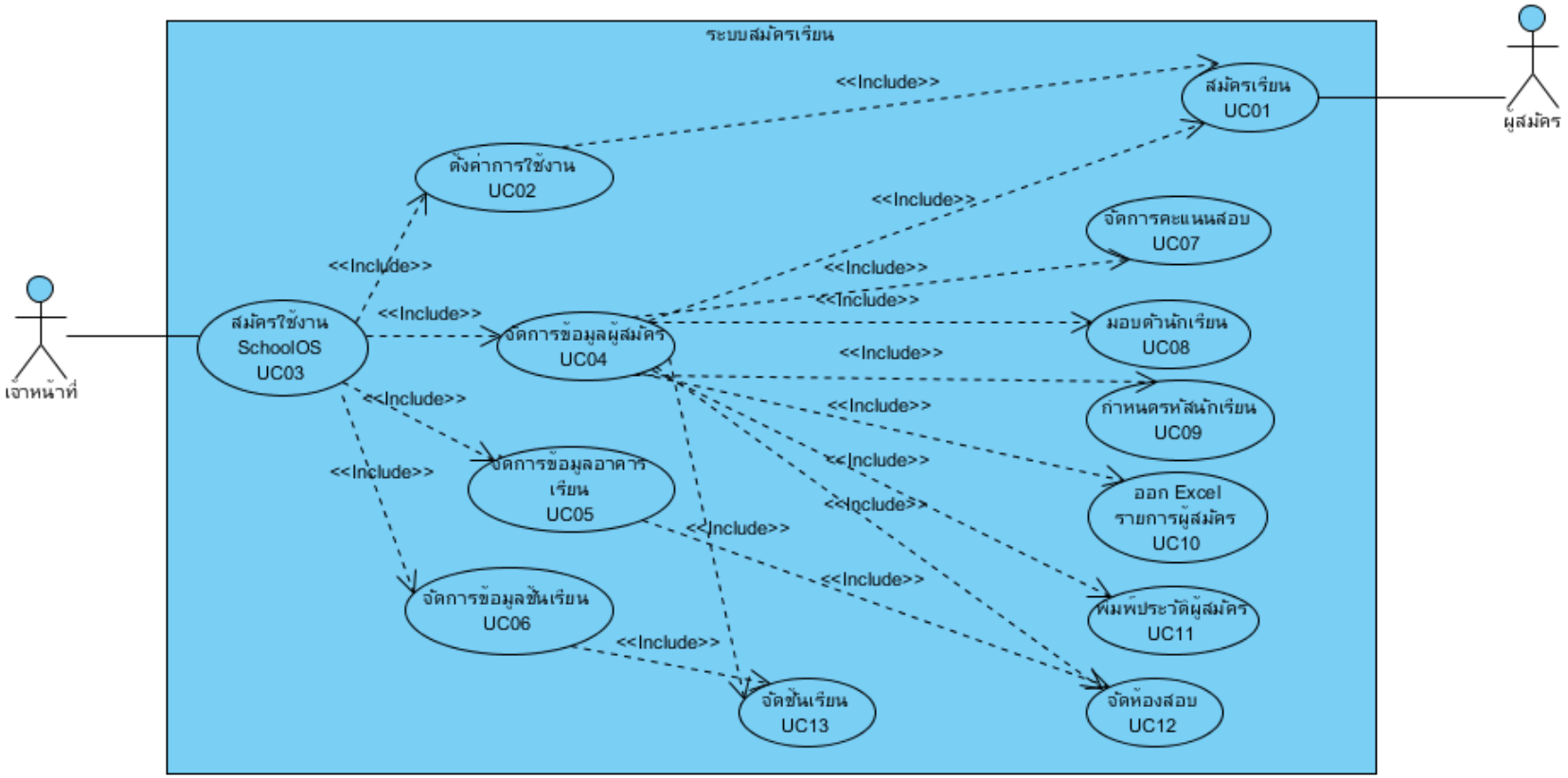
4.2.2 กรณียูสเคสไม่ได้มีการปฏิสัมพันธ์

ยูสเคสที่ไม่ได้มีการปฏิสัมพันธ์จะถูกแยกออกเป็นคนละเส้นทางที่แตกต่างกัน เพื่อเสนอถึงเส้นทางเหล่านั้น ไม่ได้มีการเชื่อมต่อกันหรือไม่มีมีการพัฒนาส่วนเชื่อมต่อประสาน

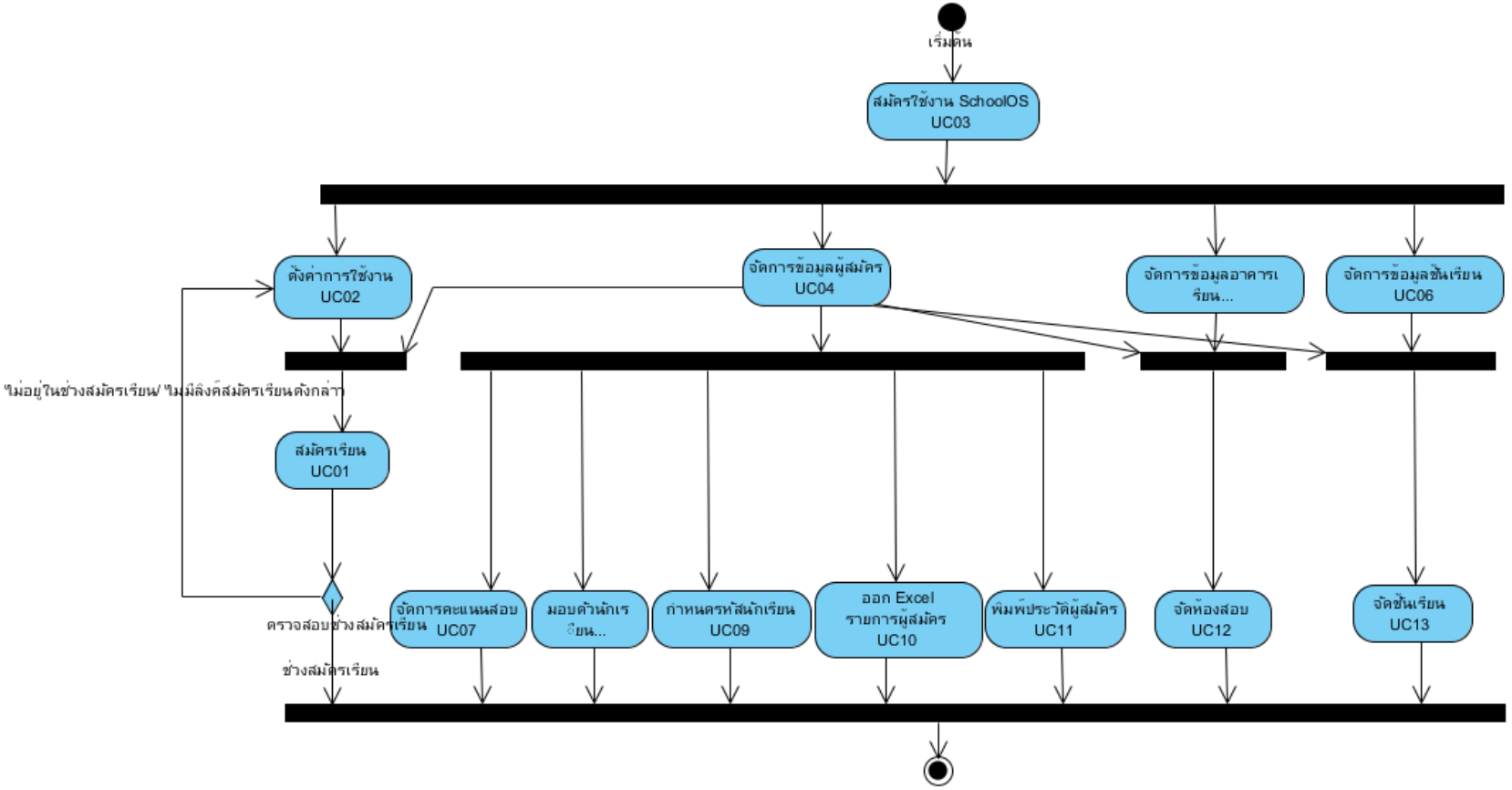
4.2.3 กรณียูสเคสมีการปฏิสัมพันธ์

ยูสเคสที่มีการปฏิสัมพันธ์กันนั้น จะต้องอยู่ในรูปแบบของความสัมพันธ์ส่วนขยาย (Extends) หรือความสัมพันธ์ส่วนประกอบ (Include) โดยความสัมพันธ์ข้างต้นนี้ แสดงถึงยูสเคสที่เป็นส่วนขยายหรือส่วนประกอบ จะต้องมีลำดับที่ต่อมาจากยูสเคสหลัก โดยยูสเคสที่มีความสัมพันธ์แบบส่วนขยายและส่วนประกอบภายในนั้น สามารถที่จะเชื่อมระหว่างกันได้ ซึ่งอาจจะอยู่ในรูปแบบของเงื่อนไขต่าง ๆ หรือการวนซ้ำที่เชื่อมระหว่างกัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 9 ตัวอย่างแผนภาพยูสเคสระบบสมัครเรียน



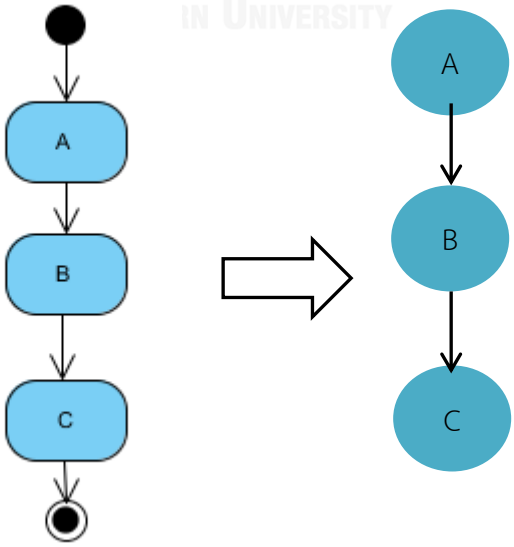
รูปที่ 10 ตัวอย่างแผนภาพกิจกรรมระดับบนของระบบสมัครเรียน

4.3 การแปลงแผนภาพกิจกรรมเป็นกราฟควบคุมกระแส

การแปลงแผนภาพกิจกรรมไปสู่กราฟควบคุมกระแส (Control Flow Graph) นั้น จะถูกใช้ในการหาขนาดในมุมมองระดับบน เพื่อหาความซับซ้อนของการปฏิสัมพันธ์ระหว่างยูสเคส ซึ่งหากแผนภาพกิจกรรมประกอบด้วย สัญลักษณ์ Fork และ Join รูปแบบในการแปลงแผนภาพกิจกรรมนั้นจะถูกแยกออกจากกัน เพื่อทำการคำนวณหาความซับซ้อนในแต่ละเส้นทาง และนำความซับซ้อนของกราฟควบคุมกระแสที่คำนวณได้ มาทำการรวมกันเพื่อแสดงถึงความซับซ้อนของแผนภาพกิจกรรมในมุมมองระดับบนนั้น ๆ ไปสู่การจัดระดับความซับซ้อนของการปฏิสัมพันธ์กันระหว่างยูสเคสต่อไป โดยตัวอย่างของการแปลงแผนภาพกิจกรรมไปสู่กราฟควบคู่กระแสนั้น เป็นการสื่อถึงมุมมองของผลกระทบต่อขนาดของซอฟต์แวร์เพียงเท่านั้น ซึ่งไม่ได้ใช้ในการอธิบายในบริบทของจำนวนเส้นทางการทำงานหลักของฟังก์ชันเหล่านั้น โดยรูปแบบของการแปลงเป็นกราฟควบคุมกระแสสามารถอธิบายมุมมองได้ดังต่อไปนี้

4.3.1 เส้นทางเชิงลำดับของแผนภาพกิจกรรม

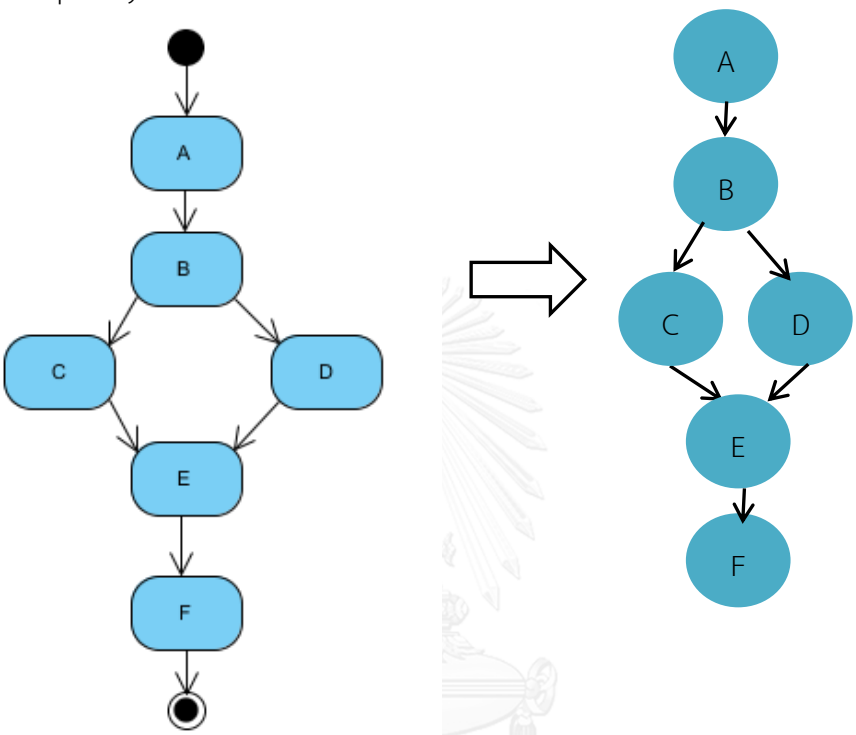
เส้นทางของแผนภาพกิจกรรมที่อยู่ในลักษณะของเส้นทางเส้นตรงเชิงลำดับ การแปลงเป็นกราฟควบคุมกระแสจะยังคงเป็นเส้นทางที่คล้ายกับแผนภาพกิจกรรมเดิม โดยมีจุดเริ่มต้นของแผนภาพกระแสคือ กิจกรรมหรือเหตุการณ์แรกของแผนภาพกิจกรรม และจุดสิ้นสุดคือกิจกรรมหรือเหตุการณ์สุดท้ายของแผนภาพกิจกรรมนั่นเอง ดังแสดงในรูปที่ 11 โดยมีค่าความซับซ้อนของกิจกรรมเท่ากับ 1 (2 – 3 + 2) ที่ได้จากการหาด้วยวิธี McCabe Complexity โดยใช้จำนวนเอดจ์ – จำนวนโหนด + 2



รูปที่ 11 เส้นทางเชิงลำดับของแผนภาพกิจกรรม

4.3.2 เส้นทางของการตัดสินใจ

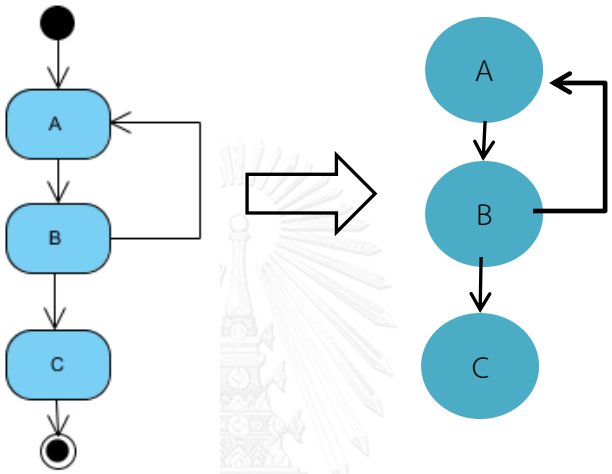
กรณีเส้นทางของแผนภาพกิจกรรมประกอบด้วยการตัดสินใจ การแปลงเป็นแผนกราฟควบคุมกระแสจะยังคงเส้นทางไม่ต่างกันกับแผนภาพกิจกรรม ดังอธิบายในรูปที่ 12 โดยมีจุดเริ่มต้นที่โหนด A และจบที่โหนด H โดยค่าความซับซ้อนของกิจกรรมมีค่าเท่ากับ 2 (6 – 6 + 2) ที่ได้จากการหาด้วยวิธี McCabe Complexity โดยใช้ จำนวนเอดจ์ – จำนวนโหนด + 2



รูปที่ 12 เส้นทางของแผนภาพกิจกรรมในรูปแบบเส้นทางตัดสินใจ

4.3.3 เส้นทางของการวนซ้ำ

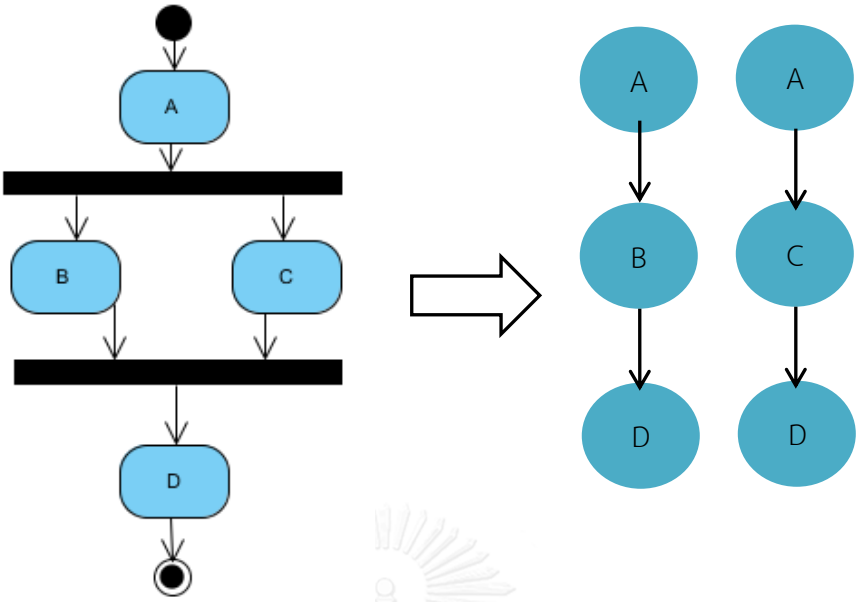
กรณีเส้นทางของแผนภาพกิจกรรมประกอบด้วยเส้นทางวนซ้ำ การแปลงเป็นแผนกราฟควบคุมกระแสจะยังคงเส้นทางไม่ต่างกันกับแผนภาพกิจกรรมเดิมเช่นกันตัวอย่างก่อนหน้านี้ ดังอธิบายในรูปที่ 13 โดยมีจุดเริ่มต้นที่โหนด A และจบที่โหนด C โดยค่าความซับซ้อนของกิจกรรมมีค่าเท่ากับ 2 (3 – 3 + 2) ที่ได้จากการหาด้วยวิธี McCabe Complexity โดยใช้ จำนวนเอดจ์ – จำนวนโหนด + 2



รูปที่ 13 เส้นทางของแผนภาพกิจกรรมที่ประกอบด้วยเส้นทางวนซ้ำ

4.3.4 เส้นทางของ Fork/Join เป็นแผนภาพเชิงลำดับ

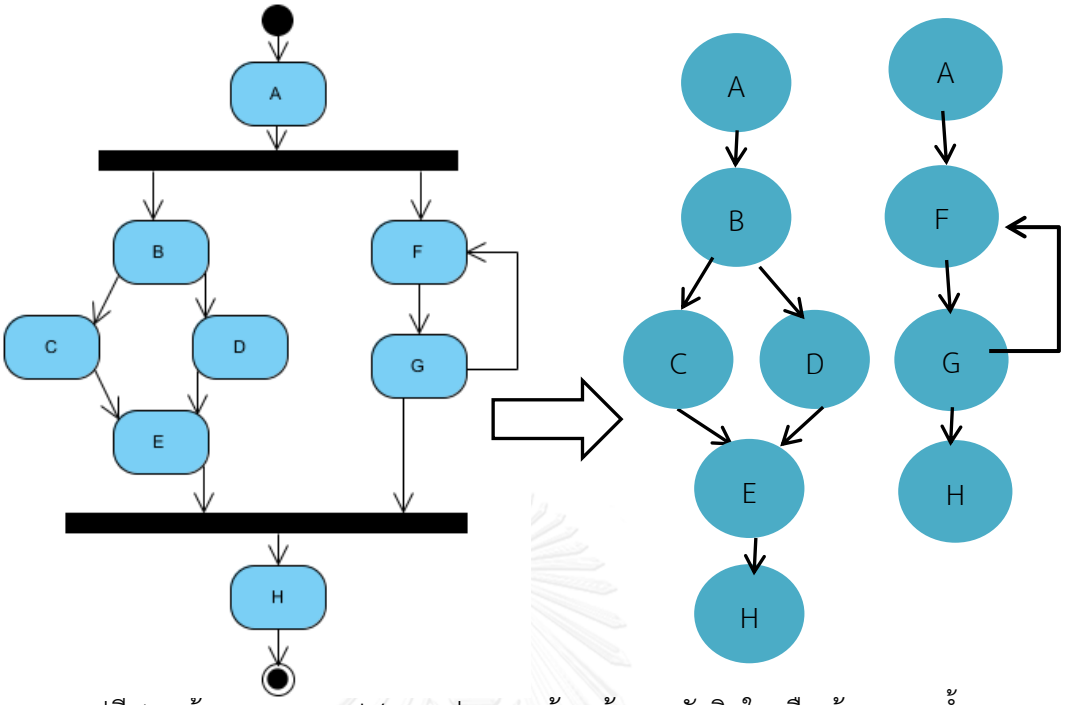
เส้นทางของแผนภาพเหล่านั้นจะถูกแยกออกเป็นสองกราฟควบคุมกระแสตามจำนวนของเส้นทางคู่ขนานที่ปรากฏในแผนภาพกิจกรรม ดังแสดงในรูปที่ 14 เนื่องจากแต่ละเส้นทางถึงแม้จะทำงานคู่ขนาน แต่ความซับซ้อนของกราฟควบคุมกระแสที่ได้นั้นแตกต่างกัน จึงทำให้ต้องมีการคำนวณความซับซ้อนของกราฟควบคุมกระแสออกจากกัน และนำความซับซ้อนที่คำนวณได้จากวิธี McCabe Complexity มารวมกัน เพื่อแสดงถึงความซับซ้อนของแผนภาพกิจกรรมในมุมมองระดับบน ไปสู่การจำแนกระดับความซับซ้อนต่อไป ซึ่งจากรูปที่ 14 แสดงถึงความซับซ้อนของแผนภาพกิจกรรมที่ได้ มีค่าเท่ากับ 2 ที่ได้จากผลรวมของความซับซ้อนของกราฟควบคุมกระแสที่ 1 ที่มีค่าเท่ากับ 1 (2 – 3 + 2) และความซับซ้อนของกราฟควบคุมกระที่ 2 เท่ากับ 1 (2 – 3 + 2) โดยกราฟควบคุมกระแสทั้งสองนั้นเทียบเท่ากันในบริบทของความซับซ้อนของการปฏิสัมพันธ์กันระหว่างยูสเคสในรูปแบบของการทำงานคู่ขนานอิสระออกจากกัน โดยใช้ค่าของ McCabe Complexity ในการเปรียบเทียบระดับความซับซ้อนของเส้นทางทั้ง 2 ดังกล่าว



รูปที่ 14 เส้นทางของ Fork/Join เป็นแผนภาพเชิงลำดับ

4.3.5 เส้นทางของ Fork/Join ประกอบด้วยเส้นทางการตัดสินใจ (Decision) หรือเส้นทางวนซ้ำ (Looping) ภายใน

ในแต่ละเส้นทางที่แตกต่างกันนั้น เส้นทางเหล่านั้นจะยังคงใช้วิธีเดียวกับวิธีที่กล่าวมาข้างต้น โดยความซับซ้อนที่ได้ จะได้มาจากผลรวมของความซับซ้อนที่ได้จากการคำนวณความซับซ้อนด้วยวิธี McCabe Complexity ซึ่งรูปที่ 15 แสดงถึงการแปลงแผนภาพกิจกรรมไปสู่กราฟควบคุมกระแส โดยมีค่าความซับซ้อนของแผนภาพกิจกรรมเท่ากับ 4 ที่ได้จากผลรวมของความซับซ้อนของกราฟกระแสที่ 1 ที่มีค่าเท่ากับ 2 และความซับซ้อนของกราฟกระแสที่ 2 ที่มีค่าเท่ากับ 2 รวมกัน

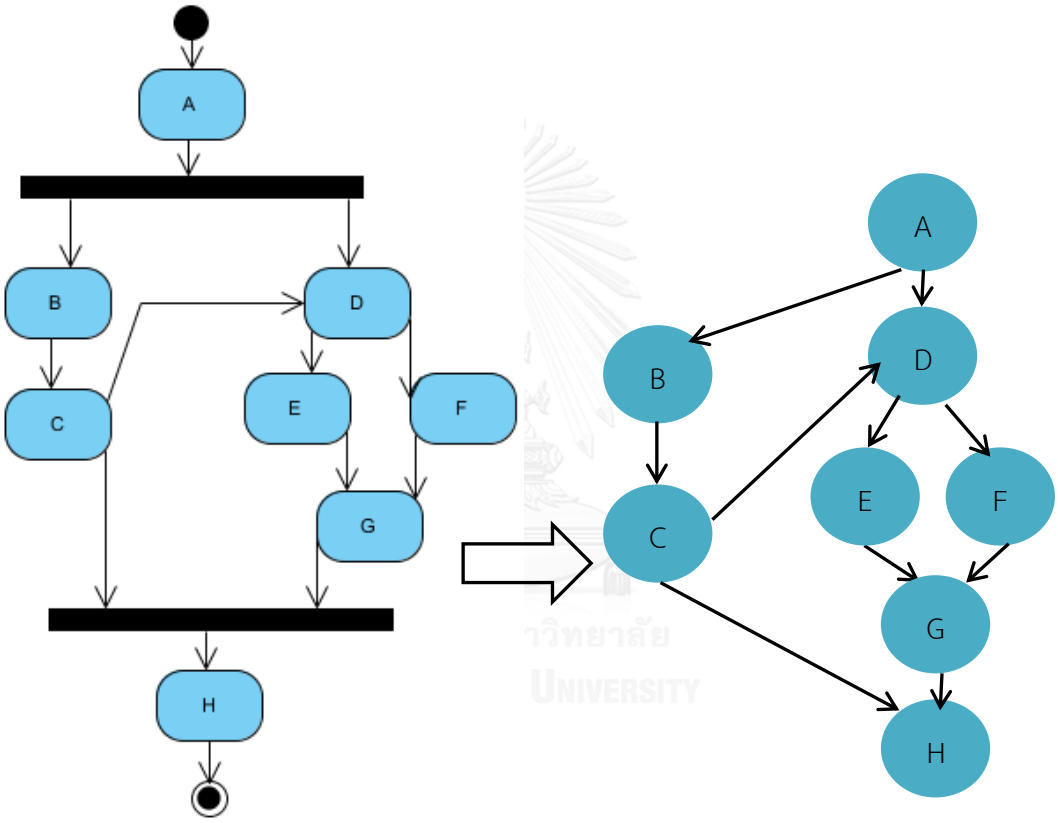


รูปที่ 15 เส้นทางของ Fork/Join ประกอบด้วยเส้นทางตัดสินใจหรือเส้นทางวนซ้ำ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.3.6 เส้นทางของ Fork/Join เป็นเส้นทางมีการเชื่อมประสานกับเส้นทางอื่น ๆ (Synchronization link)

เส้นทางที่มีการเชื่อมกับเส้นทางอื่น ๆ ดังกล่าวจะถูกนำมารวมเป็นกราฟกระแสเส้นทางเดียว เนื่องจากถึงแม้จะมีบางส่วนที่ทำงานคู่ขนานกัน แต่มีการทำงานที่ต้องมีการเชื่อมกับเส้นทางคู่ขนานจึงถือว่าเส้นทางคู่ขนานนั้นเป็นเส้นทางที่ต้องคงไว้ในเส้นทางหลักเดียวกัน ดังแสดงในรูปที่ 16 โดยมีความซับซ้อนของแผนภาพกิจกรรมตามกราฟควบคุมกระแสที่คำนวณได้จากวิธี McCabe Complexity ซึ่งมีค่าเท่ากับ 4

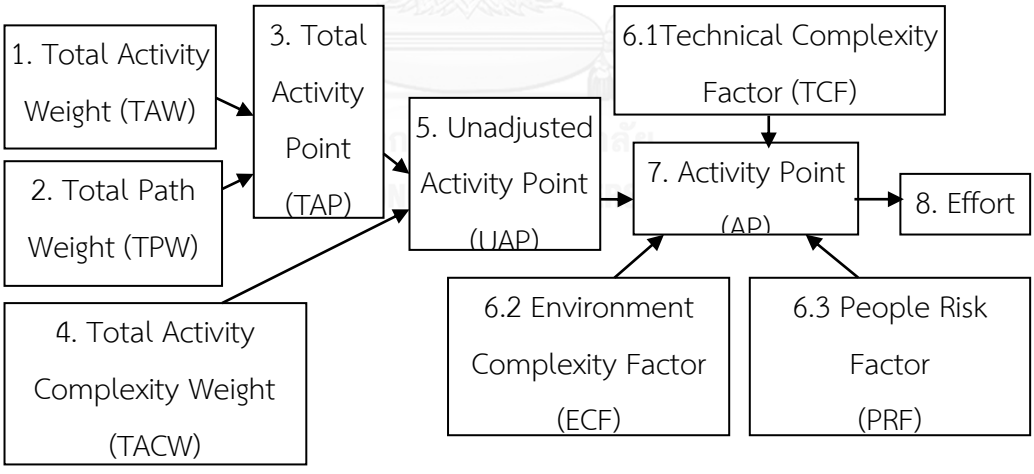


รูปที่ 16 เส้นทางของ Fork/Join ประกอบด้วยการเชื่อมประสานกับเส้นทางอื่น ๆ

4.4 ภาพรวมของวิธีการประมาณขนาดและความพยายามด้วยแอคทิวิตีพอยท์

การประมาณขนาดและความพยายามด้วยวิธีแอคทิวิตีพอยท์ เริ่มต้นจากการนำแผนภาพยูสเคสมาออกแบบแผนภาพกิจกรรมโดยแยกออกเป็น 2 กลุ่มที่ประกอบด้วย แผนภาพกิจกรรมในมุมมองระดับล่าง และ แผนภาพกิจกรรมในมุมมองระดับบน โดยแผนภาพกิจกรรมในมุมมองระดับบนนั้นจะถูกนำมาแปลงเป็นกราฟควบคุมกระแส เพื่อนำแผนภาพกิจกรรมที่ได้ทั้งหมดและแผนภาพควบคุมกระแสมาใช้ในการประมาณขนาดของซอฟต์แวร์ด้วยวิธีแอคทีวิตีพอยท์ต่อไป

จากรูปที่ 17 แสดงถึงภาพรวมของการประมาณขนาดและความพยายามด้วยวิธีแอคทิวิตีพอยท์ ซึ่งเริ่มด้วยการประมาณขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (5.Unadjusted Activity Point: UAP) ในการแทนขนาดของการพัฒนาซอฟต์แวร์ก่อนที่ได้จะปรับค่าด้วยปัจจัยภายนอก ซึ่งได้จากผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม (4.Total Activity Complexity Weight: TACW) ที่ถูกแทนที่การประมาณขนาดจากมุมมองระดับบนโดยใช้วิธี McCabe Complexity และผลรวมของแอคทิวิตีพอยท์ (3.Total Activity Point: TAP) ซึ่งแทนที่ขนาดในมุมมองระดับล่าง เพื่อนำมาปรับค่าด้วยปัจจัยการปรับค่า ซึ่งประกอบด้วย ปัจจัยความซับซ้อนทางเทคนิค (6.1.Technical Complexity Factors: TCF) ปัจจัยความซับซ้อนทางสภาพแวดล้อม (6.2.Environment Complexity Factors: ECF) และปัจจัยทางด้านความเสี่ยงของคน (6.3.People Risk Factors: PRF) ดังที่ได้เสนอในงานวิจัยนี้ โดยรายละเอียดในการประมาณจะถูกอธิบายในหัวข้อถัดไป



รูปที่ 17 ภาพรวมของวิธีการประมาณขนาดและความพยายามด้วยแอคทิวิตีพอยท์

โดยในขั้นตอนของการประมาณผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรมนั้น ได้มาจากการนำแผนภาพกิจกรรมมาแปลงเป็นกราฟควบคุมกระแสดังอธิบายในหัวข้อ การแปลงแผนภาพกิจกรรมเป็นกราฟควบคุมกระแส ซึ่งถูกอธิบายไว้ก่อนหน้านี้

4.5 วิธีการประมาณด้วยแอคทิวิตีพอยท์

4.5.1 ผลรวมของแอคทิวิตีพอยท์ (Total Activity Point)

ผลรวมของแอคทิวิตีพอยท์ เป็นขั้นตอนที่แสดงถึงขนาดในการพัฒนาซอฟต์แวร์ในมุมมองระดับล่าง ซึ่งประกอบด้วยปัจจัยที่ใช้ในการพิจารณา 2 องค์ประกอบหลักคือ ผลรวมของค่าน้ำหนักกิจกรรม (Total Activity Weight: TAW) และผลรวมของค่าน้ำหนักเส้นทาง (Total Path Weight: TPW)

ดังนั้น ผลรวมของแอคทิวิตีพอยท์จึงคำนวณได้จาก ผลรวมของค่าน้ำหนักกิจกรรม และผลรวมของค่าน้ำหนักเส้นทาง ดังแสดงในสมการที่ 1 โดยที่ TAP แทนถึงผลรวมของแอคทิวิตีพอยท์ TAW แทนถึงผลรวมของค่าน้ำหนักกิจกรรม และ TPW แทนถึงผลรวมของน้ำหนักเส้นทาง

$$TAP = TAW + TPW \quad (1)$$

ผลรวมของค่าน้ำหนักกิจกรรม (TAW) เป็นค่าที่แสดงถึงค่าน้ำหนักของมุมมองระดับล่าง ที่เป็นส่วนที่คำนวณจากระดับของสัญลักษณ์กิจกรรมหรือการกระทำ โดยแสดงถึงขั้นตอนลำดับการทำงานภายในของแต่ละยูสเคสที่ถูกระบุไว้ มาออกแบบและแสดงในรูปแบบของแผนภาพกิจกรรมดังแสดงในรูปที่ 8.ก และ 8.ข ซึ่งจำนวนของกิจกรรมหรือการกระทำนั้นจะถูกจำแนกระดับจากจำนวนของกิจกรรมหรือการกระทำ (Number of Activities/Actions Level: NA level) เพื่อใช้แทนในส่วนของลำดับขั้นตอนการทำงานของแต่ละยูสเคสหรือเรียกว่าทรานแซ็คชัน ที่แสดงอยู่ในรูปแบบของแผนภาพกิจกรรมที่ต้องใช้ในการพัฒนาในมุมมองระดับล่าง ในการนำมาถ่วงน้ำหนักที่จำแนกด้วยระดับของจำนวนของกิจกรรมหรือการกระทำ โดยรูปแบบของการคำนวณ ถูกแสดงวิธีการคำนวณด้วยสมการที่ 2 ข้างล่างนี้

$$TAW = \sum_{i=1}^{N} (NA\ level * Weight)_i \quad (2)$$

ผลรวมของค่าน้ำหนักเส้นทาง (TPW) เป็นค่าที่แสดงถึงค่าน้ำหนักของมุมมองระดับล่าง ที่เป็นส่วนที่คำนวณจากระดับของเส้นทางที่เป็นไปได้ของแผนภาพกิจกรรม ที่แสดงถึงจำนวนเส้นทางที่ถูกระบุภายในยูสเคสหรือเรียกว่าซีเนริโอ ที่นำมาออกแบบและแสดงในรูปแบบของแผนภาพกิจกรรมดังแสดงในรูปที่ 8.ก และ 8.ข ซึ่งจำนวนของเส้นทางนั้นจะถูกจำแนกระดับจากจำนวนของเส้นทาง (Number of Paths level: NP level) เพื่อใช้แทนในส่วนของระดับความซับซ้อนของจำนวนซีเนริโอภายในรายละเอียดยูสเคส ที่แสดงอยู่ในรูปแบบของแผนภาพกิจกรรมที่ต้องใช้ในการพัฒนาในมุมมองระดับล่าง ซึ่งจำนวนของเส้นทาง ระบุถึงความซับซ้อนของการพัฒนาในมุมมองระดับล่าง ในการนำมาถ่วงน้ำหนักที่จำแนกด้วยระดับของจำนวนของเส้นทาง ที่หากมีจำนวนเส้นทางที่สูง ความซับซ้อนในการพัฒนาก็จะสูงขึ้นตาม โดยรูปแบบการคำนวณ ถูกแสดงวิธีการคำนวณด้วยสมการที่ 3 ดังต่อไปนี้

$$TPW = \sum_{i=1}^{N}(NP\ level * Weight)_i \qquad (3)$$

ระดับของสัญลักษณ์กิจกรรมหรือการกระทำที่ถูกจำแนกจากจำนวนของกิจกรรมหรือการกระทำ และระดับของเส้นทางที่เป็นไปได้ของแผนภาพกิจกรรมที่ถูกจำแนกจากจำนวนของเส้นทางนั้น สามารถแบ่งได้ออกเป็น 3 ระดับคือ เรียบง่าย (Simple) ปานกลาง (Average) และซับซ้อน (Complex) โดยเกณฑ์ในการแบ่งระดับของสัญลักษณ์กิจกรรมหรือการกระทำและระดับของเส้นทางที่เป็นไปได้ ถูกอธิบายในตารางที่ 5 และ 6 ตามลำดับ

ตารางที่ 5 เกณฑ์การแบ่งระดับสัญลักษณ์กิจกรรมหรือการกระทำ

ระดับ NA	**รายละเอียด**	**น้ำหนัก**
เรียบง่าย	จำนวนขั้นตอนของสัญลักษณ์กิจกรรมหรือการกระทำ ในช่วง 1 ถึง 3 ขั้นตอน	a
ปานกลาง	จำนวนขั้นตอนของสัญลักษณ์กิจกรรมหรือการกระทำ ในช่วง 4 ถึง 7 ขั้นตอน	b
ซับซ้อน	จำนวนขั้นตอนของสัญลักษณ์กิจกรรมหรือการกระทำมากกว่า 7 ขั้นตอนขึ้นไป	c

ตารางที่ 6 เกณฑ์การแบ่งระดับของเส้นทางที่เป็นไปได้

ระดับ NP	รายละเอียด	น้ำหนัก
เรียบง่าย	จำนวนเส้นทางของแผนภาพกิจกรรมในช่วง 1 ถึง 3 เส้นทาง	d
ปานกลาง	จำนวนเส้นทางของแผนภาพกิจกรรมในช่วง 4 ถึง 7 เส้นทาง	e
ซับซ้อน	จำนวนเส้นทางของแผนภาพกิจกรรมมากกว่า 7 เส้นทางขึ้นไป	f

4.5.2 ผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม (Total Activity Complexity Weight)

ผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม เป็นขั้นตอนที่แสดงถึงขนาดในการพัฒนาซอฟต์แวร์ในมุมมองระดับบนของแผนภาพกิจกรรม ซึ่งประกอบด้วยองค์ประกอบหลักคือ ระดับค่าน้ำหนักความซับซ้อนของกิจกรรม (Total Activity Complexity Weight: TACW) ดังแสดงในรูปที่ 8.ค

ดังนั้น ผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม (TACW) จึงคำนวณจาก ระดับของความซับซ้อนของกิจกรรม (Activity Complexity Level: Ac level) ที่ใช้แทนในส่วนของขนาดที่เพิ่มขึ้นจากการพัฒนาในส่วนของการปฏิสัมพันธ์ระหว่างยูสเคส เพื่อนำมาถ่วงน้ำหนักที่จำแนกด้วยระดับของความซับซ้อนของกิจกรรม แสดงถึงหากความซับซ้อนในการปฏิสัมพันธ์ที่สูงขึ้นขนาดของซอฟต์แวร์ในการพัฒนาก็จะสูงขึ้นตามไปด้วย โดยรูปแบบการคำนวณ ถูกแสดงวิธีการคำนวณด้วยสมการที่ 4 ดังต่อไปนี้

$$TACW = AC\ level * Weight \quad (4)$$

การคำนวณความซับซ้อนนั้นของแผนภาพกิจกรรมนั้น แผนภาพกิจกรรมจะถูกแปลงเป็น Control Flow Graph เพื่อใช้ในการคำนวณหาความซับซ้อนของแผนภาพ ด้วยวิธี McCabe Complexity ซึ่งค่าของความซับซ้อนที่ได้มานั้นจะถูกจำแนกระดับออกเป็น 3 ระดับคือ เรียบง่าย ปานกลาง และ ซับซ้อนสูง ซึ่งแทนด้วย ระดับของความซับซ้อนของกิจกรรม โดยใช้เกณฑ์ในการแบ่งระดับดังที่ถูกอธิบายในตารางที่ 7

ตารางที่ 7 เกณฑ์การแบ่งระดับระดับของความซับซ้อนของกิจกรรม

ระดับ AC	รายละเอียด	น้ำหนัก
เรียบง่าย	ความซับซ้อน McCabe Complexity มีค่าในช่วง 1 ถึง 5	g
ปานกลาง	ความซับซ้อน McCabe Complexity มีค่าในช่วง 6 ถึง 10	h
ซับซ้อนสูง	ความซับซ้อน McCabe Complexity มีค่ามากกว่า 10 ขึ้นไป	i

4.5.3 ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (Unadjusted Activity Point)

ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง เป็นขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์ก่อนที่จะมีการปรับปรุงด้วยปัจจัยภายนอกต่าง ๆ ที่จะส่งผลกระทบต่อขนาดของซอฟต์แวร์ โดยขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง สามารถคำนวณได้จากผลรวมของขนาดของมุมมองระดับล่าง ในส่วนของผลรวมของแอคทิวิตีพอยท์ (TAP) และมุมมองระดับบนที่แทนด้วยค่าของผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม (TACW) ดังแสดงวิธีการคำนวณจากสมการที่ 5 โดยที่ N แทนด้วยจำนวนของยูสเคส ที่คำนวณขนาดในมุมมองระดับล่าง

$$UAP = TAP + TACW \tag{5}$$

4.5.4 แอคทิวิตีพอยท์ (Activity Point)

แอคทิวิตีพอยท์ (Activity Point: AP) คือขนาดของซอฟต์แวร์ที่ได้มีการปรับปรุงค่าด้วยปัจจัยปรับปรุงค่า โดยแอคทิวิตีพอยท์นี้สามารถคำนวณหาได้จาก ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงคูณด้วยปัจจัยที่ใช้ในการปรับปรุงค่า ซึ่งประกอบด้วย ปัจจัยความซับซ้อนทางเทคนิค (Technical Complexity Factors: TCF) ปัจจัยความซับซ้อนทางสภาพแวดล้อม (Environment Complexity Factors: ECF) และปัจจัยทางด้านความเสี่ยงของคน (People Risk Factors: PRF) [23] ดังแสดงวิธีการคำนวณจากสมการที่ 6 ดังนี้

$$AP = UAP * TCF * ECF * PRF \tag{6}$$

ปัจจัยภายในที่ใช้ในการปรับค่านั้นจะถูกแสดงภายในตารางที่ 8 – 10 ซึ่งปัจจัยความซับซ้อนทางเทคนิคและปัจจัยความซับซ้อนทางสภาพแวดล้อม ยังคงเหมือนกับปัจจัยที่ได้เสนอในวิธีการประมาณขนาดด้วยยูสเคสพอยท์ แต่ในงานวิจัยนี้ได้นำเสนอปัจจัยใหม่ขึ้น ในด้านปัจจัยทางด้านความเสี่ยงของคน ซึ่งจะส่งผลต่อความพยายามที่ต้องใช้สูงขึ้น

ตารางที่ 8 ปัจจัยความซับซ้อนทางเทคนิค [9]

ปัจจัย	รายละเอียด	น้ำหนัก
T1	ระบบแบบกระจาย	2
T2	เวลาในการตอบกลับ/วัตถุประสงค์ด้านประสิทธิภาพ	2
T3	ประสิทธิภาพของผู้ใช้งานระบบ	1
T4	ความซับซ้อนของกระบวนการ	1
T5	รองรับการใช้งานใหม่ของรหัสข้อมูล	1
T6	ความง่ายในการติดตั้ง	0.5
T7	ความง่ายในการใช้งาน	0.5
T8	สามารถใช้ใน Platform ไหนก็ได้	2
T9	ง่ายในการแก้ไข	1
T10	ประมวลผลพร้อมกัน ๆ ในระดับคู่ขนาน	1
T11	มุ่งเน้นต่อความปลอดภัยเป็นหลัก	1
T12	มีการเชื่อมต่อไปยังระบบที่สาม	1
T13	มีการอบรมผู้ใช้งาน	1

ตารางที่ 9 ปัจจัยความซับซ้อนทางสภาพแวดล้อม [9]

ปัจจัย	รายละเอียด	น้ำหนัก
E1	ลักษณะกระบวนการพัฒนามีความคล้ายคลึงกับระบบก่อน ๆ	1.5
E2	มีประสบการณ์ในด้านซอฟต์แวร์ที่พัฒนา	0.5
E3	ทีมมีองค์ความรู้ด้านการพัฒนาเชิงวัตถุ	1.0
E4	สมาชิกทีมสามารถวิเคราะห์ระบบได้ด้วยตนเอง	0.5
E5	ทีมมีการให้แรงจูงใจในการทำงาน	1.0
E6	ความต้องการมีความคงที่	2.0
E7	มีพนักงานพาร์ทไทม์	-1.0
E8	ใช้ภาษายากในการพัฒนา	-1.0

ตารางที่ 10 ปัจจัยทางด้านความเสี่ยงของคน

ปัจจัย	รายละเอียด	น้ำหนัก
R1	สมาชิกในทีมผลัดวันประกันพรุ่ง	2
R2	สมาชิกในทีมมาประชุมสาย/ขาดประชุม	0.5
R3	สมาชิกในทีมส่งงานไม่ครบตามกำหนด	2
R4	การวางแผน/กลยุทธ์ ไม่มีประสิทธิภาพ	1
R5	ลูกค้ามีความเชื่อในมุมมองของลูกค้าที่ไม่ตรงกับความเป็นจริง	1.5

การคำนวณหาปัจจัยในการปรับปรุงค่านั้น ประกอบด้วย ปัจจัยทางด้านเทคนิค (Technical Factor: TF) ปัจจัยทางด้านสิ่งแวดล้อม (Environment Factor: EF) และปัจจัยทางด้านคน (People Factor: PF) ซึ่งคะแนนจะถูกระบุคะแนนตั้งแต่ 0 จนถึง 5 ที่แสดงถึงความสัมพันธ์ของปัจจัยเหล่านั้นต่อขนาดของซอฟต์แวร์หรือความพยายามที่ใช้ในการพัฒนาซอต์แวร์ (ระดับ 0 คือลักษณะจำเพาะนี้ไม่มีผลกระทบกับระบบ จนถึงระดับ 5 คือลักษณะจำเพาะนี้มีผลกระทบกับระบบมาก) ซึ่งสมการที่ใช้ในการคำนวณของ ปัจจัยความซับซ้อนทางเทคนิค ปัจจัยความซับซ้อนทางสภาพแวดล้อม และปัจจัยทางด้านความเสี่ยงของคน ดังแสดงวิธีการคำนวณจากสมการที่ 7 – 9 ตามลำดับดังต่อไปนี้

$$TCF = 0.6 + TF/100 \quad (7)$$

$$ECF = 1.4 + (-0.03 * EF) \quad (8)$$

$$PRF = 1 + (0.12 * PF) \quad (9)$$

จากสมการที่ 9 สร้างขึ้นโดยใช้ค่าขนาด AP ที่ได้จากชุดโครงการ 7 โครงการที่ไม่ได้มีการปรับค่าด้วย PRF (ค่า PF มีค่าเท่ากับ 0) จึงทำให้ PRF มีค่าเท่ากับ 1 ซึ่งทำให้การปรับค่ามีลักษณะเหมือนวิธีการของยูสเคสพอยท์ (AP = UAP * TCF * ECF * 1) ซึ่งค่า AP ที่ได้เดิมนั้นยังมีความคลาดเคลื่อนจากค่าความพยายามจริงในการพัฒนาซอฟต์แวร์อยู่ จึงได้เพิ่มการปรับค่าเพิ่มเติมด้วย PRF ด้วยการกำหนดค่าคะแนนและถ่วงน้ำหนักของแต่ละโครงการเพื่อให้ได้ค่า PF ในแต่ละโครงการและทำการคูณด้วยค่าสัมประสิทธิ์ 0.12 ซึ่งเป็นค่าที่ทำให้ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (Average MRE) ในการระบุถึงความคลาดเคลื่อนของแบบจำลองของทั้ง 7 โครงการมีค่าน้อยที่สุด ซึ่งแสดงภายในสมการที่ 10 โดยที่ N แสดงถึงจำนวนของโครงการที่ใช้ในการสร้างแบบจำลอง ซึ่งจากสมการที่ 9 นี้บ่งบอกได้ว่าหากความเสี่ยงสูงสุดในทุกปัจจัยจะส่งผลให้ขนาดของซอฟต์แวร์เพิ่มขึ้นสูงสุด 5.2 เท่า ก็ต่อเมื่อปัจจัยทางด้านคนมีค่าสูงสุดเท่ากับ 35

$$\text{Average MRE} = \frac{\sum_{i=1}^{N}\left(\frac{Actual\ Effort_i - Estimated\ Effort_i}{Estimated\ Effort_i}\right)}{N} \quad (10)$$

4.5.5 ความพยายาม (Effort)

แบบจำลองในการสร้างความพยายาม ถูกสร้างด้วยข้อมูลนำเข้าขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์นั้น จะใช้วิธีการสมการเส้นตรงแบบถดถอย (Linear Regression) โดยมีองค์ประกอบด้วยตัวแปรในการทำนายคือ ความพยายามจริง (Actual Effort: AP) และ ตัวแปรอิสระ (Independence Variable) คือ ขนาดซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์ โดยรูปแบบของแบบจำลองการประมาณความพยายามคำนวณได้จากสมการที่ 11

$$Effort = a + b * AP \quad (11)$$

4.6 การกำหนดค่าน้ำหนักของระดับสัญลักษณ์กิจกรรม ระดับเส้นทางที่เป็นไปได้ และระดับความซับซ้อนของกิจกรรม

ค่าถ่วงน้ำหนักในวิธีแอคทิวิตีพอยท์ ที่แสดงในตารางที่ 14 - 16 นั้น คำนวนด้วยการเริ่มต้นจากการค่าสหสัมพันธ์ของ จำนวนของทรานแซคชันทั้งหมดของทุกโครงการ จำนวนของซีเนริโอทั้งหมดของทุกโครงการ จำนวนของค่าความซับซ้อนของกิจกรรมของทุกโครงการ และ ค่าความพยายามในการพัฒนาซอฟต์แวร์จริงจากทั้ง 10 โครงการ เพื่อหาปัจจัยใดที่มีผลต่อค่าความพยายามในการพัฒนาสูงที่สุดซึ่ง โดยมีผลลัพธ์ ที่คำนวนด้วยโปรแกรม SPSS โดยมีผลลัพธ์ตามลำดับ ดังนี้ ค่า 0.849 สำหรับผลรวมของค่าน้ำหนักเส้นทาง ค่า 0.829 สำหรับค่าผลรวมของค่าน้ำหนักกิจกรรม และ ค่า 0.509 สำหรับค่าผลรวมของความซับซ้อนกิจกรรม ดังแสดงในรูปที่ 18

Correlations

		EFFORT	NS	NT	COMPLEXITY
EFFORT	Pearson Correlation	1	.849**	.829**	.509
	Sig. (2-tailed)		.002	.003	.133
	N	10	10	10	10
NS	Pearson Correlation	.849**	1	.932**	.495
	Sig. (2-tailed)	.002		.000	.146
	N	10	10	10	10
NT	Pearson Correlation	.829**	.932**	1	.613
	Sig. (2-tailed)	.003	.000		.060
	N	10	10	10	10
COMPLEXITY	Pearson Correlation	.509	.495	.613	1
	Sig. (2-tailed)	.133	.146	.060	
	N	10	10	10	10

รูปที่ 18 ค่าสหสัมพันธ์ที่มีผลกับความพยายามจริงในการพัฒนาซอฟต์แวร์

ค่าสหสัมพันธ์ที่ได้จะถูกนำมารวมกัน เพื่อนำมาหาสัดส่วนผลกระทบในแต่ละปัจจัย โดยผลรวมของค่าสหสัมพันธ์ของผลรวมของค่าน้ำหนักเส้นทาง ผลรวมของค่าน้ำหนักกิจกรรม และผลรวมของความซับซ้อนกิจกรรม มีค่าเท่ากับ 2.187 (0.849 + 0.829 + 0.509) ซึ่งถูกนำมาใช้ในการหาสัดส่วนของ 3 ปัจจัยในรูปแบบของเปอร์เซ็น ซึ่งจะได้ค่าเท่ากับ 38.82 : 37.90 : 23.27 เมื่อเทียบกับความพยายามจริงในการพัฒนาซอฟต์แวร์

สัดส่วนที่ได้จะถูกนำมาใช้ในการแบ่งสัดส่วนของค่าความพยายามในการพัฒนาซอฟต์แวร์จริงของชุดโครงการ 7 โครงการที่ใช้ในการสร้างแบบจำลอง ซึ่งสัดส่วนของ TPW มีความสัมพันธ์กับสมการที่ 3 ($TPW = \sum_{i=1}^{N}(NP\ level * Weight)_i$) สัดส่วนของ TAW มีความสัมพันธ์กับสมการที่ 2 ($TAW = \sum_{i=1}^{N}(NA\ level * Weight)_i$) และสัดส่วนของ TACW มีความสัมพันธ์กับสมการที่ 4 ($TACW = AC\ level * Weight$) จะซึ่งสรุปได้ดังตารางที่ 11

ตารางที่ 11 สัดส่วนของปัจจัยที่มีความสัมพันธ์กับความพยายามในการพัฒนาซอฟต์แวร์จริง

โครงการ	**ความพยายามจริงในการพัฒนาซอฟต์แวร์**	**สัดส่วนของ TPW (38.82%)**	**สัดส่วนของ TAW (37.90%)**	**สัดส่วนของ TACW (23.27%)**
1	97	37.64	36.76	22.60
2	28	10.86	10.61	6.52
3	64	24.83	24.26	14.91
4	186	72.17	70.49	43.34
5	109.5	42.49	41.50	25.51
6	306	118.73	115.97	71.30
7	197	76.44	74.66	45.90

จำนวนของระดับของทรานแซคชัน จำนวนของระดับของเส้นทาง และจำนวนของระดับความซับซ้อนของกิจกรรม ของแต่ละโครงการที่แสดงในตารางที่ 12 นั้น จะถูกใช้ในการหาค่าน้ำหนักในแต่ละระดับที่จะส่งผลให้ค่าที่ประมาณได้ใกล้เคียงกับความพยายามในการพัฒนาซอฟต์แวร์จริงสูงที่สุด โดยค่าน้ำหนักที่ได้ในแต่ละโครงการจะถูกนำมาหาค่าเฉลี่ยและทำการปัดเลขขึ้นเป็นจำนวนเต็มบวกเพื่อแทนถึงค่าน้ำหนักมาตรฐานของระดับของปัจจัยนั้น ๆ ในการประมาณขนาดด้วยวิธีแอคทิวิตีพอยท์ต่อไป โดยผลลัพธ์ของค่าน้ำหนักที่ทำให้ค่าใกล้เคียงที่สุดของแต่ละโครงการจะถูกแสดงในตารางที่ 13 โดยใช้ Pseudocode ดังรูปที่ 19 จากการวนซ้ำเพื่อหาค่าน้ำหนักที่ส่งผลค่าให้ไกล้เคียง

สุดกับค่าความพยายามในการพัฒนาซอฟต์แวร์จริงมากที่สุดในแต่ละโครงการ โดยระบุช่วงน้ำหนักที่เป็นไปได้ช่วงค่า 1 จนถึง 20 ในแต่ละระดับของแต่ละปัจจัยที่ถูกแบ่งตามสัดส่วนไว้

ตารางที่ 12 จำนวนระดับของเส้นทาง ทรานแซคชัน และค่าความซับซ้อนของกิจกรรม

โครงการ	จำนวนระดับของเส้นทาง			จำนวนระดับของทรานแซคชัน			จำนวนระดับของค่าความซับซ้อนของกิจกรรม		
	เรียบง่าย	ปานกลาง	ซับซ้อน	เรียบง่าย	ปานกลาง	ซับซ้อน	เรียบง่าย	ปานกลาง	ซับซ้อน
1	3	8	1	0	1	11	0	1	0
2	1	3	1	0	0	5	1	0	0
3	7	0	2	0	5	4	0	1	0
4	9	1	2	2	6	4	0	1	0
5	5	6	0	1	4	6	0	1	0
6	14	16	1	2	9	20	0	0	1
7	18	11	0	5	9	15	0	0	1

ตารางที่ 13 ค่าน้ำหนักที่ทำให้ค่าประมาณมีความใกล้เคียงความพยายามในการพัฒนาซอฟต์แวร์จริงสูงสุด

โครงการ	ค่าน้ำหนักของเส้นทาง			ค่าน้ำหนักของทรานแซคชัน			ค่าน้ำหนักของค่าความซับซ้อนของกิจกรรม		
	เรียบง่าย	ปานกลาง	ซับซ้อน	เรียบง่าย	ปานกลาง	ซับซ้อน	เรียบง่าย	ปานกลาง	ซับซ้อน
1	1	2	18	1	2	3	1	19	20
2	0	0	0	0	0	0	7	19	20
3	2	3	5	1	2	3	14	15	20
4	1	5	14	2	4	5	1	19	20
5	1	6	7	1	2	5	1	19	20
6	1	6	8	2	3	4	1	2	20
7	1	5	6	1	2	3	1	2	20
เฉลี่ย	1.00	3.86	8.29	1.14	2.14	3.29	3.71	13.57	20
มาตรฐาน	1	4	9	2	3	4	4	14	20

ยกตัวอย่างจากตารางที่ 11 – 13 ในส่วนของ สัดส่วนของ TPW ของโครงการที่ 1 มีค่าเท่ากับ 37.64 โดยมีจำนวนของระดับของเส้นทางที่ประกอบด้วย ระดับเรียบง่าย 3 จำนวน ระดับปานกลาง 8 จำนวน และระดับซับซ้อน 1 จำนวน จากยูสเคสทั้งหมด 12 ยูสเคส โดยน้ำหนักที่ทำให้ค่าใกล้เคียงกับ 37.64 สูงที่สุดคือ ค่าน้ำหนักเรียบง่ายที่มีค่า 1 ค่าน้ำหนักปานกลางที่มีค่า 2 และ ค่าน้ำหนักซับซ้อนที่มีค่า 18 โดยให้ค่าที่ประมาณได้เท่ากับ 37 (3*1 + 8*2 + 1*18) ซึ่งใกล้เคียง 37.64 ที่สุด

1	INPUT ข้อมูลโครงการที่ประกอบด้วย ความพยายามจริง จำนวนของกิจกรรมหรือการกระทำในแต่ละระดับ จำนวนของเส้นทางในแต่ละระดับ และค่าความซับซ้อนของกิจกรรม
2	FIND ค่าสหสัมพันธ์ของ ผลรวมของค่าน้ำหนักกิจกรรม (TAW) ผลรวมของค่าน้ำหนักเส้นทาง (TPW) และผลรวมของความซับซ้อนกิจกรรม (TACW) กับ ความพยายามจริง (Actual Effort) SET ค่าสัดส่วนของ TAW ค่าสัดส่วนของ TPW และ ค่าสัดส่วนของ TACW
3	LOOP หาค่าน้ำหนักที่เหมาะสมของกิจกรรมในแต่ละโครงการ โดยใช้สัดส่วนของ TAW ในการเปรียบเทียบ โดยเริ่มจากค่าน้ำหนัก 1 จนถึง 20 เพื่อทำการคูณจำนวนทั้งหมดของแต่ละระดับ ในการส่งผลให้ค่าใกล้เคียงกับสัดส่วน TAW มากที่สุด STORE ค่าน้ำหนักที่เหมาะสมที่ได้ในส่วนของ TAW ของทุกโครงการ
4	LOOP หาค่าน้ำหนักที่เหมาะสมของเส้นทางในแต่ละโครงการ โดยใช้สัดส่วนของ TPW ในการเปรียบเทียบ โดยเริ่มจากค่าน้ำหนัก 1 จนถึง 20 เพื่อทำการคูณจำนวนทั้งหมดของแต่ละระดับ ในการส่งผลให้ค่าใกล้เคียงกับสัดส่วน TPW มากที่สุด STORE ค่าน้ำหนักที่เหมาะสมที่ได้ในส่วนของ TPW ของทุกโครงการ
5	LOOP หาค่าน้ำหนักที่เหมาะสมของค่าความซับซ้อนกิจกรรมในแต่ละโครงการ โดยใช้สัดส่วนของ TACW ในการเปรียบเทียบ โดยเริ่มจากค่าน้ำหนัก 1 จนถึง 20 เพื่อการทำคูณจำนวนทั้งหมดของแต่ละระดับ ในการส่งผลให้ค่าใกล้เคียงกับสัดส่วน TACW มากที่สุด STORE ค่าน้ำหนักที่เหมาะสมที่ได้ในส่วนของ TACW ของทุกโครงการ
6	PRINT ช่วงค่าน้ำหนักที่เหมาะสมของ ค่าผลรวมของค่าน้ำหนักกิจกรรม (TAW) ค่าผลรวมของค่าน้ำหนักกิจกรรม (TPW) ค่าผลรวมของค่าน้ำหนักกิจกรรม (TACW)
7	SET ค่าน้ำหนักมาตรฐานของแต่ละระดับในแต่ละปัจจัย ด้วยค่าเฉลี่ยของค่าน้ำหนักของระดับนั้น ๆ ในทุกโครงการที่นำมาใช้สร้างแบบจำลอง

รูปที่ 19 ลำดับรหัสเทียมการหาค่าน้ำหนักของวิธีแอคทิวิตีพอยท์

ผู้วิจัยได้ทำการพัฒนาระบบคำนวนอัตโนมัติด้วยภาษา Java โดยการนำเข้าชุดข้อมูลด้วยไฟล์ CSV ที่ภายในประกอบด้วย ค่าความพยายามในการพัฒนาซอฟต์แวร์จริง ค่าจำนวนของแต่ละระดับเส้นทาง ค่าจำนวนของแต่ละระดับทรานแซคชัน และค่าจำนวนของแต่ละระดับค่าความซับซ้อนของกิจกรรม โดยรายละเอียดเพิ่มเติมถูกแสดงใน ภาคผนวก ข

ตารางที่ 14 น้ำหนักเกณฑ์การแบ่งระดับสัญลักษณ์กิจกรรมหรือการกระทำ

ระดับ NA	**รายละเอียด**	**น้ำหนัก**
เรียบง่าย	จำนวนขั้นตอนของสัญลักษณ์กิจกรรมหรือการกระทำ ในช่วง 1 ถึง 3 ขั้นตอน	2
ปานกลาง	จำนวนขั้นตอนของสัญลักษณ์กิจกรรมหรือการกระทำ ในช่วง 4 ถึง 7 ขั้นตอน	3
ซับซ้อน	จำนวนขั้นตอนของสัญลักษณ์กิจกรรมหรือการกระทำมากกว่า 7 ขั้นตอนขึ้นไป	4

ตารางที่ 15 น้ำหนักเกณฑ์การแบ่งระดับของเส้นทางที่เป็นไปได้

ระดับ NP	**รายละเอียด**	**น้ำหนัก**
เรียบง่าย	จำนวนเส้นทางของแผนภาพกิจกรรมในช่วง 1 ถึง 3 เส้นทาง	1
ปานกลาง	จำนวนเส้นทางของแผนภาพกิจกรรมในช่วง 4 ถึง 7 เส้นทาง	4
ซับซ้อน	จำนวนเส้นทางของแผนภาพกิจกรรมมากกว่า 7 เส้นทางขึ้นไป	9

Chulalongkorn University

ตารางที่ 16 น้ำหนักเกณฑ์การแบ่งระดับระดับของความซับซ้อนของกิจกรรม

ระดับ AC	**รายละเอียด**	**น้ำหนัก**
เรียบง่าย	ความซับซ้อน McCabe Complexity มีค่าในช่วง 1 ถึง 5	4
ปานกลาง	ความซับซ้อน McCabe Complexity มีค่าในช่วง 6 ถึง 10	14
ซับซ้อนสูง	ความซับซ้อน McCabe Complexity มีค่ามากกว่า 10 ขึ้นไป	20

บทที่ 5
การวิเคราะห์และออกแบบการพัฒนาเครื่องมือวิจัย

ในบทนี้กล่าวถึงการวิเคราะห์ความต้องการ และออกแบบเครื่องมืองานวิจัยซึ่งประกอบด้วย แผนภาพยูสเคส แผนภาพกิจกรรม และแผนภาพคลาส ที่จะถูกนำมาใช้ในการพัฒนาเครื่องมือวิจัยที่ถูกใช้ในงานวิจัยนี้ โดยมีวัตถุประสงค์ให้สามารถวิเคราะห์ผลลัพธ์การประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์

5.1 การวิเคราะห์ความต้องการของเครื่องมืองานวิจัย

การประมาณด้วยวิธีการแอคทิวิตีพอยท์นั้นจำเป็นต้องมีการนับปัจจัยต่าง ๆ ตามรูปแบบการประมาณที่ได้เสนอก่อนหน้านี้ ซึ่งอาจจะทำให้เกิดความไม่สะดวกต่อผู้ที่นำไปใช้ในการวางแผนโครงการ งานวิจัยนี้จึงได้พัฒนาเครื่องมือขึ้นโดยมีรูปแบบฟังก์ชันการทำงานที่รองรับวิธีการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยแอคทิวิตีพอยท์ ซึ่งประกอบด้วย

1. การนำแผนภาพกิจกรรมที่ออกแบบด้วยโปรแกรม Visual Paradigm ในรูปแบบไฟล์ XML มาใช้ในการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ในแต่ละขั้นตอน
2. การจำแนกแผนภาพกิจกรรมทั้งหมดที่ถูกระบุและออกแบบไว้จากข้อมูลนำเข้าของไฟล์ XML ข้างต้น เพื่อใช้ในการระบุระดับของแผนภาพกิจกรรมว่า แผนภาพใดเป็นแผนภาพมุมมองระดับล่าง และแผนภาพใดเป็นแผนภาพมุมมองระดับบน
3. การหาค่าจำนวนของกิจกรรมแต่ละการทำงานของแผนภาพกิจกรรมในมุมมองระดับล่าง
4. การหาค่าจำนวนของเส้นทางในแต่ละการทำงานของแผนภาพกิจกรรมในมุมมองระดับล่าง
5. การคำนวณค่าผลรวมของค่าน้ำหนักกิจกรรม (TAW)
6. การคำนวณค่าผลรวมของค่าน้ำหนักเส้นทาง (TPW)
7. การคำนวณค่าผลรวมของแอคทิวิตีพอยท์ (TAP)
8. การคำนวณค่าผลรวมค่าน้ำหนักความซับซ้อนของกิจกรรม (TACW)
9. การคำนวณค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (UAP)
10. การคำนวณค่าปัจจัยในการปรับค่า ประกอบด้วย ปัจจัยความซับซ้อนทางเทคนิค (TCF) ปัจจัยความซับซ้อนทางสภาพแวดล้อม (ECF) และปัจจัยทางด้านความเสี่ยงของคน (PRF)
11. การคำนวณหาค่าขนาดของซอฟต์แวร์ในหน่วย แอคทิวิตีพอยท์
12. การคำนวณหาค่าความพยายามในการพัฒนาซอฟต์แวร์

5.2 การออกแบบเครื่องมืองานวิจัย

5.2.1 แผนภาพยูสเคสของเครื่องมืองานวิจัย

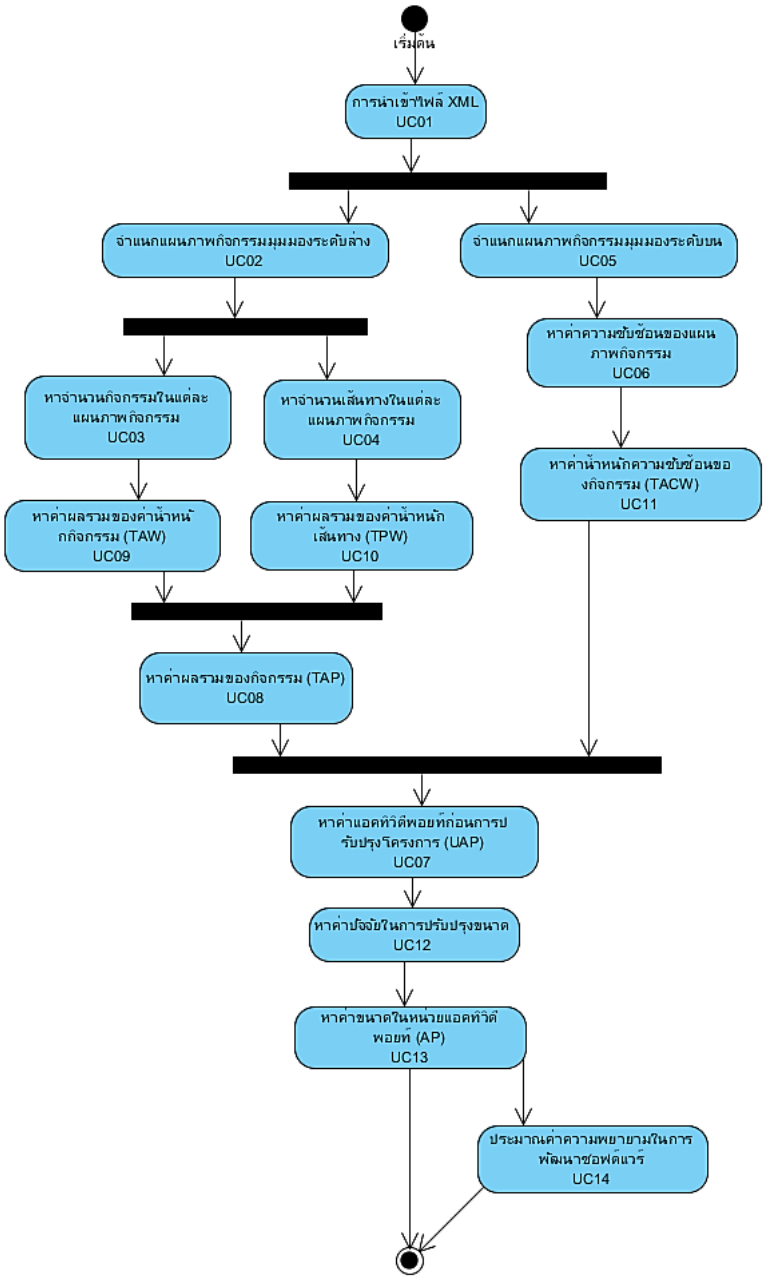
จากรายละเอียดฟังก์ชันการทำงานข้างต้น จะถูกนำมาออกแบบเป็นแผนภาพยูสเคส เพื่อใช้ในการสรุปการทำงานให้อยู่ในรูปของแผนภาพที่เข้าใจง่ายต่อผู้ที่ต้องการต่อยอดหรือนำหลักการของวิธีการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์ โดยมีรายละเอียดดังรูปที่ 20

รูปที่ 20 แผนภาพยูสเคสเครื่องมืองานวิจัย

รายละเอียดของขั้นตอนการทำงานในแต่ละยูสเคสจะถูกอธิบายโดยใช้รายละเอียดของยูสเคสข้างล่างนี้ในการแสดงถึงลำดับขั้นตอน เหตุการณ์ที่เป็นไปได้ และผู้ใช้ที่เกี่ยวข้องกับยูสเคส โดยมีรายละเอียดของยูสเคสดังแสดงในภาคผนวก ก

5.2.2 แผนภาพกิจกรรมของเครื่องมืองานวิจัย

แผนภาพยูสเคสที่ได้ จะถูกนำมาออกแบบกิจกรรมเพื่ออธิบายถึงลำดับขั้นตอนของกิจกรรมที่ใช้ในการประมาณขนาดและความพยายามซอฟต์แวร์ เพื่อสร้างความเข้าใจในขั้นตอนของการประมาณให้ง่ายยิ่งขึ้น โดยถูกอธิบายในรูปที่ 21

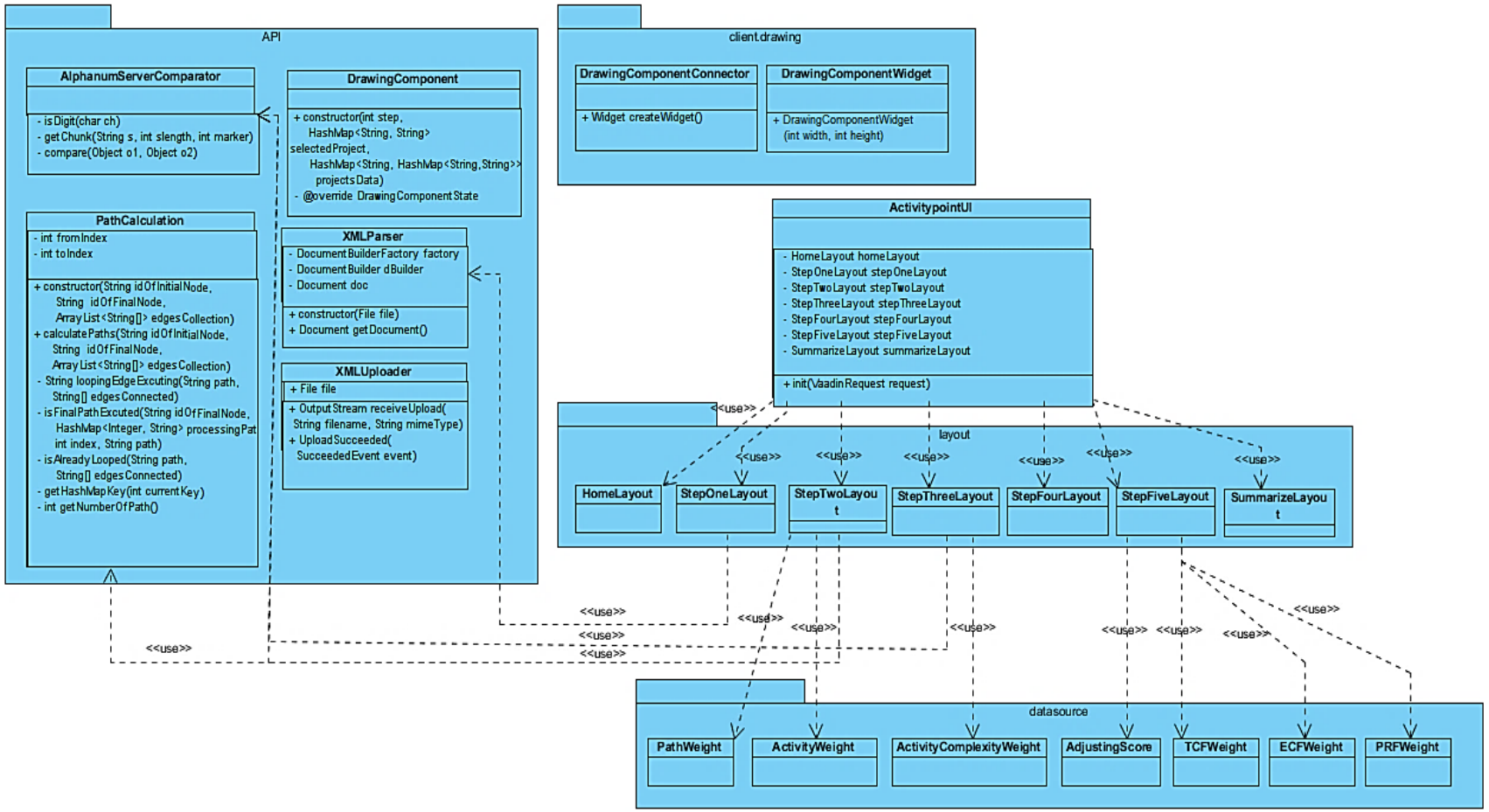


รูปที่ 21 แผนภาพกิจกรรมของเครื่องมืองานวิจัย

5.2.3 แผนภาพคลาสของเครื่องมืองานวิจัย

แผนภาพคลาสของเครื่องมืองานวิจัยที่ได้ จะถูกนำมาออกแบบโครงสร้างของชุดรหัสข้อมูล เพื่อเข้าสู่ขั้นตอนการการพัฒนาเครื่องมืองานวิจัยตามเทคโนโลยีที่ได้กำหนดไว้ โดยในรูปที่ 22 แสดงถึงโครงสร้างของโครงการที่จะถูกพัฒนาโดยประกอบด้วย 5 องค์ประกอบหลัก ๆ คือ

1. **API** เป็นส่วนที่ใช้ในการเก็บคลาสในส่วนของการทำงานกลาง ๆ ที่จะถูกใช้ในฟังก์ชันหลัก ๆ ของโครงการพัฒนา โดยภายในประกอบด้วยไฟล์การทำงานหลักคือ
   - **AlphanumServerComparator** เป็นการทำงานที่ใช้ในการเรียงลำดับของไฟล์ XML ที่ได้เข้ามาให้อยู่ในรูปแบบการเรียงที่มนุษย์เข้าใจง่าย เช่น img1 img011 img2 ให้อยู่ในรูปการเรียงของ img1 img2 img11 เป็นต้น
   - **PathCalculation** เป็นการทำงานที่ใช้ในการคำนวณเส้นทางของแผนภาพกิจกรรมที่ได้จากอัลกอริทึมกลางที่พัฒนา โดยมีหลักการคือให้ระบบทำการร่างแผนภาพขึ้นมาใหม่ตั้งแต่ จุดเริ่มต้น ไปจนถึงจุดสิ้นสุด โดยระหว่างของการร่างในแต่ละองค์ประกอบระบบจะทำการนับเส้นทางอัตโนมัติจนสิ้นสุดที่โหนด End ของแผนภาพกิจกรรม
   - **DrawingComponent** เป็นการทำงานที่พัฒนาขึ้น โดยใช้ในการร่างแผนภาพกิจกรรมให้แสดงบนหน้าจออีกครั้ง ในกรณีผู้พัฒนาต้องการนำแผนภาพกิจกรรมในรูปแบบของภาพ
   - **XMLParser** เป็นการทำงานที่ใช้ในการแปลงไฟล์นำเข้าจาก Visual Paradigm ให้อยู่ในรูปของ XML จากการอัพโหลดผ่านหน้าจอการใช้งาน เพื่อให้ระบบสามารถนำข้อมูลมาใช้ในการประมวลผลในขั้นตอนต่าง ๆ ต่อไป
   - **XMLUploader** เป็นการทำงานที่ใช้ในการลดระยะเวลาในการทดลองระบบโดย ให้ระบบดึงไฟล์ข้อมูลนำเข้าจาก Visual Paradigm จากตำแหน่งของไฟล์โดยตรง โดยไม่จำเป็นต้องผ่านการอัพโหลดผ่านหน้าจอการใช้งาน
2. **client.drawing** เป็นส่วนที่ใช้ในการวาดแผนภาพกิจกรรมใหม่อีกครั้ง ซึ่งแยกออก 2 ประเภทคือ การวาดเส้นเชื่อม และการวาดสัญลักษณ์ของแผนภาพกิจกรรม
3. **datasource** เป็นส่วนที่ใช้ในการเก็บข้อมูลคงที่ เปรียบเสมือนฐานข้อมูลในรูปแบบไฟล์ Java
4. **layout** เป็นส่วนที่ใช้ในการเก็บหน้าจอการทำงานในแต่ละฟังก์ชันการทำงานในแต่ละหน้า
5. **Entry Point** เป็นไฟล์หลักของระบบ ที่ทำงานในรูปแบบเว็บไซต์หน้าเดียว (Single Page Website



รูปที่ 22 แผนภาพคลาสเครื่องมืองานวิจัย

บทที่ 6
การประเมินผล

บทนี้เป็นส่วนที่แสดงถึงการทดลองด้วยชุดข้อมูลทดสอบ เพื่อประเมินผลแบบจำลองที่สร้างขึ้นในการตรวจสอบคุณภาพของแบบจำลองว่ามีอัตราความผิดพลาดมากน้อยเพียงใด โดยในงานวิจัยนี้ใช้มาตรวัด ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ และคุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.25 ในการประเมินคุณภาพของแบบจำลองการประมาณขนาดด้วยแอคทิวิตีพอยท์

6.1 ผลการตรวจสอบวิธีการประมาณด้วยวิธีแอคทิวิตีพอยท์

โครงการ 3 โครงการถูกใช้ในการตรวจสอบความถูกต้องของการประมาณขนาดด้วยวธีแอคทิวิตีพอยท์ซึ่งประกอบด้วย ซอฟต์แวร์ระดับองค์กรพัฒนาด้วยภาษาจาวา 2 โครงการ และ แอปพลิเคชันพัฒนาด้วยภาษาจาวาสคริปต์ 1 โครงการ ซึ่งค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (Average Magnitude Relative error: Average MRE) และ คุณภาพของการประมาณ (Prediction Quality: PRED) ถูกนำมาใช้ในการประเมินความถูกต้องของแบบจำลองการประมาณขนาดด้วยวิธีแอคทิวิตีพอยท์

ตารางที่ 17 แสดงถึงผลรวมค่าน้ำหนักจำนวนของกิจกรรมแต่ละโครงการของทั้งหมด 3 โครงการที่นำมาใช้ในการประเมินผล จากการนำสมการที่ 2 ในการคำนวน ซึ่งเป็นค่าที่แสดงถึงขนาดมุมมองระดับล่าง โดยมีรายละเอียดดังนี้

- โครงการที่ 1 ประกอบด้วยจำนวนกิจกรรมระดับเรียบง่าย 0 จำนวน จำนวนกิจกรรมระดับปานกลาง 3 จำนวน และจำนวนกิจกรรมระดับซับซ้อน 10 จำนวน ซึ่งส่งผลให้ค่าน้ำหนักรวมของกิจกรรมในโครงการที่ 1 มีค่าเท่ากับ 49
- โครงการที่ 2 ประกอบด้วยจำนวนกิจกรรมระดับเรียบง่าย 0 จำนวน จำนวนกิจกรรมระดับปานกลาง 2 จำนวน และจำนวนกิจกรรมระดับซับซ้อน 5 จำนวน ซึ่งส่งผลให้ค่าน้ำหนักรวมของกิจกรรมในโครงการที่ 2 มีค่าเท่ากับ 26
- โครงการที่ 3 ประกอบด้วยจำนวนกิจกรรมระดับเรียบง่าย 4 จำนวน จำนวนกิจกรรมระดับปานกลาง 7 จำนวน และจำนวนกิจกรรมระดับซับซ้อน 10 จำนวน ซึ่งส่งผลให้ค่าน้ำหนักรวมของกิจกรรมในโครงการที่ 3 มีค่าเท่ากับ 69

ตารางที่ 17 ผลรวมค่าน้ำหนักของกิจกรรมของแต่ละโครงการ

โครงการ	ระดับของจำนวนปัจจัยกิจกรรม	จำนวนของระดับกิจกรรม (NA level)	น้ำหนัก (Weight)	ค่าน้ำหนักของกิจกรรม (NA level * weight)
1	เรียบง่าย	0	2	0
	ปานกลาง	3	3	9
	ซับซ้อน	10	4	40
	ผลรวมค่าน้ำหนักของกิจกรรม (TAW= $\sum_{i=1}^{N}(\boldsymbol{NA\ level} * \boldsymbol{Weight})_i$)			49
2	เรียบง่าย	0	2	0
	ปานกลาง	2	3	6
	ซับซ้อน	5	4	20
	ผลรวมค่าน้ำหนักของกิจกรรม (TAW= $\sum_{i=1}^{N}(\boldsymbol{NA\ level} * \boldsymbol{Weight})_i$)			26
3	เรียบง่าย	4	2	8
	ปานกลาง	7	3	21
	ซับซ้อน	10	4	40
	ผลรวมค่าน้ำหนักของกิจกรรม (TAW= $\sum_{i=1}^{N}(\boldsymbol{NA\ level} * \boldsymbol{Weight})_i$)			69

ตารางที่ 18 แสดงถึงผลรวมค่าน้ำหนักจำนวนของเส้นทางแต่ละโครงการของทั้งหมด 3 โครงการที่นำมาใช้ในการประเมินผล จากการนำสมการที่ 3 ในการคำนวน ซึ่งเป็นค่าที่แสดงถึงขนาดมุมมองระดับล่าง โดยมีรายละเอียดดังนี้

- โครงการที่ 1 ประกอบด้วยจำนวนเส้นทางระดับเรียบง่าย 5 จำนวน จำนวนเส้นทางระดับปานกลาง 5 จำนวน และจำนวนเส้นทางระดับซับซ้อน 3 จำนวน ซึ่งส่งผลให้ค่าน้ำหนักรวมของเส้นทางในโครงการที่ 1 มีค่าเท่ากับ 52
- โครงการที่ 2 ประกอบด้วยจำนวนเส้นทางระดับเรียบง่าย 3 จำนวน จำนวนเส้นทางระดับปานกลาง 2 จำนวน และจำนวนเส้นทางระดับซับซ้อน 2 จำนวน ซึ่งส่งผลให้ค่าน้ำหนักรวมของเส้นทางในโครงการที่ 2 มีค่าเท่ากับ 29
- โครงการที่ 3 ประกอบด้วยจำนวนเส้นทางระดับเรียบง่าย 16 จำนวน จำนวนเส้นทางระดับปานกลาง 5 จำนวน และจำนวนเส้นทางระดับซับซ้อน 0 จำนวน ซึ่งส่งผลให้ค่าน้ำหนักรวมของเส้นทางในโครงการที่ 3 มีค่าเท่ากับ 36

ตารางที่ 18 ผลรวมค่าน้ำหนักของกิจกรรมของแต่ละโครงการ

โครงการ	**ระดับของจำนวนปัจจัยเส้นทาง**	**จำนวนของระดับเส้นทาง (NP level)**	**น้ำหนัก (Weight)**	**ค่าน้ำหนักของเส้นทาง (NP level * weight)**
1	เรียบง่าย	5	1	5
	ปานกลาง	5	4	20
	ซับซ้อน	3	9	27
	ผลรวมค่าน้ำหนักของเส้นทาง (TPW= $\sum_{i=1}^{N}(\boldsymbol{NP\ level} * \boldsymbol{Weight})_i$)			52
2	เรียบง่าย	3	1	3
	ปานกลาง	2	4	8
	ซับซ้อน	2	9	18
	ผลรวมค่าน้ำหนักของเส้นทาง (TPW= $\sum_{i=1}^{N}(\boldsymbol{NP\ level} * \boldsymbol{Weight})_i$)			29
3	เรียบง่าย	16	1	16
	ปานกลาง	5	4	20
	ซับซ้อน	0	9	0
	ผลรวมค่าน้ำหนักของเส้นทาง (TPW= $\sum_{i=1}^{N}(\boldsymbol{NP\ level} * \boldsymbol{Weight})_i$)			36

ตารางที่ 19 แสดงถึงผลรวมของแอคทิวิตีพอยท์แต่ละโครงการของทั้งหมด 3 โครงการที่นำมาใช้ในการประเมินผล จากการนำสมการที่ 1 ในการคำนวน ซึ่งเป็นค่าที่แสดงถึงขนาดรวมทั้งหมดของมุมมองระดับล่าง โดยมีรายละเอียดดังนี้

- โครงการที่ 1 ประกอบด้วย ค่าผลรวมค่าน้ำหนักของกิจกรรมหรือการกระทำ (TAW) ที่มีค่าเท่ากับ 49 ค่าผลรวมค่าน้ำหนักของเส้นทาง (TPW) ที่มีค่าเท่ากับ 52 ซึ่งส่งผลให้ค่าผลรวมของแอคทิวิตีพอยท์ในโครงการที่ 1 มีค่าเท่ากับ 101
- โครงการที่ 2 ประกอบด้วย ค่าผลรวมค่าน้ำหนักของกิจกรรมหรือการกระทำ (TAW) ที่มีค่าเท่ากับ 26 ค่าผลรวมค่าน้ำหนักของเส้นทาง (TPW) ที่มีค่าเท่ากับ 29 ซึ่งส่งผลให้ค่าผลรวมของแอคทิวิตีพอยท์ในโครงการที่ 2 มีค่าเท่ากับ 55
- โครงการที่ 3 ประกอบด้วย ค่าผลรวมค่าน้ำหนักของกิจกรรมหรือการกระทำ (TAW) ที่มีค่าเท่ากับ 69 ค่าผลรวมค่าน้ำหนักของเส้นทาง (TPW) ที่มีค่าเท่ากับ 36 ซึ่งส่งผลให้ค่าผลรวมของแอคทิวิตีพอยท์ในโครงการที่ 3 มีค่าเท่ากับ 105

ตารางที่ 19 ผลรวมของแอคทิวิตีของแต่ละโครงการ

โครงการ	**ค่าผลรวมค่าน้ำหนักของกิจกรรมหรือการกระทำ (TAW)**	**ค่าผลรวมค่าน้ำหนักของเส้นทาง (TPW)**	**ผลรวมของแอคทิวิตีพอยท์ (TAP = TAW + TPW)**
1	49	52	101
2	26	29	55
3	69	36	105

ตารางที่ 20 แสดงถึงผลรวมค่าน้ำหนักจำนวนของความซับซ้อนกิจกรรมแต่ละโครงการของทั้งหมด 3 โครงการที่นำมาใช้ในการประเมินผล จากการนำสมการที่ 4 ในการคำนวน ซึ่งเป็นค่าที่แสดงถึงขนาดมุมมองระดับบน โดยมีรายละเอียดดังนี้

- โครงการที่ 1 ประกอบด้วย ค่า McCabe Complexity เท่ากับ 12 ซึ่งอยู่ในระดับของจำนวนความซับซ้อนกิจกรรมในระดับซับซ้อน ซึ่งส่งผลให้ค่าน้ำหนักความซับซ้อนของกิจกรรมค่าเท่ากับ 20
- โครงการที่ 2 ประกอบด้วย ค่า McCabe Complexity เท่ากับ 5 ซึ่งอยู่ในระดับของจำนวนความซับซ้อนกิจกรรมในระดับซับซ้อน ซึ่งส่งผลให้ค่าน้ำหนักความซับซ้อนของกิจกรรมค่าเท่ากับ 4
- โครงการที่ 3 ประกอบด้วย ค่า McCabe Complexity เท่ากับ 21 ซึ่งอยู่ในระดับของจำนวนความซับซ้อนกิจกรรมในระดับซับซ้อน ซึ่งส่งผลให้ค่าน้ำหนักความซับซ้อนของกิจกรรมค่าเท่ากับ 20

Chulalongkorn University

ตารางที่ 20 ผลรวมค่าน้ำหนักความซับซ้อนกิจกรรมของแต่ละโครงการ

โครงการ	ค่า McCabe	ระดับของจำนวนปัจจัยความซับซ้อนกิจกรรม	จำนวนของระดับความซับซ้อนกิจกรรม (AC level)	น้ำหนัก (Weight)	ค่าน้ำหนักของความซับซ้อนกิจกรรม (AC level * weight)
1	13	ซับซ้อน	1	20	20
	ผลรวมค่าน้ำหนักของความซับซ้อนกิจกรรม (TACW= $\boldsymbol{AC\ level * Weight}$)				20
2	5	เรียบง่าย	1	4	4
	ผลรวมค่าน้ำหนักของความซับซ้อนกิจกรรม (TACW= $\boldsymbol{AC\ level * Weight}$)				4
3	21	ซับซ้อน	1	20	20
	ผลรวมค่าน้ำหนักของความซับซ้อนกิจกรรม (TACW= $\boldsymbol{AC\ level * Weight}$)				20

ตารางที่ 21 แสดงถึงขนาดแอคทิวิตีพอยท์ก่อนการปรับปรุงแต่ละโครงการของทั้งหมด 3 โครงการที่นำมาใช้ในการประเมินผล จากการนำสมการที่ 5 ในการคำนวน ซึ่งเป็นค่าที่แสดงถึงขนาดที่รอการปรับปรุงด้วยปัจจัยในการปรับปรุง โดยมีรายละเอียดดังนี้

- โครงการที่ 1 ประกอบด้วย ผลรวมของแอคทิวิตีพอยท์ที่มีค่าเท่ากับ 101 ค่าน้ำหนักความซับซ้อนของกิจกรรมที่มีค่า 20 ซึ่งส่งผลให้ค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงเท่ากับ 121
- โครงการที่ 2 ประกอบด้วย ผลรวมของแอคทิวิตีพอยท์ที่มีค่าเท่ากับ 55 ค่าน้ำหนักความซับซ้อนของกิจกรรมที่มีค่า 4 ซึ่งส่งผลให้ค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงเท่ากับ 59
- โครงการที่ 3 ประกอบด้วย ผลรวมของแอคทิวิตีพอยท์ที่มีค่าเท่ากับ 105 ค่าน้ำหนักความซับซ้อนของกิจกรรมที่มีค่า 20 ซึ่งส่งผลให้ค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงเท่ากับ 125

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 21 ขนาดของแอคทิวีตีพอยท์ก่อนการปรับปรุงของแต่ละโครงการ

โครงการ	ผลรวมของแอคทิวิตีพอยท์ (TAP)	ผลรวมค่าน้ำหนักความซับซ้อนกิจกรรม (TACW)	ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (UAP = TAP + TACW)
1	101	20	121
2	55	4	59
3	105	20	125

ตารางที่ 22 แสดงถึงขนาดซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์แต่ละโครงการของทั้งหมด 3 โครงการที่นำมาใช้ในการประเมินผล จากการนำสมการที่ 6 ในการคำนวน ซึ่งเป็นค่าที่แสดงถึงขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ โดยมีรายละเอียดดังนี้

- โครงการที่ 1 ประกอบด้วย ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงที่ค่าเท่ากับ 121 และค่าปัจจัยในการปรับค่าทั้ง 3 ปัจจัยที่มีค่า ปัจจัยความซับซ้อนทางเทคนิคที่ค่าเท่ากับ 1.12 ค่าปัจจัยความซับซ้อนทางด้านสิ่งแวดล้อมที่มีค่า 0.62 และค่าปัจจัยความเสี่ยงของคนที่ค่าเท่ากับ 3.28 ซึ่งส่งผลให้ค่าขนาดของแอคทิวิตีพอยท์เท่ากับ 275.59
- โครงการที่ 2 ประกอบด้วย ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงที่ค่าเท่ากับ 59 และค่าปัจจัยในการปรับค่าทั้ง 3 ปัจจัยที่มีค่า ปัจจัยความซับซ้อนทางเทคนิคที่ค่าเท่ากับ 1.01 ค่าปัจจัยความซับซ้อนทางด้านสิ่งแวดล้อมที่มีค่า 0.38 และค่าปัจจัยความเสี่ยงของคนที่ค่าเท่ากับ 2.32 ซึ่งส่งผลให้ค่าขนาดของแอคทิวิตีพอยท์เท่ากับ 52.53
- โครงการที่ 3 ประกอบด้วย ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงที่ค่าเท่ากับ 125 และค่าปัจจัยในการปรับค่าทั้ง 3 ปัจจัยที่มีค่า ปัจจัยความซับซ้อนทางเทคนิคที่ค่าเท่ากับ 1.16 ค่าปัจจัยความซับซ้อนทางด้านสิ่งแวดล้อมที่มีค่า 0.35 และค่าปัจจัยความเสี่ยงของคนที่ค่าเท่ากับ 1.84 ซึ่งส่งผลให้ค่าขนาดของแอคทิวิตีพอยท์เท่ากับ 93.38

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 22 ค่าขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ของแต่ละโครงการ

ระบบ	ขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (UAP)	ปัจจัยความซับซ้อนทางเทคนิค (TCF)	ปัจจัยความซับซ้อนทางด้านสิ่งแวดล้อม (ECF)	ปัจจัยความเสี่ยงของคน (PRF)	ขนาดซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ (AP = UAP * TCF * ECF * PRF)
1	121	1.12	0.62	3.28	275.59
2	59	1.01	0.38	2.32	52.53
3	125	1.16	0.35	1.84	93.38

ค่าที่ได้จากการประมาณค่าขนาดด้วยแอคทิวิตีพอยท์ของทุกโครงการของชุดข้อมูลที่ใช้ในการสร้างแบบจำลอง จะถูกนำมาใช้ในการหาแบบจำลองของการประมาณความพยายามในการพัฒนาซอฟต์แวร์ ที่สร้างด้วยวิธีสมการเส้นตรงแบบถดถอย (Linear regression) เพื่อหาค่าสัมประสิทธิ์ของสมการที่ 11 ดังถูกอธิบายในบทที่ 4 โดยกำหนดให้ตัวแปรอิสระคือค่าความพยายามในการพัฒนาซอฟต์แวร์จริง และค่าตัวแปรไม่อิสระคือค่าขนาดในหน่วยแอคทิวิตีพอยท์ โดยแบบจำลองที่ได้ในการประมาณความพยายามที่คาดหวังที่ได้ จะอยู่ในรูปแบบของสมการทำนายความพยายามในการพัฒนาซอฟต์แวร์ ดังถูกแสดงในสมการที่ 12 และมีรายละเอียดค่าทางสถิติดังแสดงในรูปที่ 24

$$Expected\ EFFORT = 39.086 + 1.064 * AP \qquad (12)$$

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.748[a]	.560	.505	67.56234

a. Predictors: (Constant), AP

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	46491.262	1	46491.262	10.185	.013[b]
	Residual	36517.363	8	4564.670		
	Total	83008.625	9			

a. Dependent Variable: EFFORT

b. Predictors: (Constant), AP

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	39.086	40.330		.969	.361
	AP	1.064	.334	.748	3.191	.013

a. Dependent Variable: EFFORT

รูปที่ 23 ข้อมูลสถิติการสร้างแบบจำลองความพยายามด้วยแอคทิวิตีพอยท์

ค่าแอคทิวิตีพอยท์ของโครงการทั้ง 3 ที่ได้นั้นดังแสดงในตารางที่ 23 จะถูกใช้ในการเป็นข้อมูลนำเข้าสู่แบบจำลองการประมาณความพยายามในสมการที่ 12 ที่ได้จากการสร้างด้วยวิธีสมการเส้นตรงเชิงเส้นถดถอย เพื่อใช้ในการหาค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MRE) และคุณภาพของการประมาณว่ามีอัตราความคลาดเคลื่อน (PRED) อยู่ที่เท่าใด โดยค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ของทั้ง 3 โครงการมีค่าเท่ากับ 0.168 ซึ่งได้จากค่าความคลาดเคลื่อนสัมพัทธ์ของแต่ละโครงการมาหาค่าเฉลี่ย อีกทั้งคุณภาพของการประมาณที่มีอัตราความคลาดเคลื่อนไม่เกิน 25% หรือ PRED(0.25) มีค่าเท่ากับ 0.67 โดยคิดจากจำนวนโครงการที่มีค่า RE ต่ำกว่า 0.25 ทั้งหมด 2 โครงการหารด้วยจำนวนของ 3 โครงการทั้งหมดที่นำมาทดสอบ นั้นแสดงถึงแบบจำลองนี้มีอัตราค่าประมาณที่มีความคลาดเคลื่อนที่ต่ำกว่า 25% อยู่ที่ 67%

ตารางที่ 23 ข้อมูลผลสรุปการประมาณขนาดของซอฟต์แวร์และความคลาดเคลื่อน

โครงการ	**แอคทิวิตีพอยท์ (AP)**	**ความพยายามที่คาดหวัง (Expected Effort)**	**ความพยายามจริง (Actual Effort)**	**ความคลาดเคลื่อนสัมพัทธ์ (RE)**
1	275.59	332.32	301	0.09
2	52.53	94.98	102	0.07
3	93.38	138.44	92	0.34
ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MRE)				0.168
คุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.25 (PRED(25))				0.67

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

6.2 ผลการเปรียบเทียบวิธีการประมาณด้วยวิธียูสเคสพอยท์และวิธีแอคทิวิตีพอยท์

หากใช้วิธีการประมาณด้วยยูสเคสพอยท์ และนำมาเปรียบเทียบกับค่าของการประมาณด้วยวิธีการแอคทิวิตีพอยท์ โดยที่ค่าที่ใช้ในการปรับขนาดในส่วนของ TCF และ ECF มีค่าเท่ากันในทั้งสองวิธี โดยการประมาณความพยายามด้วยยูสเคสพอยท์สามารถคำนวนได้จาก ขนาดยูสเคสพอยท์ * จำนวนชั่วโมงที่ใช้ต่อยูสเคสพอยท์ โดย Karner ผู้เสนอวิธียูสเคสพอยท์แนะนำให้ใช้ค่า 20 ชั่วโมงต่อยูสเคสพอยท์ ค่าความเพียรพยายามในการพัฒนาซอฟต์แวร์ของโครงการทั้ง 3 ที่ถูกประมาณด้วยวิธียูสเคสพอยท์ มีค่าเท่ากับ 2583.2, 330 และ 803.8 ตามลำดับ ดังสรุปภายในตารางที่ 24

ตารางที่ 24 ตารางเปรียบเทียบการประมาณความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธียูสเคสพอยท์และแอคทิวิตีพอยท์

โครงการ	ยูสเคสพอยท์ (UCP)	แอคทิวิตีพอยท์ (AP)	ค่าความพยายามที่คาดหวังด้วยวิธียูสเคสพอยท์ (Expected Effort with UCP)		ค่าความพยายามที่คาดหวังด้วยวิธีแอคทิวิตีพอยท์ (Expected Effort with AP)	ความพยายามจริง (Actual Effort)
			Hours /UCP	Expected Effort		
1	129.16	275.59	20	2583.2	332.32	301
2	16.50	52.53	20	330	94.98	102
3	40.19	93.38	20	803.8	138.44	92

โดยค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ของทั้ง 3 โครงการมีค่าเท่ากับ 0.82 ซึ่งได้จากค่าความคลาดเคลื่อนสัมพัทธ์ของแต่ละโครงการมาหาค่าเฉลี่ย อีกทั้งคุณภาพของการประมาณที่มีอัตราความคลาดเคลื่อนไม่เกิน 25% หรือ PRED(0.25) มีค่าเท่ากับ 0.00 โดยคิดจากจำนวนโครงการที่มีค่า RE ต่ำกว่า 0.25 หารด้วยจำนวนของโครงการทั้งหมดที่นำมาทดสอบ ซึ่งแสดงถึงแบบจำลองนี้มีอัตราค่าประมาณที่มีความคลาดเคลื่อนที่ต่ำกว่า 25% อยู่ที่ 0% หรือไม่มีโครงการใดเลยที่ความคลาดเคลื่อนน้อยกว่า 25% ดังอธิบายภายในตารางที่ 25

ตารางที่ 25 ตารางเปรียบเทียบความคลาดเคลื่อนสัมพัทธ์ของค่าความพยายามในการพัฒนาซอฟต์แวร์ด้วยการประมาณด้วยวิธียูสเคสพอยท์และวิธีแอคทิวิตีพอยท์

โครงการ	**ค่าความพยายามที่คาดหวังด้วยวิธียูสเคสพอยท์ (Expected Effort with UCP)**	**ค่าความพยายามที่คาดหวังด้วยวิธีแอคทิวิตีพอยท์ (Expected Effort with AP)**	**ความพยายามจริง (Actual Effort)**	**ความคลาดเคลื่อนสัมพัทธ์ (RE)**	
				UCP	**AP**
1	2583.17	332.32	301	0.88	0.09
2	330.07	94.98	102	0.69	0.07
3	803.88	138.44	92	0.89	0.34
ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์ (MRE)				0.82	0.168
คุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.25 (PRED(25))				0	0.67

เพื่อความถูกต้องที่มีความคลาดเคลื่อนน้อยลงนั้น จำเป็นต้องใช้ข้อมูลทดสอบและข้อมูลในการสร้างแบบจำลองที่มากกว่านี้ เพื่อประสิทธิผลของแบบจำลองที่สูงยิ่งขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 7
บทสรุปงานวิจัยและงานวิจัยในอนาคต

7.1 สรุปผลงานวิจัย

การประมาณขนาดของซอฟต์แวร์ เป็นหนึ่งในปัจจัยนำเข้าที่แบบจำลองการประมาณความพยายามใช้ในการประมาณ งานวิจัยนี้นำเสนอถึงการประมาณขนาดของซอฟต์แวร์และความพยายามโดยใช้แผนภาพกิจกรรมในชื่อว่า แอคทิวิตีพอยท์ เพื่อเน้นการนำเสนอปัจจัยเพิ่มเติมให้ครอบคลุมมากยิ่งขึ้นจากวิธียูสเคสพอยท์ โดยปัจจัยดังกล่าวนั้นประกอบด้วยจำนวนของกิจกรรมหรือการกระทำที่แทนด้วยจำนวนของทรานแซ็คชัน จำนวนของเส้นทางที่แทนด้วยจำนวนของซีเนริโอ และความซับซ้อนของแผนภาพกิจกรรมในระดับการปฏิสัมพันธ์ระหว่างยูสเคส โดยงานวิจัยนี้ได้นำค่าการปรับปรุงทางด้านความซับซ้อนทางเทคนิค ความซับซ้อนทางสภาพแวดล้อม จากวิธียูสเคสพอยท์มาใช้พร้อมนำเสนอปัจจัยในการปรับปรุงค่าเพิ่มเติมในมุมมองปัจจัยทางด้านความเสี่ยงของคน ซึ่งเป็นปัจจัยทางด้านคุณภาพที่มีผลต่อขนาดและความพยายาม

การประเมินแบบจำลองแอคทิวิตีพอยท์ ใช้ตัวชี้วัด 2 ปัจจัยคือ ค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์และคุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.25 โดยค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์หาได้จากค่าความคลาดเคลื่อนสัมพัทธ์ของข้อมูลทดสอบแต่ละโครงการ ซึ่งมีค่าเฉลี่ยความคลาดเคลื่อนสัมพัทธ์เท่ากับ 0.09 และค่าคุณภาพของการประมาณที่ความคลาดเคลื่อนต่ำกว่า 0.25 ของทั้ง 3 โครงการมีค่าเท่ากับ 1 ซึ่งแสดงถึงโครงการทุกโครงการที่ประมาณขนาดและความพยายามด้วยวิธีแอคทิวีตีพอยท์มีความคลาดเคลื่อนที่น้อยกว่า 25% อยู่ที่ 100%

7.2 งานวิจัยในอนาคต

สำหรับงานวิจัยในอนาคต นักวิจัยมีแผนที่จะเพิ่มประสิทธิภาพของโครงการด้วยการเพิ่มข้อมูลในการสร้างแบบจำลองและข้อมูลในการทดสอบที่มากกว่านี้ เพื่อให้ได้มาซึ่งค่าน้ำหนักที่เหมาะสมและครอบคลุมกับการพัฒนาซอฟต์แวร์ ณ ปัจจุบัน พร้อมกับเสนอถึงการกำหนดรูปแบบการให้คะแนนของตัวแปรที่ใช้ในการปรับปรุงค่า เพื่อให้ผู้นำแบบจำลองไปใช้มีการกำหนดคะแนนที่มีประสิทธิภาพยิ่งขึ้น โดยไม่ขึ้นอยู่กับการประมาณส่วนบุคคล รวมถึงเครื่องมือในการประมาณการอัตโนมัติที่ฉลาดยิ่งขึ้นขึ้นเพื่อให้นักวิจัยหรือองค์กรสามารถที่จะทำการทดลองด้วยชุดข้อมูลที่มี เพื่อให้ได้มาซึ่งการต่อยอดแบบจำลองที่เหมาะสมกับองค์กรที่เหมาะสมที่สุด

รายการอ้างอิง

[1] B. L. C. a. W. I, "Resource estimation in software engineering Encyclopedia of software engineering. ," *New York, John Wiley & Sons,* pp. 1160-1196, 2002.

[2] A. J. Albrecht, ""“Measuring Application Development Productivity,” Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, October 14–17, IBM Corporation (1979)," pp. 83–92, 1979.

[3] W. L. S. E. Kim, and D. Simmons, "An Effort Estimation by UML Points in Early Stage of Software Development," *Software Engineering Research,* 2016.

[4] ISO/IEC20926:2009, "Software measurement – IFPUG functional size measurement method," ed, 2010.

[5] ISO/IEC24750, "Software engineering-NESMA functional size measurement method version 2.1-Definitions and counting guidelines for the application of Function Points Analysis," *Geneva:International Organization for Standardization-ISO,* 2005.

[6] ISO/IEC19761, "Software Engineering-COSMIC-FFP," *Geneva: International Organization for Standardization-ISO,* 2003.

[7] N. A. S. Abdullah, N. I. A. Rusli, and M. F. Ibrahim, "A case study in COSMIC functional size measurement: Angry Bird Mobile Application," *Open Systems (ICOS), 2013 IEEE Conference on,* pp. 139-144, 2-4 Dec. 2013 2013.

[8] I. I. S. e. M. I. F. P. Analysis, "Counting Practices Manual," *Geneva: International Organization for Standardization-ISO,* p. 93, 2002.

[9] K. G, "Metrics for Objectory," *Diploma thesis, University of Linköping,* 1993.

[10] A. Sellami, H. Hakim, A. Abran, and H. Ben-Abdallah, "A measurement method for sizing the structure of UML sequence diagrams," *Information and Software Technology,* vol. 59, pp. 222-232, 2015.

[11] G. Costagliola, F. Ferrucci, G. Tortora, and G. Vitiello, "Class point: an approach for the size estimation of object-oriented systems," *Software Engineering, IEEE Transactions on,* vol. 31, pp. 52-74, 2005.

[12] W. Dennis, & Tegarden, "Systems Analysis and Design with UML, 3rd Edition," *John Wiley & Sons, Inc.,* p. 160, 2009.

[13] F. E. Allen, "Control flow analysis," *SIGPLAN Notices,* vol. 1–19, 1970.

[14] V. N. S. D.-R. T. T. B. Boehm, "A SLOC Counting Standard," *Center for Systems and Software Engineering , University of Southern California,* 2007.

[15] t. f. e. Wikipedia, "Use Case Points."

[16] P. Pandey, "Analysis of the Techniques for Software Cost Estimation," in Advanced Computing and Communication Technologies (ACCT)," *2013 Third International Conference on,* 2013.

[17] B. H. a. M. Cotterell, "Software Project Management: The McGRAW-HILL COMPANIES," 2002.

[18] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," *Software Engineering, IEEE Transactions on,* vol. vol. SE-4, pp. 345-361, 1978.

[19] J. D. S. d. S. I. F. de Barcelos Tronto, and N. Sant'Anna, "Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation," in Neural Networks," *International Joint Conference on,* pp. 771-776, 2007.

[20] A. I. a. A. Zahi, "Software cost estimation by classical and Fuzzy Analogy for Web Hypermedia Applications: A replicated study in Computational Intelligence and Data Mining (CIDM)," *2013 IEEE Symposium on,* pp. 207-213, 2013.

[21] G. Robiolo, C. Badano, and R. Orosco, "Transactions and paths: Two use case based metrics which improve the early effort estimation," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 422-425.

[22] M. D. 1976), "A Complexity Measure," *IEEE Transactions on Software Engineering: 308–320,* 1976.

[23] a. M. W. H. Madhavji, "The Impact of Non-Technical Factors on Software Architecture,," *University of Western Ontario and IBM Canada Ltd, Canada,* 2009.

[24] R. O. G. Robiolo, "An Alternative Method Employing Uses Cases for Early Effort Estimation," *Universidad Austral and Universidad Argentina de la Empresa,* 2012.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก
รายละเอียดยูสเคสเครื่องมืองานวิจัย

<table>
<tr><td colspan="2">Use case name: การนำเข้าไฟล์</td><td>ID: UC01</td><td>Important Level : High</td></tr>
<tr><td colspan="2">Primary Actor : นักวิจัย / ผู้จัดการโครงการ</td><td colspan="2">Use case Type : Detail , Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ</td></tr>
<tr><td colspan="4">Brief Description : การทำงานที่ใช้ในการนำเข้าข้อมูลของแผนภาพกิจกรรมที่ถูกออกแบบโดย Visual Paradigm ในรูปแบบของชนิดไฟล์ XML</td></tr>
<tr><td colspan="4">Association :</td></tr>
<tr><td colspan="4">Basic Flow of Events</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td colspan="2">1. ผู้ใช้เลือกเริ่มต้นการประมาณขนาดด้วยวิธีแอคทิวิตีพอยท์</td><td colspan="2">2. ระบบแสดงแบบฟอร์มในการนำเข้าไฟล์แผนภาพกิจกรรมในรูปแบบ XML</td></tr>
<tr><td colspan="2">3. ผู้ใช้เลือกค้นหาไฟล์ XML ของแผนภาพกิจกรรมที่ต้องการ</td><td colspan="2"></td></tr>
<tr><td colspan="2">4. ผู้ใช้เลือก Upload ไฟล์ XML ของแผนภาพกิจกรรม</td><td colspan="2">5. ระบบอัพโหลดไฟล์ลงบน Server เพื่อประมวลผล</td></tr>
<tr><td colspan="2"></td><td colspan="2">6. ระบบแสดงแผนภาพกิจกรรมทั้งหมดที่ถูกระบุในไฟล์ XML ดังกล่าว</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
4.1 ผู้ใช้เลือกยกเลิกการอัพโหลด
4.1a ระบบเคลียร์ข้อมูลของแผนภาพกิจกรรมที่ผู้ใช้ Upload ไฟล์ XML
4.2 ผู้ใช้เลือกย้อนกลับไปยังหน้าแรก
4.2a ระบบกลับไปยังหน้าแรกของเครื่องมืองานวิจัย</td></tr>
</table>

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Use case name: จำแนกแผนภาพกิจกรรมมุมมองระดับล่าง	ID: UC02	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : การทำงานที่ใช้ในการจำแนกแผนภาพกิจกรรมที่ถูกออกแบบเพื่อระบุแผนภาพใดบ้างที่เป็นแผนภาพในมุมมองระดับล่าง		
Association :		
Basic Flow of Events		
Actor Action	System Response	
	1. ระบบแสดงแผนภาพกิจกรรมทั้งหมดที่ได้จากไฟล์แผนภาพกิจกรรมในรูปแบบ XML	
2. ผู้ใช้เลือกแผนภาพกิจกรรมที่ต้องการที่อยู่ในประเภทของแผนภาพกิจกรรมในมุมมองระดับล่าง		
3. ผู้ใช้กดยืนยันการเลือกแผนภาพกิจกรรมในมุมมองระดับล่าง	4. ระบบทำการจำแนกแผนภาพกิจกรรมระดับล่างจากที่ผู้ใช้เลือก	
	5. ระบบแสดงแผนภาพกิจกรรมในมุมมองระดับล่างที่ผู้ใช้เลือก	
Alternate/Exceptional Flows: 5.1 ผู้ใช้เลือกเปลี่ยนแผนภาพกิจกรรมในมุมมองระดับล่างที่เลือกไว้กรณีเลือกแผนภาพกิจกรรมผิดในมุมมองระดับบนมาเป็นแผนภาพกิจกรรมระดับล่าง 5.1a ระบบทำการยกเลิกแผนภาพกิจกรรมที่ถูกเลือกเพื่อให้ผู้ใช้เลือกแผนภาพกิจกรรมใหม่อีกครั้ง		
Use case name: หาจำนวนกิจกรรมในแต่ละแผนภาพกิจกรรม	ID: UC03	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : การทำงานที่ใช้ในการหาจำนวนกิจกรรมในแต่ละแผนภาพกิจกรรมที่ถูกระบุให้เป็นแผนภาพในมุมมองระดับล่าง		

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

<table>
<tr><td colspan="3">Association :</td></tr>
<tr><td colspan="3">Basic Flow of Events</td></tr>
<tr><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1. ผู้ใช้ทำการกดคำนวณหาจำนวนกิจกรรมจากแผนภาพกิจกรรมระดับล่างที่เลือกไว้</td><td colspan="2">2. ระบบทำการนับจำนวนของกิจกรรมหรือเหตุการณ์ของแผนภาพกิจกรรมทั้งหมดที่ผู้ใช้เลือก</td></tr>
<tr><td></td><td colspan="2">3. ระบบทำการเปรียบเทียบจำนวนของแต่ละแผนภาพกิจกรรมเพื่อทำการเปรียบเทียบค่าน้ำหนักที่อยู่ในช่วงระดับที่ถูกกำหนดไว้</td></tr>
<tr><td></td><td colspan="2">4. ระบบทำการแสดงข้อมูลช่วงของน้ำหนักในแต่ละระดับของจำนวนกิจกรรมหรือเหตุการณ์ที่ถูกระบุในแผนภาพกิจกรรม</td></tr>
<tr><td colspan="3">Alternate/Exceptional Flows:
4.1 ผู้ใช้เลือกย้อนกลับเพื่อไปยังขั้นตอนการจำแนกแผนภาพกิจกรรมในมุมมองระดับล่าง กรณีเลือกแผนภาพกิจกรรมผิด เช่น เลือกแผนภาพกิจกรรมมุมมองระดับบนให้อยู่ในประเภทของมุมมองระดับล่าง
4.1a ระบบแสดงแผนภาพกิจกรรมที่ถูกเลือกเพื่อให้ผู้ใช้ทำการเปลี่ยนแปลงอีกครั้ง</td></tr>
<tr><td>Use case name: หาจำนวนเส้นทางในแต่ละแผนภาพกิจกรรม</td><td>ID: UC04</td><td>Important Level : High</td></tr>
<tr><td>Primary Actor : นักวิจัย / ผู้จัดการโครงการ</td><td colspan="2">Use case Type : Detail , Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ</td></tr>
<tr><td colspan="3">Brief Description : การทำงานที่ใช้ในการหาจำนวนเส้นทางในแต่ละแผนภาพกิจกรรมที่ถูกระบุให้เป็นแผนภาพในมุมมองระดับล่าง</td></tr>
<tr><td colspan="3">Association :</td></tr>
</table>

<table>
<tr><td colspan="3">Basic Flow of Events</td></tr>
<tr><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1. ผู้ใช้ทำการกดคำนวณหาจำนวนเส้นทางจากแผนภาพกิจกรรมระดับล่างที่เลือกไว้</td><td colspan="2">2. ระบบทำการคำนวณหาเส้นทางที่เป็นไปได้ทั้งหมดของแต่ละแผนภาพกิจกรรมในมุมมองระดับล่างที่ผู้ใช้เลือก</td></tr>
<tr><td></td><td colspan="2">3. ระบบทำการเปรียบเทียบค่าของจำนวนเส้นทางของแต่ละแผนภาพกิจกรรมเพื่อทำการเปรียบเทียบค่าน้ำหนักที่อยู่ในช่วงระดับที่ถูกกำหนดไว้</td></tr>
<tr><td></td><td colspan="2">4. ระบบทำการแสดงข้อมูลช่วงของน้ำหนักในแต่ละระดับของเส้นทางที่ถูกระบุในแผนภาพกิจกรรม</td></tr>
<tr><td colspan="3">Alternate/Exceptional Flows:
4.1 ผู้ใช้เลือกย้อนกลับเพื่อไปยังขั้นตอนการจำแนกแผนภาพกิจกรรมในมุมมองระดับล่าง กรณีเลือกแผนภาพกิจกรรมผิด เช่น เลือกแผนภาพกิจกรรมมุมมองระดับบนให้อยู่ในประเภทของมุมมองระดับล่าง
4.1a ระบบแสดงแผนภาพกิจกรรมที่ถูกเลือกเพื่อให้ผู้ใช้ทำการเปลี่ยนแปลงอีกครั้ง</td></tr>
<tr><td>Use case name: จำแนกแผนภาพกิจกรรมมุมมองระดับบน</td><td>ID: UC05</td><td>Important Level : High</td></tr>
<tr><td>Primary Actor : นักวิจัย / ผู้จัดการโครงการ</td><td colspan="2">Use case Type : Detail , Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ</td></tr>
<tr><td colspan="3">Brief Description : การทำงานที่ใช้ในการจำแนกแผนภาพกิจกรรมที่ถูกออกแบบเพื่อระบุแผนภาพใดบ้างที่เป็นแผนภาพในมุมมองระดับบน</td></tr>
<tr><td colspan="3">Association :</td></tr>
<tr><td colspan="3">Basic Flow of Events</td></tr>
<tr><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td></td><td colspan="2">1. ระบบแสดงแผนภาพกิจกรรมที่ยังไม่ถูกเลือกจากขั้นตอนการเลือกมุมมองระดับล่าง</td></tr>
</table>

2. ผู้ใช้เลือกแผนภาพกิจกรรมที่ต้องการที่อยู่ในประเภทของแผนภาพกิจกรรมในมุมมองระดับบน	
3. ผู้ใช้กดยืนยันการเลือกแผนภาพกิจกรรมในมุมมองระดับบน	4. ระบบทำการจำแนกแผนภาพกิจกรรมระดับบนจากที่ผู้ใช้เลือก
	5. ระบบแสดงแผนภาพกิจกรรมในมุมมองระดับบนที่ผู้ใช้เลือก

Alternate/Exceptional Flows:

5.1 ผู้ใช้เลือกเปลี่ยนแผนภาพกิจกรรมในมุมมองระดับบนที่เลือกไว้กรณีเลือกแผนภาพกิจกรรมผิดในมุมมองระดับล่างมาเป็นแผนภาพกิจกรรมระดับบน

5.1a ระบบทำการยกเลิกแผนภาพกิจกรรมที่ถูกเลือกเพื่อให้ผู้ใช้เลือกแผนภาพกิจกรรมใหม่อีกครั้ง

Use case name: หาค่า McCabe Complexity	ID: UC06	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : การทำงานที่ใช้ในการหาความซับซ้อนของแผนภาพกิจกรรมในระดับบนโดยใช้วิธี McCabe Complexity ในการระบุความซับซ้อนของแผนภาพกิจกรรม		
Association :		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้ทำการกดคำนวณหาความซับซ้อนของแผนภาพกิจกรรมจากแผนภาพกิจกรรมระดับบนที่เลือกไว้	2. ระบบทำการนับจำนวนของกิจกรรมหรือเหตุการณ์ที่ถูกระบุในแผนภาพกิจกรรมในมุมมองระดับบนที่ผู้ใช้เลือก	
	3. ระบบทำการนับเส้นเชื่อมทั้งหมดที่ถูกระบุในแผนภาพกิจกรรมในมุมมองระดับบนที่ผู้ใช้เลือก	

Chulalongkorn University

	4. ระบบทำการคำนวณหาความซับซ้อนของแผนภาพกิจกรรมด้วยวิธี McCabe Complexity
	5. ระบบทำการเปรียบเทียบค่าของความซับซ้อนของแผนภาพกิจกรรมเพื่อทำการเปรียบเทียบค่าน้ำหนักที่อยู่ในช่วงระดับที่ถูกกำหนดไว้
	6. ระบบทำการแสดงข้อมูลช่วงของน้ำหนักในแต่ละระดับของเส้นทางที่ถูกระบุในแผนภาพกิจกรรม
Alternate/Exceptional Flows: 6.1 ผู้ใช้เลือกย้อนกลับเพื่อไปยังขั้นตอนการจำแนกแผนภาพกิจกรรมในมุมมองระดับบน กรณีเลือกแผนภาพกิจกรรมผิด เช่น เลือกแผนภาพกิจกรรมมุมมองระดับล่างให้อยู่ในประเภทของมุมมองระดับบน 6.1a ระบบแสดงแผนภาพกิจกรรมที่ถูกเลือกเพื่อให้ผู้ใช้ทำการเปลี่ยนแปลงอีกครั้ง	

Use case name: หาค่าผลรวมของกิจกรรม (TAP)	ID: UC07	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : เป็นหาค่าผลรวมของกิจกรรมที่ได้จากผลรวมของปัจจัยที่ได้จากค่าผลรวมของค่าน้ำหนักกิจกรรมและผลรวมของค่าน้ำหนักของเส้นทาง		
Association :		
Basic Flow of Events		
Actor Action	System Response	
	1. ระบบหาผลรวมของกิจกรรม (TAP) จากค่าที่คำนวณได้จากผลรวมของค่าน้ำหนักกิจกรรม (TAW) และผลรวมของค่าน้ำหนักเส้นทาง (TPW) ในมุมมองระดับระดับล่าง	
Alternate/Exceptional Flows:		

Use case name: หาค่าผลรวมของค่าน้ำหนักกิจกรรม (TAW)	ID: UC08	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : เป็นการหาค่าผลรวมของค่าน้ำหนักกิจกรรม (TAW) ที่ได้จากผลรวมจากการถ่วงน้ำหนักของจำนวนกิจกรรมในแต่ละแผนภาพกิจกรรมในมุมมองระดับล่าง		
Association :		
Basic Flow of Events		
Actor Action	System Response	
	1. ระบบนำค่าจำนวนของจำนวนกิจกรรมของแต่ละแผภาพกิจกรรมถ่วงด้วยน้ำหนักที่อยู่ในช่วงที่กำหนดไว้ในแต่ละแผนภาพกิจกรรม	
	2. ระบบหาผลรวมของค่าน้ำหนักกิจกรรม (TAW) จากค่าที่ได้จากการถ่วงน้ำหนักในแต่ละแผนภาพกิจกรรมในมุมมองระดับล่าง	
Alternate/Exceptional Flows:		

Use case name: หาค่าผลรวมของค่าน้ำหนักเส้นทาง (TPW)	ID: UC09	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : เป็นการหาค่าผลรวมของค่าน้ำหนักเส้นทาง (TPW) ที่ได้จากผลรวมจากการถ่วงน้ำหนักของจำนวนเส้นทางในแต่ละแผนภาพกิจกรรมในมุมมองระดับล่าง		
Association :		
Basic Flow of Events		
Actor Action	System Response	
	1. ระบบนำค่าเส้นทางของแต่ละแผนภาพกิจกรรมถ่วงด้วยน้ำหนักที่อยู่ในช่วงที่กำหนดไว้ในแต่ละแผนภาพกิจกรรม	

	2. ระบบหาผลรวมของค่าน้ำหนักเส้นทาง (TPW) ของค่าที่ได้จากการถ่วงน้ำหนักของแผนภาพกิจกรรมในมุมมองระดับล่าง	
Alternate/Exceptional Flows:		
Use case name: หาค่าน้ำหนักความซับซ้อนของกิจกรรม (TACW)	ID: UC10	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : เป็นการหาค่าน้ำหนักความซับซ้อนของกิจกรรม (TACW) ที่ได้จากการถ่วงน้ำหนักของค่าความซับซ้อนของแผนภาพกิจกรรมในมุมมองระดับบน		
Association :		
Basic Flow of Events		
Actor Action	System Response	
	1. ระบบนำค่าความซับซ้อนของแผนภาพกิจกรรมถ่วงด้วยน้ำหนักที่อยู่ในช่วงที่กำหนดไว้ในแต่ละแผนภาพกิจกรรม	
	2. ระบบหาผลรวมของค่าน้ำหนักกิจกรรม (TAW) ของค่าที่ได้จากการถ่วงน้ำหนักของแผนภาพกิจกรรมในมุมมองระดับบน	
Alternate/Exceptional Flows:		
Use case name: หาค่าแอคทิวิตีพอยท์ก่อนการปรับปรุงโครงการ (UAP)	ID: UC11	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : เป็นการหาค่าแอคทิวิตีพอยท์ก่อนการปรับปรุงจากการนำค่าที่ได้จากมุมมองระดับบนและระดับล่างรวมกัน		
Association :		

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Basic Flow of Events		
Actor Action	System Response	
	1. ระบบหาค่าแอคทิวิตีพอยท์ก่อนการปรับปรุง (UAP) จากค่าที่คำนวณได้จากผลรวมของกิจกรรม (TAP) ที่ได้จากแผนภาพกิจกรรมในมุมมองระดับล่าง และผลรวมของค่าน้ำหนักความซับซ้อนของแผนภาพกิจกรรม (TACW) ที่ได้จากแผนภาพกิจกรรมในมุมมองระดับบน	
Alternate/Exceptional Flows:		
Use case name: หาค่าปัจจัยในการปรับปรุงขนาด	ID: UC12	Important Level : High
Primary Actor : นักวิจัย / ผู้จัดการโครงการ	Use case Type : Detail , Essential	
Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ		
Brief Description : เป็นการหาค่าปัจจัยในการปรับปรุงขนาดที่ได้จากการถ่วงน้ำหนักของค่าที่ถูกระบบในแต่ละระดับของ ผลรวมความซับซ้อนทางด้านเทคนิค (TCF) ผลรวมความซับซ้อนทางด้านสภาพแวดล้อม (ECF) และ ผลรวมของปัจจัยความเสี่ยงของมนุษย์ (PRF)		
Association :		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้เลือกคำนวณหาปัจจัยที่ใช้ในการปรับปรุงค่าน้ำหนัก	2. ระบบแสดงรายละเอียดของแต่ละปัจจัยย่อยของค่าปัจจัยความซับซ้อนทางเทคนิค (TCF) พร้อมค่าน้ำหนักมาตรฐานของปัจจัยนั้น ๆ	
	3. ระบบแสดงรายละเอียดของแต่ละปัจจัยย่อยของค่าปัจจัยความซับซ้อนทางสภาพแวดล้อม (ECF) พร้อมค่าน้ำหนักมาตรฐานของปัจจัยนั้น ๆ	

	4. ระบบแสดงรายละเอียดของแต่ละปัจจัยย่อยของค่าปัจจัยความซับซ้อนทางความเสี่ยงมนุษย์ (PRF) พร้อมค่าน้ำหนักมาตรฐานของปัจจัยนั้น ๆ
	5. ระบบแสดงแบบฟอร์มการกำหนดระดับของค่าปัจจัยความซับซ้อนทางเทคนิค (TCF)
	6. ระบบแสดงแบบฟอร์มการกำหนดระดับของค่าปัจจัยความซับซ้อนทางสภาพแวดล้อม (ECF)
	7. ระบบแสดงแบบฟอร์มการกำหนดระดับของค่าปัจจัยความซับซ้อนทางความเสี่ยงมนุษย์ (PRF)
8. ผู้ใช้กำหนดค่าน้ำหนักของค่าปัจจัยความซับซ้อนทางเทคนิค (TCF) ค่าปัจจัยความซับซ้อนทางสภาพแวดล้อม (ECF) และค่าปัจจัยความเสี่ยงมนุษย์ (PRF)	
9. ผู้ใช้เลือกคำนวณค่าปัจจัยที่ใช้ในการปรับปรุงขนาด	10. ระบบคำนวณค่าปัจจัยการปรับปรุงขนาด
	11. ระบบแสดงค่าปัจจัยของการปรับปรุงขนาด
Alternate/Exceptional Flows: 9.1 ผู้ใช้เลือกย้อนกลับเพื่อทำการกำหนดค่าน้ำหนักในแต่ละปัจจัยใหม่ 9.1a ระบบแสดงค่าน้ำหนักเดิมที่ผู้ใช้เลือกก่อนหน้าเพื่อให้ผู้ใช้แก้ไขค่าน้ำหนักดังกล่าว	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

<table>
<tr><td>Use case name: หาค่าขนาดในหน่วยแอคทิวิตีพอยท์ (AP)</td><td>ID: UC13</td><td>Important Level : High</td></tr>
<tr><td>Primary Actor : นักวิจัย / ผู้จัดการโครงการ</td><td colspan="2">Use case Type : Detail , Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ</td></tr>
<tr><td colspan="3">Brief Description : เป็นการหาค่าขนาดในหน่วยแอคทิวิตีพอยท์ (AP) ที่ได้จากการนำค่าน้ำหนักของแอคทิวิตีพอยท์ก่อนการปรับปรุง ปรับค่าด้วยค่าปัจจัยในการปรับปรุงขนาดเพื่อให้ได้ขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์</td></tr>
<tr><td colspan="3">Association :</td></tr>
<tr><td colspan="3">Basic Flow of Events</td></tr>
<tr><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td></td><td colspan="2">1. ระบบแสดงค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (UAP)</td></tr>
<tr><td></td><td colspan="2">2. ระบบแสดงค่าปัจจัยที่ใช้ในการปรับปรุงขนาด (Adjusting Factor)</td></tr>
<tr><td></td><td colspan="2">3. ระบบหาขนาดของซอฟต์แวร์ในหน่วนของแอคทิวิตีพอยท์ (AP) จากผลคูณของขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงและค่าปัจจัยที่ใช้ในการปรับปรุงขนาด</td></tr>
<tr><td colspan="3">Alternate/Exceptional Flows:</td></tr>
<tr><td>Use case name: ประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์</td><td>ID: UC14</td><td>Important Level : High</td></tr>
<tr><td>Primary Actor : นักวิจัย / ผู้จัดการโครงการ</td><td colspan="2">Use case Type : Detail , Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests : นักวิจัย / ผู้จัดการโครงการ</td></tr>
<tr><td colspan="3">Brief Description : เป็นการประมาณค่าความพยายามในการพัฒนาซอฟต์แวร์โดยใช้ขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์มาใช้เป็นข้อมูลนำเข้าในแบบจำลองการประมาณความพยายามการพัฒนาซอฟต์แวร์โดยมีหน่วยเป็น คน/ชั่วโมง</td></tr>
<tr><td colspan="3">Association :</td></tr>
</table>

Basic Flow of Events	
Actor Action	System Response
1. ผู้ใช้เลือกประมาณความพยายามการพัฒนาซอฟต์แวร์ด้วยขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์	2. ระบบทำการประมาณความพยายามในการพัฒนาซอฟต์แวร์จากขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์
	3. ระบบแสดงผลการประมาณขนาดของการประมาณความพยายามในการพัฒนาซอฟต์แวร์ในหน่วยของ คน/ชั่วโมง
4. ผู้ใช้กดเสร็จสิ้นขั้นตอนการประมาณความพยายามด้วยวิธีแอควิตีพอยท์	5. ระบบเก็บข้อมูลการประมาณเพื่อใช้ในการเก็บสถิติต่อไป
Alternate/Exceptional Flows: 1. ผู้ใช้กดเสร็จสิ้นเพียงขั้นตอนการประมาณขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ 1.a ระบบเก็บข้อมูลการประมาณขนาดในหน่วยของแอคทิวิตีพอยท์เพื่อใช้เป็นข้อมูลสถิติต่อไป 4.1 ผู้ใช้เลือกย้อนกลับในขั้นตอนก่อนหน้าเพื่อทำการดูผลลัพธ์ของขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์อีกครั้ง 4.1a ระบบกลับไปยังหน้าการสรุปผลการประมาณขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์	

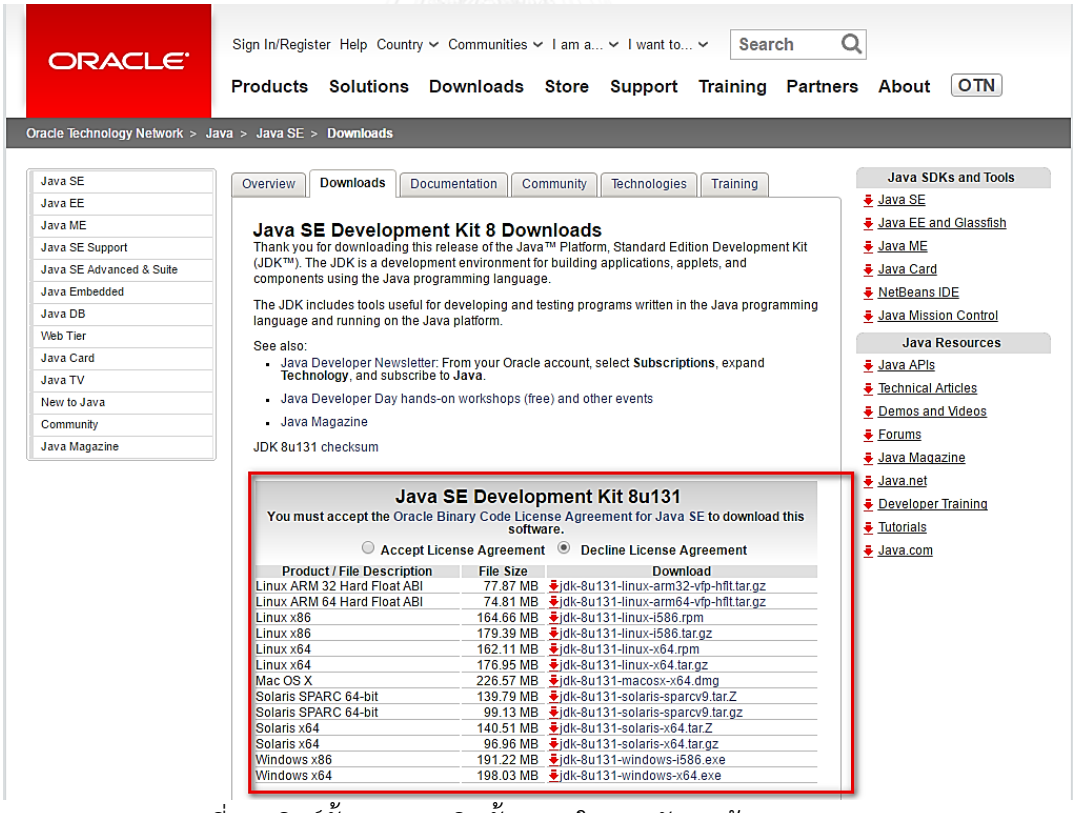
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข
วิธีการติดตั้งเครื่องมืองานวิจัยสำหรับการพัฒนา

เครื่องมืองานวิจัยที่ถูกนำเสนอนี้ถูกพัฒนาด้วยภาษา Java โดยใช้เครื่องมือพัฒนา Eclipse เวอร์ชัน Java EE Developers (เวอร์ชัน 3.4 หรือใหม่กว่า) และ Framework Vaadin สำหรับพัฒนาหน้าส่วนต่อประสานกราฟฟิคผู้ใช้งาน (Graphic User Interface) และ SourceTree ที่เป็นเครื่องมือในการทำการควบคุมเวอร์ชันชุดรหัสข้อมูล (Version Control) โดยชุดรหัสข้อมูลดังกล่าวถูกเก็บอยู่ใน www.bitbucket.org

1. วิธีการติดตั้ง JDK เพื่อใช้ในการพัฒนาด้วยภาษา Java

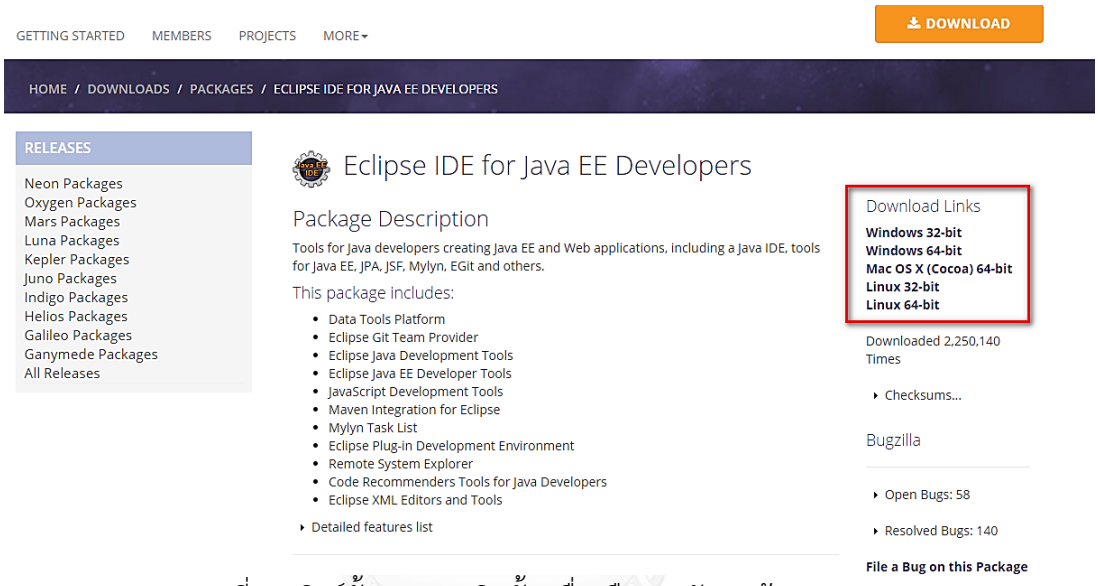
เนื่องจากเครื่องมืองานวิจัยนี้ถูกพัฒนาด้วยภาษา Java ผู้พัฒนาจึงจำเป็นต้องติดตั้ง JDK เพื่อให้ผู้พัฒนาสามารถคอมไพล์โครงการจากภาษา Java ได้ โดยในงานวิจัยนี้ใช้ JDK version 8 ซึ่งผู้พัฒนาสามารถดาวน์โหลดตามระบบปฏิบัติการของผู้พัฒนาที่ใช้อยู่ได้ที่ลิงค์ http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html ตามรูปที่ 25



รูปที่ 24 ลิงค์ขั้นตอนการติดตั้ง JDK ในการพัฒนาด้วยภาษา Java

2. วิธีการติดตั้ง Eclipse

เริ่มต้นทางผู้ที่ต้องการพัฒนาต่อยอดเครื่องมืองานวิจัยต้องทำการเลือกเครื่องมือในการพัฒนาซอฟต์แวร์ก่อนเป็นอันดับต้นตามที่ถนัด ซึ่งสามารถที่จะใช้เครื่องมือในการพัฒนาใด ๆ ก็ได้ เช่น Netbean, Editplus, Sublime, Visual Code เป็นต้น แต่ในตัวอย่างต่อไปนี้จะใช้ Eclipse IDE for Java EE Developers (Mar.1) โดยสามารถดาวน์โหลดได้ที่ https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/mars1 ตามรูปที่ 26 โดยการติดตั้งเพียงแตกไฟล์ Zip ใน Folder ที่ต้องการ



รูปที่ 25 ลิงค์ขั้นตอนการติดตั้งเครื่องมือการพัฒนาด้วย Eclipse

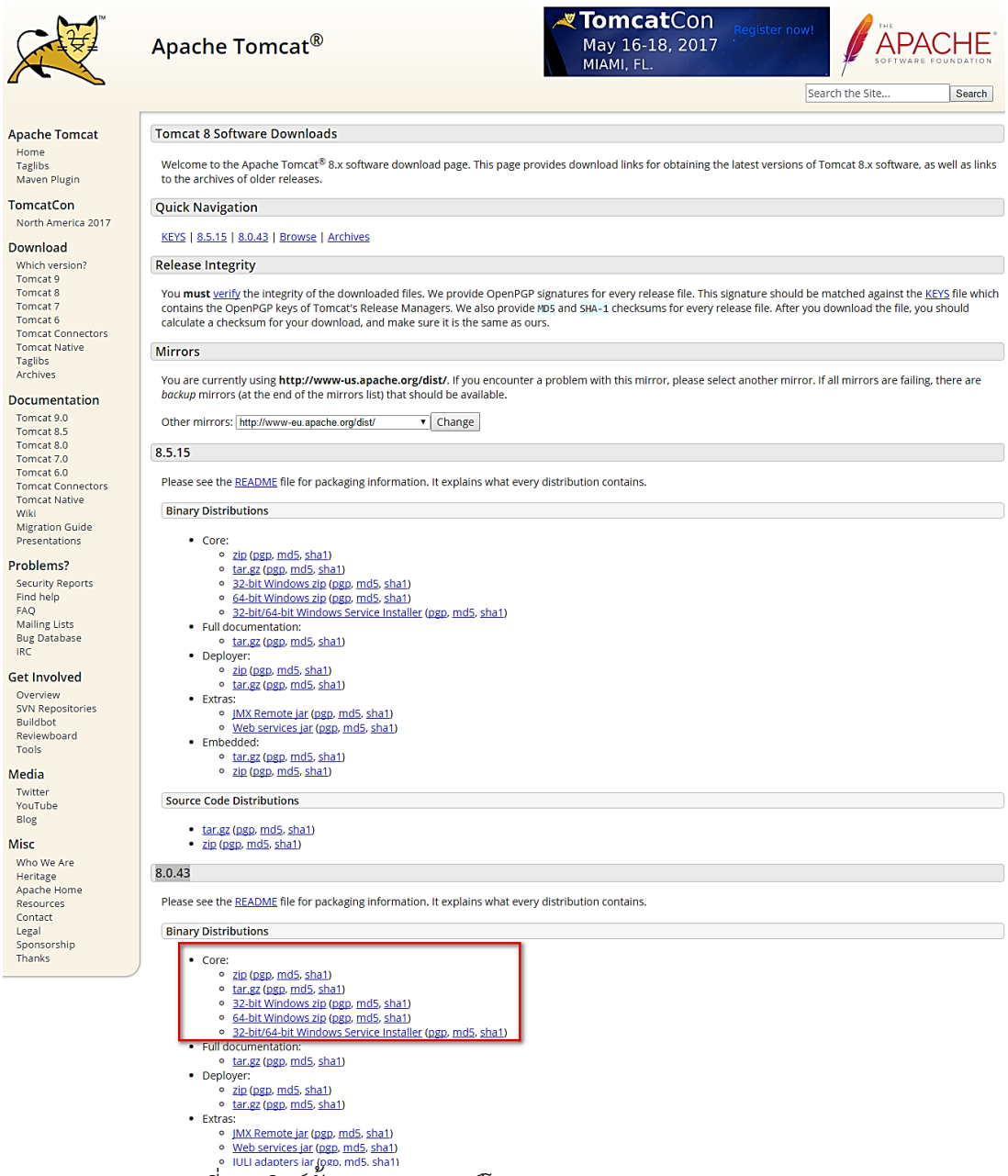
3. วิธีการติดตั้ง Tomcat Server

สำหรับการรันตัวโครงการให้สามารถเปิดบนเว็บเบราเซอร์ได้นั้น ผู้พัฒนาจำเป็นต้องทำการติดตั้งเซอร์เวอร์เสมือนจำลองภายในเครื่อง โดยในตัวอย่างนี้จะนำ Server Tomcat Version 8 (Revision 8.0.43) ในการรันตัวเครื่องมืองานวิจัย โดยผู้พัฒนาสามารถดาวน์โหลดได้ตามลิ้งค์นี้ https://tomcat.apache.org/download-80.cgi ในรูปแบบ Zip โดยทำการแตกในโฟลเดอร์ที่ต้องการ ตามรูปที่ 27

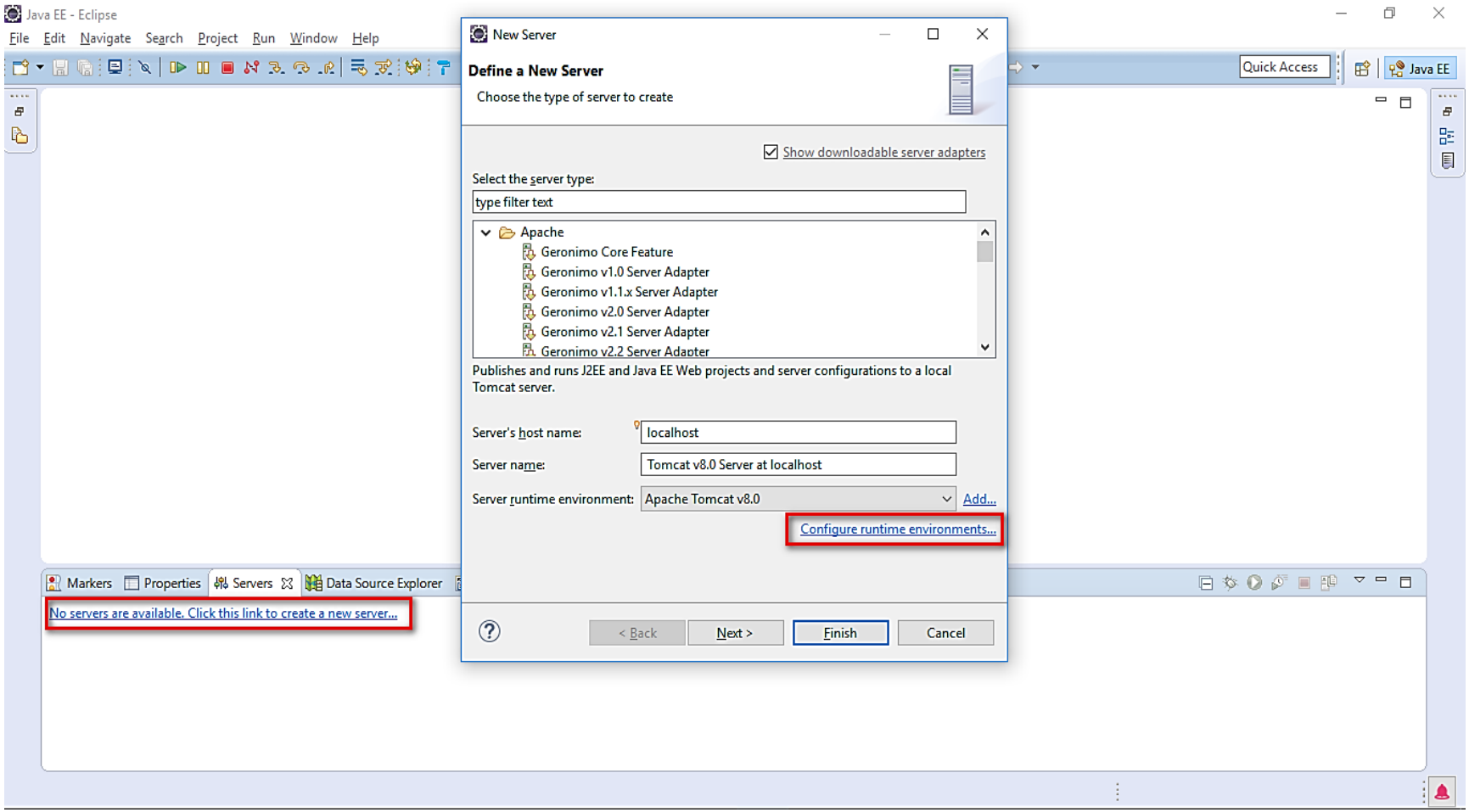
เมื่อทำการติดตั้ง Server Tomcat เสร็จสิ้น ผู้พัฒนาจำเป็นต้องตั้งค่าเครื่องมือการพัฒนา Eclipse โดยใช้ Tomcat ที่ติดตั้งดังกล่าวในการรันชุดรหัสข้อมูลของเครื่องมืองานวิจัยนี้ โดยให้ผู้พัฒนาทำการเปิด Eclipse ขึ้นแล้วเลือกแท็บ Servers และคลิก No servers are available. Click this link to create a new server และเลือก Configure runtime environment ตามรูปที่ 28 เพื่อทำการติดตั้ง Server Tomcat และทำการเลือกตำแหน่งของ Server Tomcat ที่ติดตั้งในเครื่องก่อนหน้านี้ตามรูปที่ 29 – 30 และทำการกด Finish เพื่อกลับไปยังรูปที่ 28 และทำ

การเลือก Apache > Tomcat v.8.0 Server ส่วนของ Select the server type และกด Finish เพื่อทำการติดตั้ง Server Tomcat ใน Eclipse เสร็จสิ้น

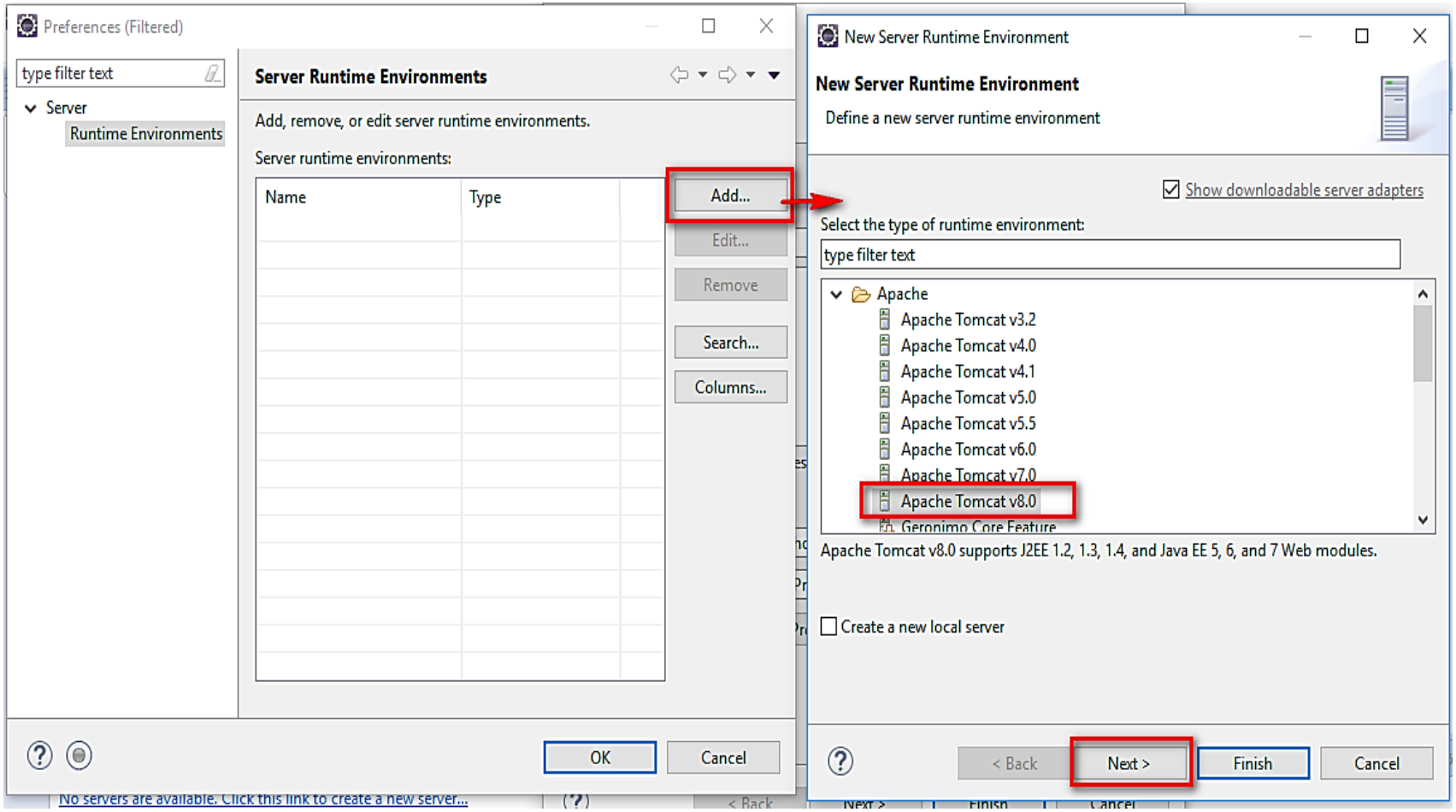
เมื่อทำการติดตั้งเสร็จสิ้นในแท็บ Servers จะปรากฏ Server Tomcat ที่ติดตั้งก่อนหน้า เพื่อให้ผู้พัฒนาสามารถ Start Server ได้ภายใน Eclipse เพื่อใช้ในการรันตัวโครงการต่อไป โดยการ Start ทำได้โดยการคลิกขวาที่ Server ที่ติดตังแล้วเลือกปุ่ม Start ดังแสดงในรูปที่ 28



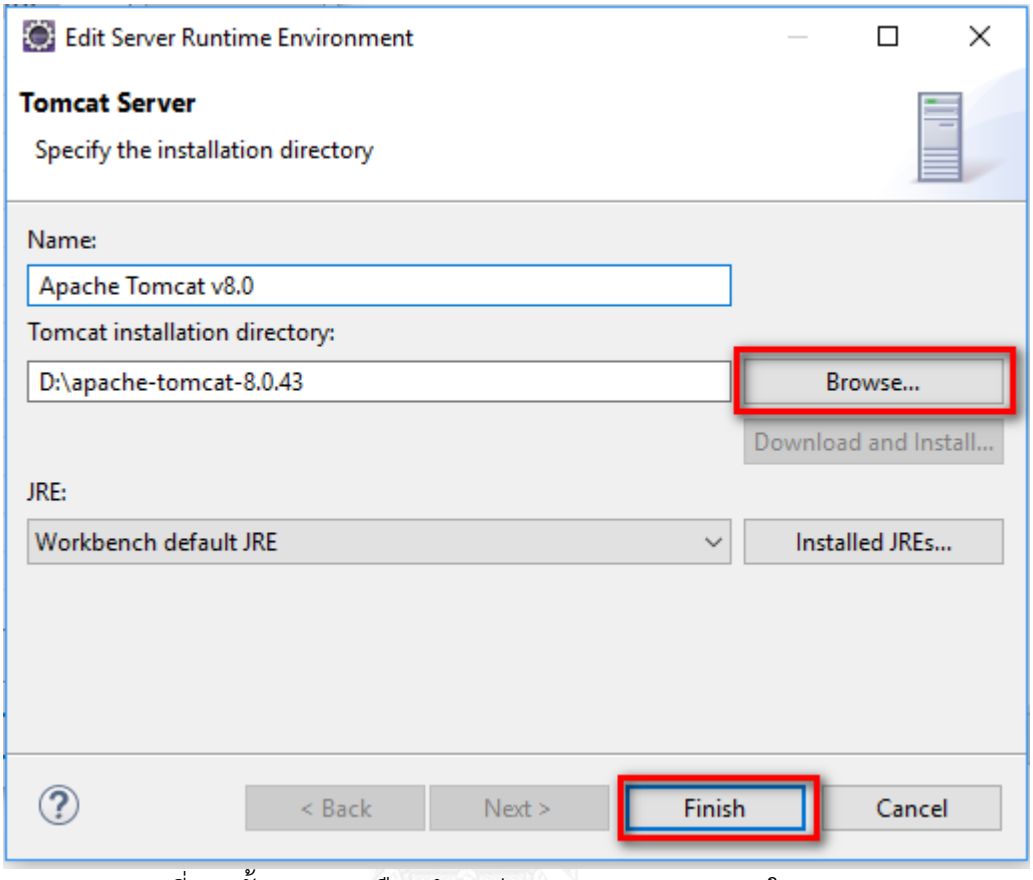
รูปที่ 26 ลิงค์ขั้นตอนการดาวน์โหลด Server Tomcat version 8



รูปที่ 27 ขั้นตอนการตั้งค่า Server Tomcat ใน Eclipse

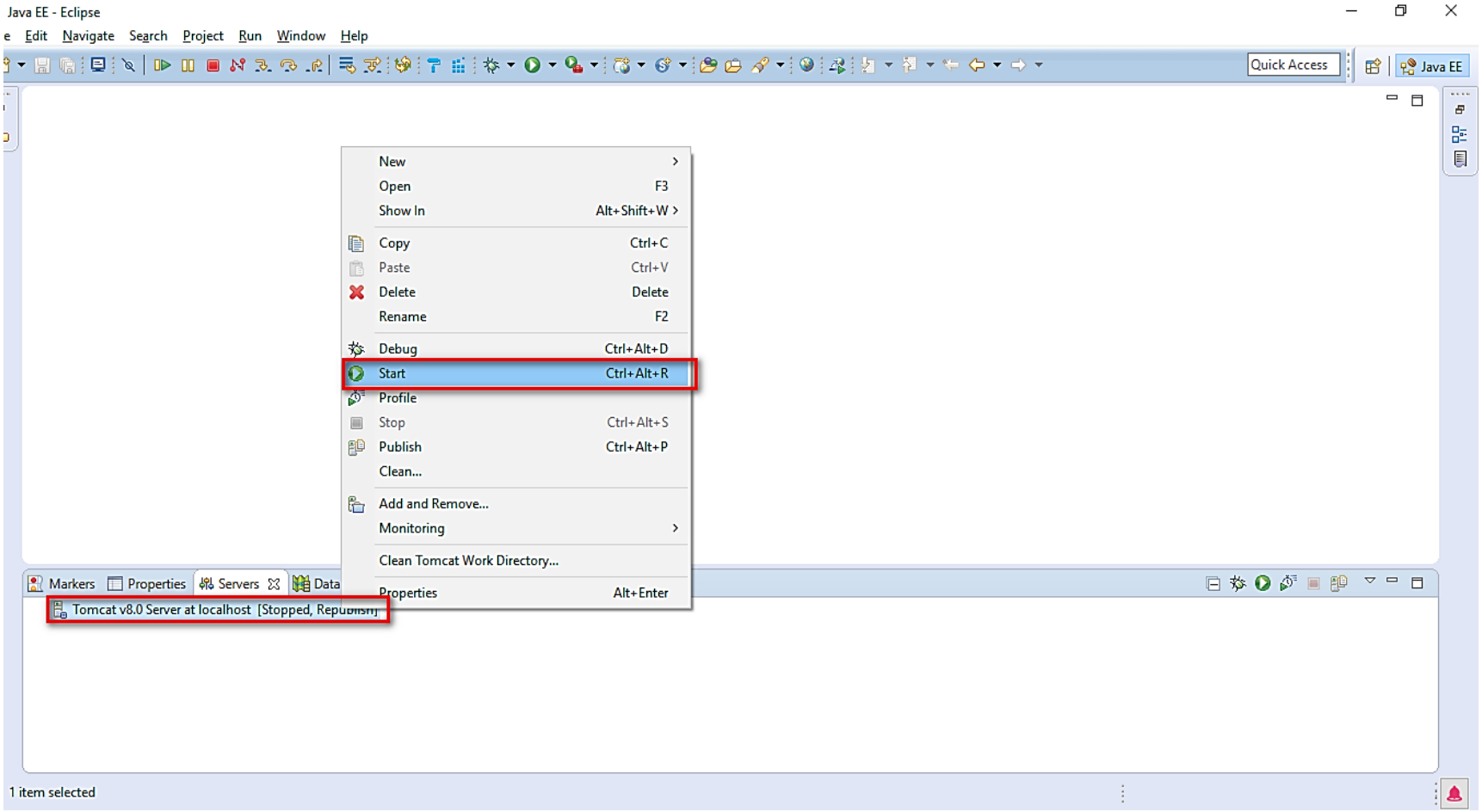


รูปที่ 28 ขั้นตอนการเลือก Server Tomcat ใน Eclipse



รูปที่ 29 ขั้นตอนการเลือกตำแหน่งของ Server Tomcat ใน Eclipse

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 30 ขั้นตอนการ Start Server Tomcat ใน Eclipse

4. วิธีการติดตั้ง Vaadin Eclipse Plugin

เมื่อทำการติดตั้งเครื่องมือการพัฒนาเสร็จสิ้นแล้ว ทางผู้พัฒนาต้องทำการติดตั้ง Framework Vaadin ซึ่งเป็น Framework สำเร็จรูป ที่จัดสรรคอมโพเนนท์พร้อมใช้ที่สำคัญในการพัฒนาซอฟต์แวร์ อาทิเช่น ปุ่ม ตาราง เป็นต้น เพื่อให้ผู้พัฒนาสามารถใช้คอมโพเนนท์ดังกล่าวได้เลย โดยไม่ต้องสร้างใหม่ตั้งแต่ต้น อีกทั้ง Vaadin ยังสามารถทำงานฝั่ง Server ในตัวที่ไม่เพียงแค่ทำงานเพียงฝั่ง Client ที่แสดงหน้าตาการใช้งานเพียงเท่านั้น โดยสามารถอ่านรายละเอียดเพิ่มเติมได้ตามลิงค์นี้ https://vaadin.com

ทาง Vaadin ได้พัฒนา Plugin สำหรับ Eclipse เพื่อให้ผู้พัฒนาสามารถใช้การทำงานลากและวาง (Drag and Drop) ของคอมโนเนนท์ที่ทาง Vaadin เตรียมไว้ให้ เพื่อช่วยลดเวลาในการเขียนชุดรหัสในส่วนของหน้าตาการใช้งานให้รวดเร็วและมีประสิทธิภาพยิ่งขึ้น โดยผู้พัฒนาจำเป็นต้องติดตั้ง Plugin สำหรับ Eclipe ได้ตามลิงค์ https://vaadin.com/eclipse โดยทำการก็อปปี้ลิงค์ของ Plugin ตาม Version ของ Eclipse ที่ทางผู้พัฒนาใช้ โดยในตัวอย่างนี้ทางผู้วัจยวิจัยใช้ Eclipse Version 4.5.1 จึงต้องใช้ลิงค์ของ https://vaadin.com/eclipse ในการติดตั้ง โดยดูรายละเอียดได้ดังรูปที่ 32

เมื่อผู้พัฒนาทำการก็อปปี้ลิงค์ดังกล่าวให้ทำการเปิด Eclipse โดยเลือกเมนู Help > Install New Software เพื่อทำการติดตั้ง Plugin สำหรับ Eclipse ดังแสดงในรูปที่ 33 ต่อมาผู้พัฒนาให้ทำการวางลิงค์ Vaadin Plugin ที่ก็อปปี้ไว้ในช่อง Work with และกดปุ่ม Enter เพื่อทำการเลือก Plugin ที่แสดงบนหน้าจอ โดยการติดตั้งให้ทำการกด Next เรื่อย ๆ จนเสร็จสิ้นขั้นตอนของการติดตั้ง พร้อมรีสตาร์ท Eclipse ใหม่อีกครั้ง ดังแสดงในรูปที่ 34

หากติดตั้ง Plugin Vaadin เสร็จสิ้น ผู้พัฒนาสามารถที่จะสร้างโครงการพัฒนาด้วย Vaadin ได้ โดยเลือกที่เมนู File > New > Vaadin 8 Project ก็สามารถใช้คอมโนเนนท์สำเร็จรูปได้ในโครงการนั้นๆ ดังแสดงในรูปที่ 35

vaadin }> Framework Elements Community Tools&Services Pricing Company 2 Sign In Register

Vaadin Plug-in for Eclipse

The easiest way to get started with Vaadin development is to install the Vaadin Plug-in for Eclipse. You don't need to download or install anything else but the plug-in to your Eclipse IDE for Java EE Developers (3.4 or newer) to start development. Learn more about the features.

Install the plugin

Option 1: Drag and drop install

This is the easiest way to install the plugin. Drag and drop the following icon into a running Eclipse workspace to install Vaadin Plugin for Eclipse:

Install

Option 2: Download and install

To install the plugin:

- Select *Install New Software...* from the *Help* -menu
- Add new update site:
  - **Eclipse 4.4 and newer: *http://vaadin.com/eclipse***
  - **Eclipse 4.3: *http://vaadin.com/eclipse/4.3***
  - **Eclipse 3.5: *http://vaadin.com/eclipse/3.5***
  - **Eclipse 3.4: *http://vaadin.com/eclipse/3.4***

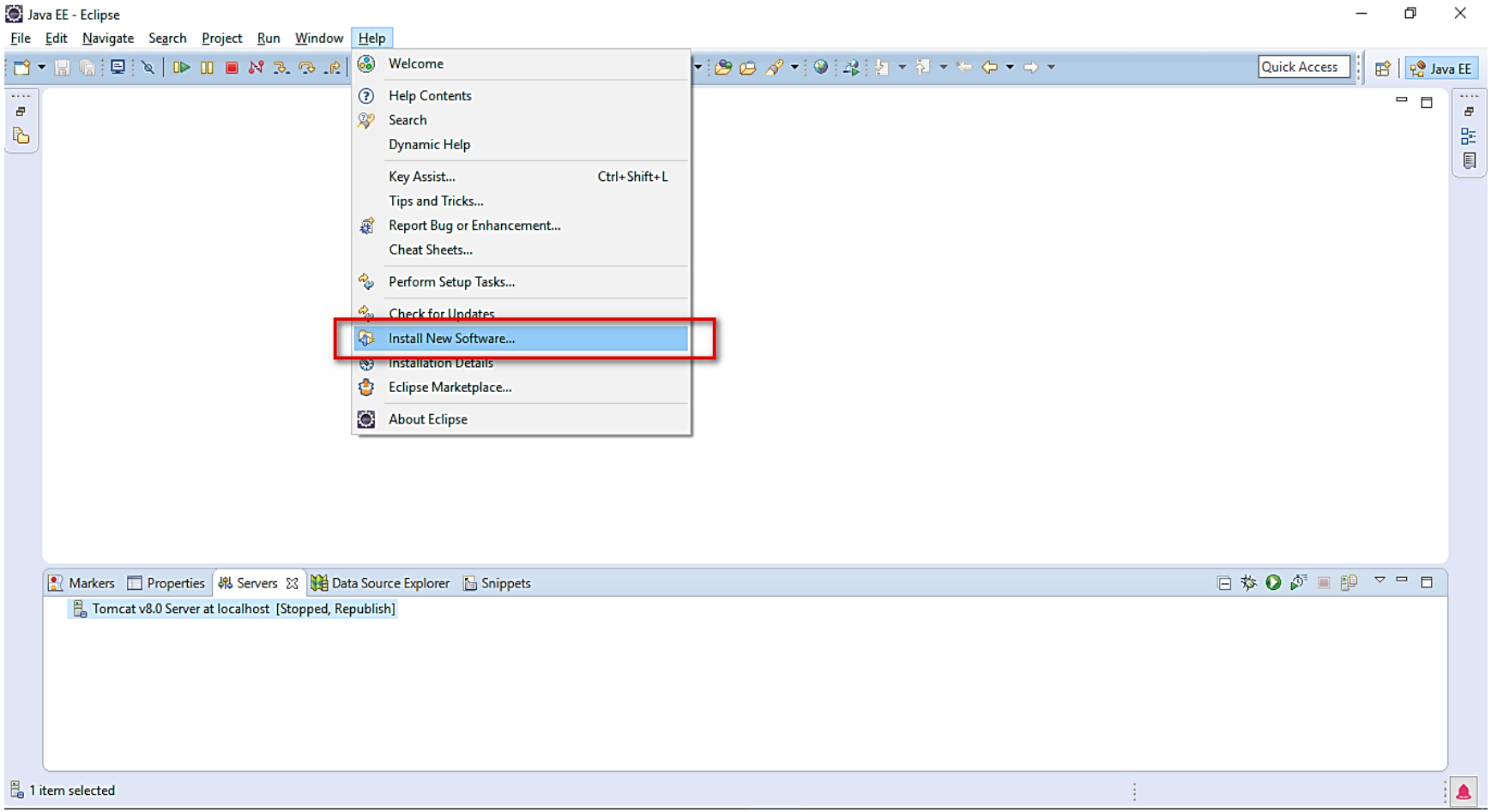
  Vaadin Designer requires Eclipse version 4.4.2 or later.
- Select the features you want to install and click *Next* to proceed with the installation.

Alternatively you can install the Vaadin Plug-in for Eclipse from the Eclipse Marketplace.

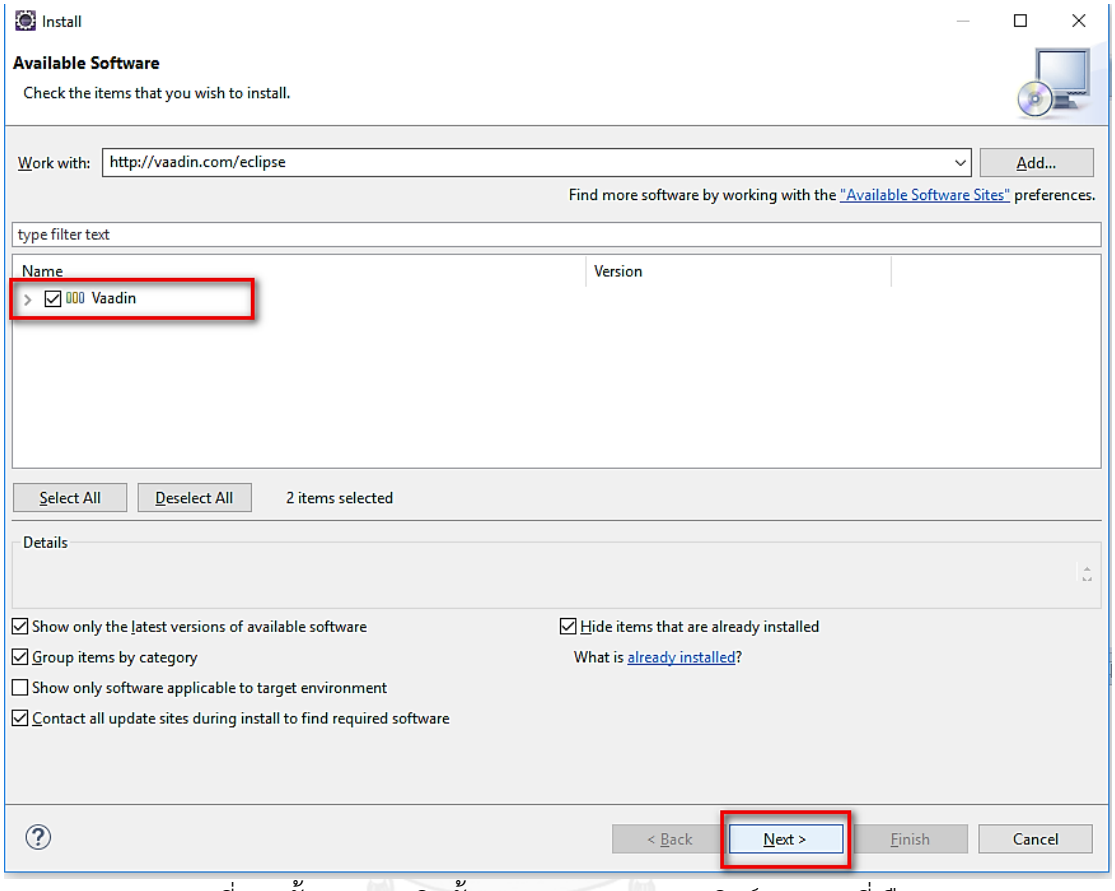
Plug-in Features

รูปที่ 31 ลิงค์การติดตั้ง Vaadin Plugin for Eclipse

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

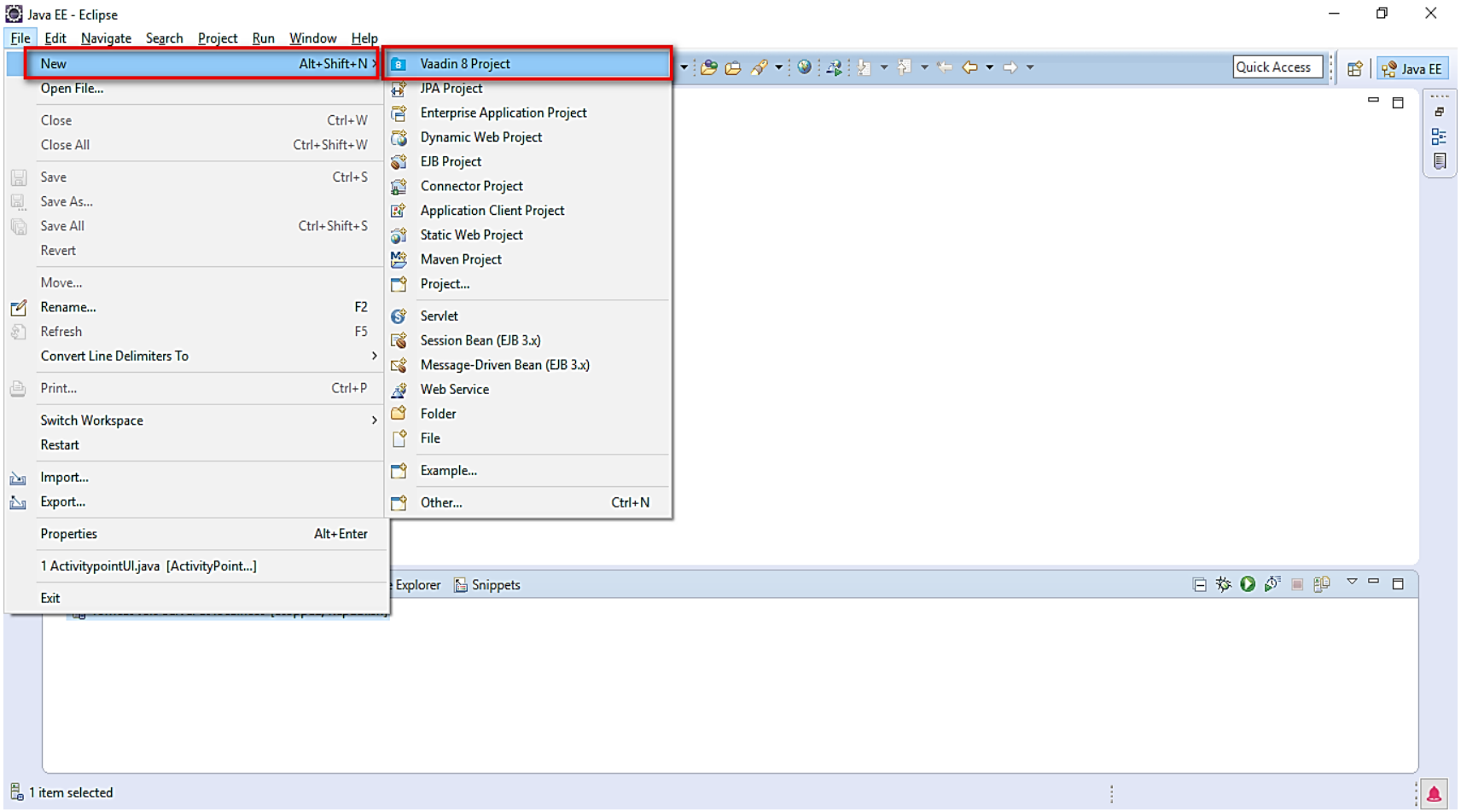


รูปที่ 32 ขั้นตอนการเลือกติดตั้ง Plugin ใน Eclipse



รูปที่ 33 ขั้นตอนการติดตั้ง Vaadin Plugin จากลิงค์ Plugin ที่เลือก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

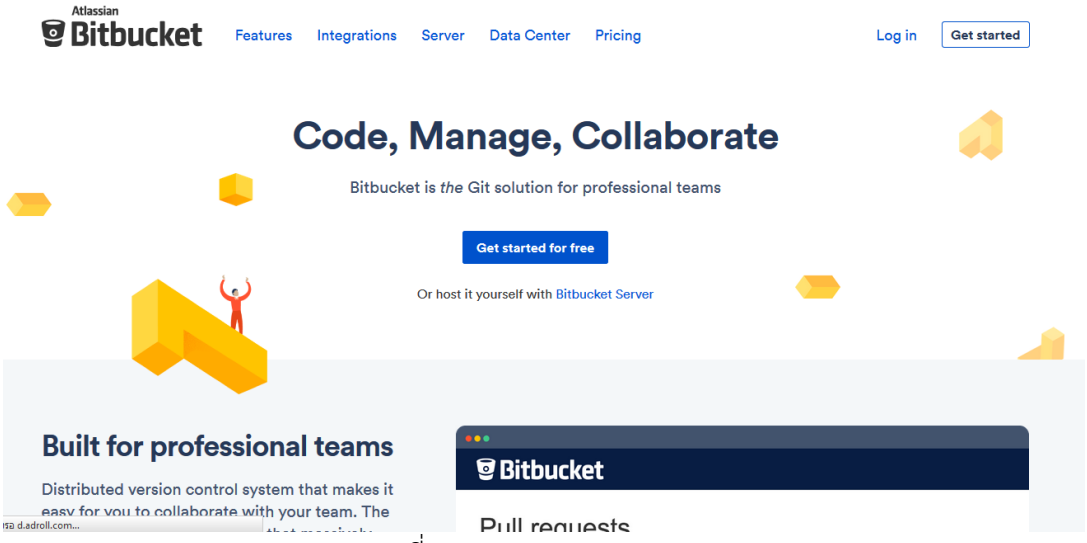


รูปที่ 34 การติดตั้ง Vaadin Plugin ใน Eclipse เสร็จสิ้น

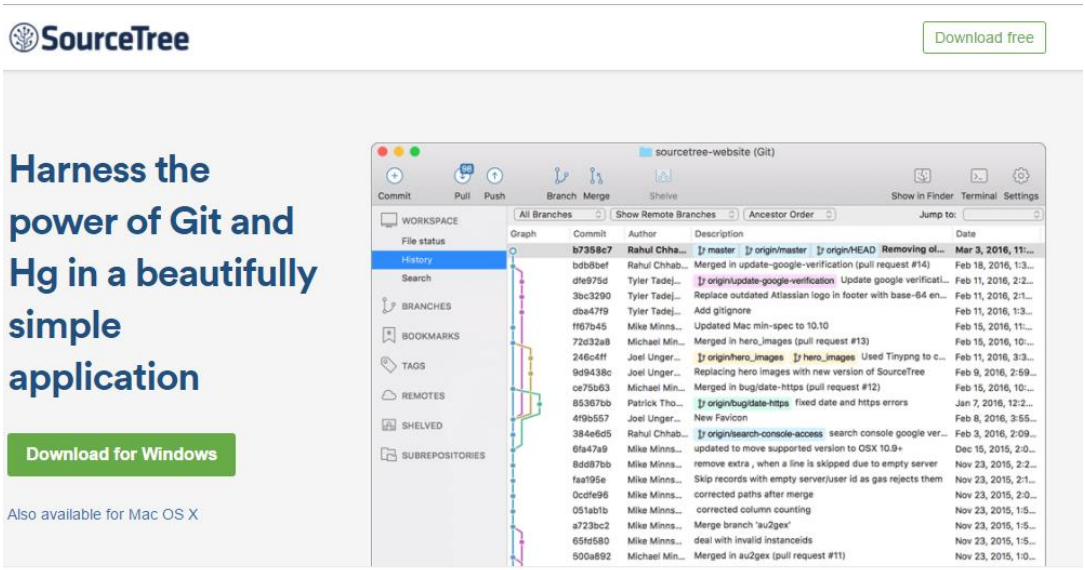
5. วิธีการติดตั้ง SourceTree และการ Clone ชุดรหัสข้อมูลเครื่องมือวิจัย

ชุดรหัสข้อมูลของงานวิจัยได้ใช้ www.bitbucket.org เพื่อทำการจัดการชุดรหัสข้อมูลในการควบคุมเวอร์ชันของการพัฒนาเพิ่มเติมของโครงการเครื่องมืองานวิจัยนี้ ดังแสดงในรูปที่ 36 โดยผู้พัฒนาสามารถใช้เครื่องมือในการจัดการเวอร์ชันของชุดรหัสข้อมูลตามที่ถนัด อาทิเช่น Tortoise SVN หรืออื่น ๆ เป็นต้น แต่ในตัวอย่างทางผู้วิจัยได้ใช้โปรแกรม SourceTree ในการจัดการเวอร์ชันต่าง ๆ ที่พัฒนาขึ้นภายในสภาพแวดล้อมของการพัฒนาไปสู่สภาพแวดล้อมการใช้งานจริง โดยผู้พัฒนาสามารถดาวน์โหลดโปรแกรม SourceTree ได้ที่ลิงค์ https://www.sourcetreeapp.com/ นี้โดยกดปุ่ม Download และติดตั้งในเครื่องของผู้พัฒนาดังอธิบายในรูปที่ 37

เมื่อโปรแกรม SourceTree ติดตั้งเสร็จเรียบร้อย ผู้พัฒนาสามารถ Clone ชุดรหัสข้อมูลที่ทางผู้วิจัยได้พัฒนาขึ้นมาติดตั้งในเครื่องได้ โดยการเปิดโปรแกรม SourceTree โดยเลือกเมนู Clone/New และนำลิงค์ https://axeusonline@bitbucket.org/axeusonline/activity-point.git ไปวางที่ Source Path / Url แล้วทำกด Enter และทำการเลือกตำแหน่งที่ตั้งของโครงการ (โดยผู้วิจัยแนะนำให้ผู้พัฒนาทำการเลือกตำแหน่งที่ตั้งของโครงการเครื่องมืองานวิจัยในโฟลเดอร์ของ Workspace ของโปรแกรม Eclipse เพื่อให้พร้อมพัฒนาได้ทันทีเมื่อเปิดโปรแกรม Eclipse) พร้อมกับกดปุ่ม Clone เพื่อให้โปรแกรม SourceTree ทำการดาวน์โหลดชุดรหัสของเครื่องมือการพัฒนาติดตั้งในเครื่อง ดังอธิบายในรูปที่ 38

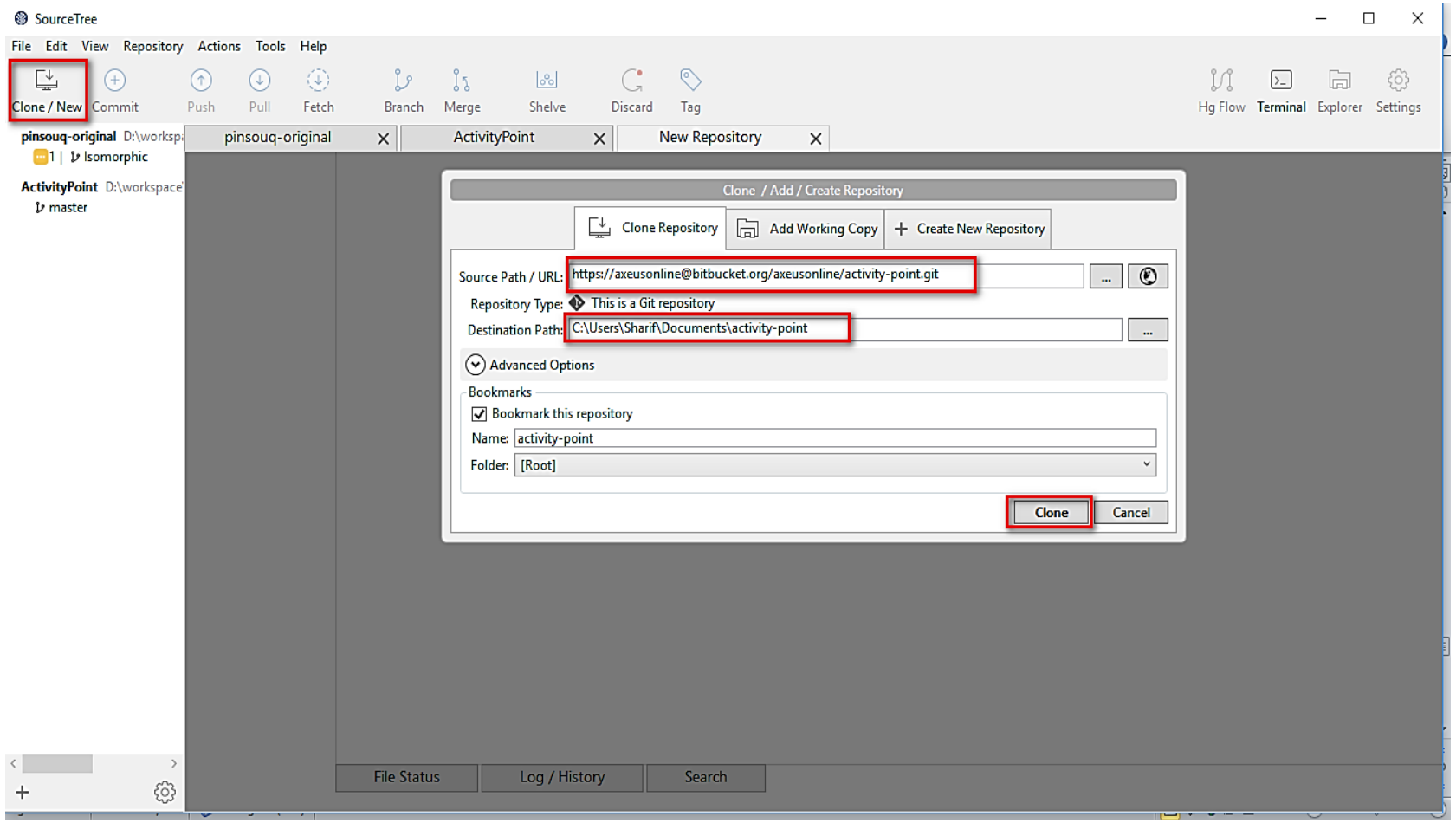


รูปที่ 35 www.bitbucket.org



รูปที่ 36 ลิงค์การติดตั้งโปรแกรม SourceTree โปรแกรมจัดการเวอร์ชันชุดรหัสข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 37 ขั้นตอนการ Clone ชุดรหัสเครื่องมือวิจัยด้วยโปรแกรม SourceTree

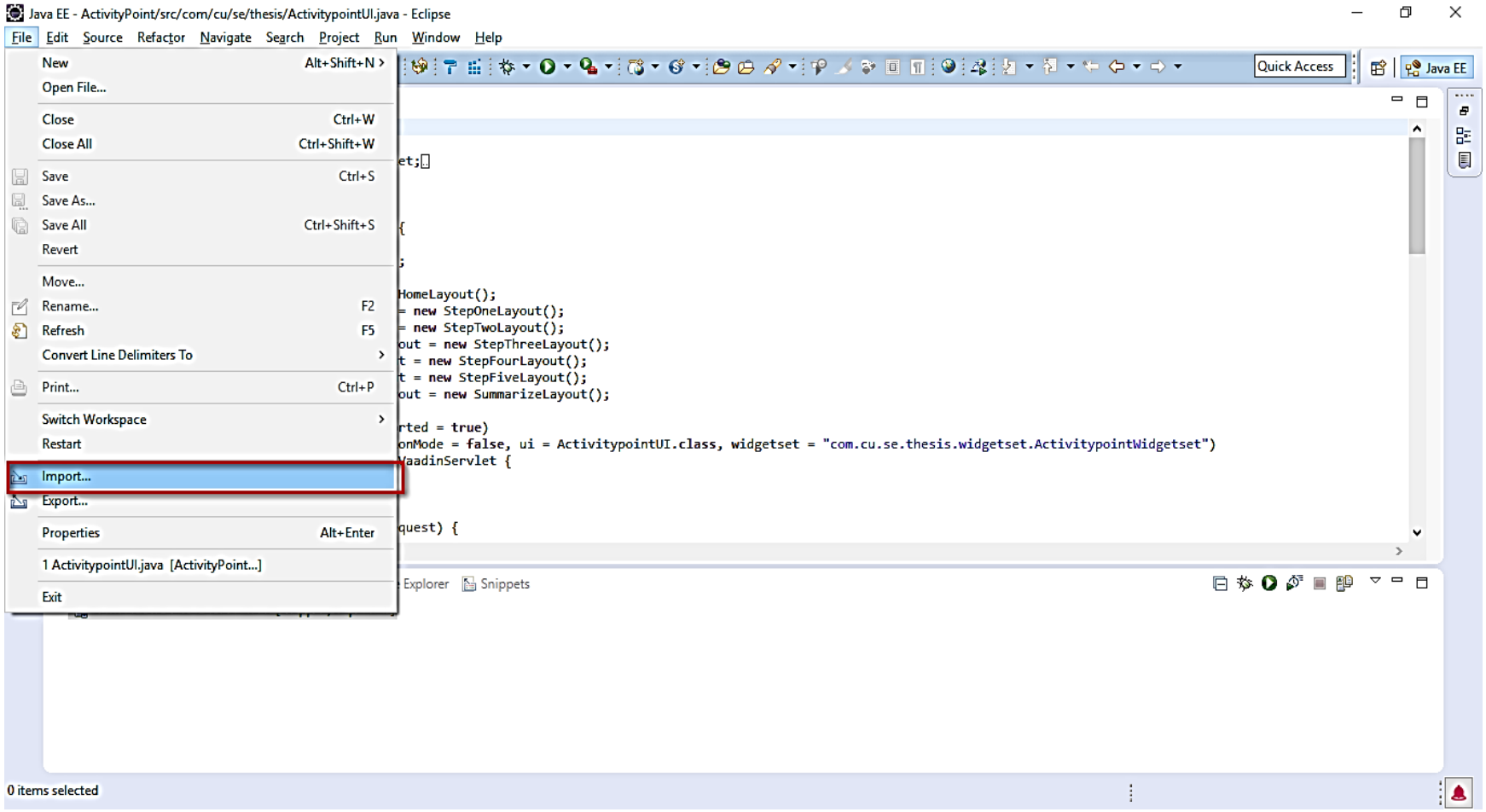
6. วิธีการ Run เครื่องมืองานวิจัย

กรณีเครื่องมือการพัฒนาได้ถูกติดตั้งเรียบร้อย การรันตัวโครงการให้ผู้พัฒนาทำการเปิดโปรแกรม Eclipse หากชุดรหัสโครงการไม่ได้ถูกตั้งอยู่ในโฟลเดอร์ workspace ของ Eclipse ให้ทำการเลือกเมนู File > import > General > Existing Projects into Workspace และทำการเลือกตำแหน่งที่ตั้งของโครงการที่ทำการ Clone ไว้ก่อนหน้าเพื่อนำโครงการนั้นมาอยู่ในโฟลเดอร์ Workspace ของ Eclipse ดังแสดงตัวอย่างในรูปที่ 39 – 41

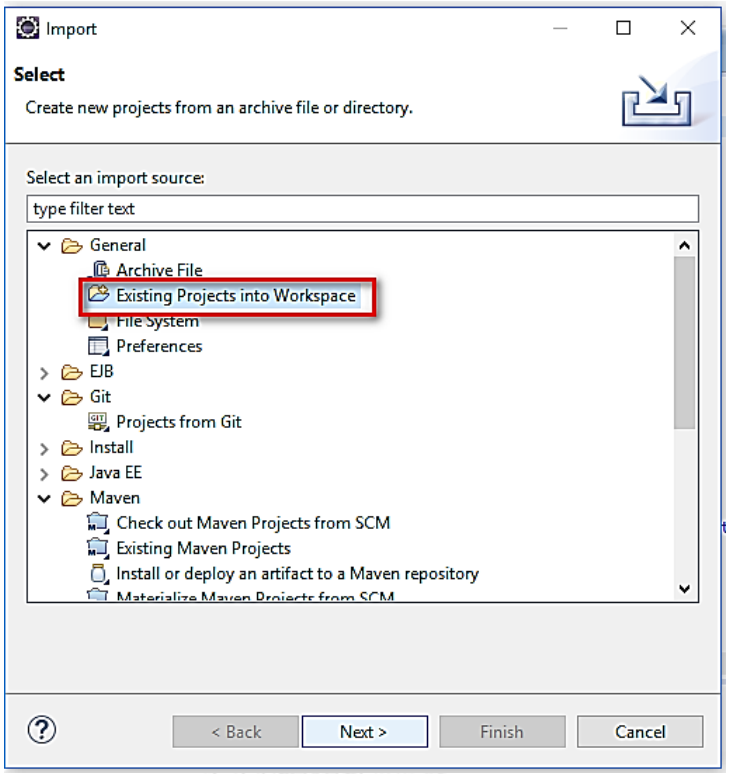
โครงการที่อยู่ภายในโฟลเดอร์ของ Workspace ของโปรแกรม Eclipse ทางผู้พัฒนาให้ทำการเปิดโครงการงานวิจัยที่นำเข้านั้น ๆ โดยให้ขยายตัวโครงการโดยเลือก Java Resources > src และทำการเปิดไฟล์ ActivityPointUI.java ที่อยู่ภายในโฟลเดอร์ src ดังแสดงในรูปที่ 42

ต่อมาให้ผู้พัฒนาทำการเลือกเมนู ดังแสดงในรูปที่ 43 และเลือก Server ที่ได้ติดตั้งก่อนหน้า พร้อมกดปุ่ม Finish เพื่อให้โปรแกรม Eclipse ทำการรันตัวโครงการนั้น ๆ โดย Eclipse จะทำการแสดงตัวโครงการที่รันสำเร็จโดยทำการเปิดแท็บใหม่ภายในโปรแกรม Eclipse เพื่อให้ผู้พัฒนาสามารถลองใช้โปรแกรมผ่านโปรแกรม Eclipse ได้หรือสามารถก็อปปี้ลิงค์ http://localhost:8080/ActivityPoint/servlet/com.cu.se.thesis.ActivitypointUI$Servlet ในแท็บนั้น ๆ เพื่อไปใช้งานในเบราเซอร์ที่ต้องการได้ เพียงเท่านี้ก็สามารถรันตัวโครงการงานวิจัยเพื่อนำมาต่อยอดหรือทำการศึกษาได้ต่อไป ดังแสดงในรูปที่ 44

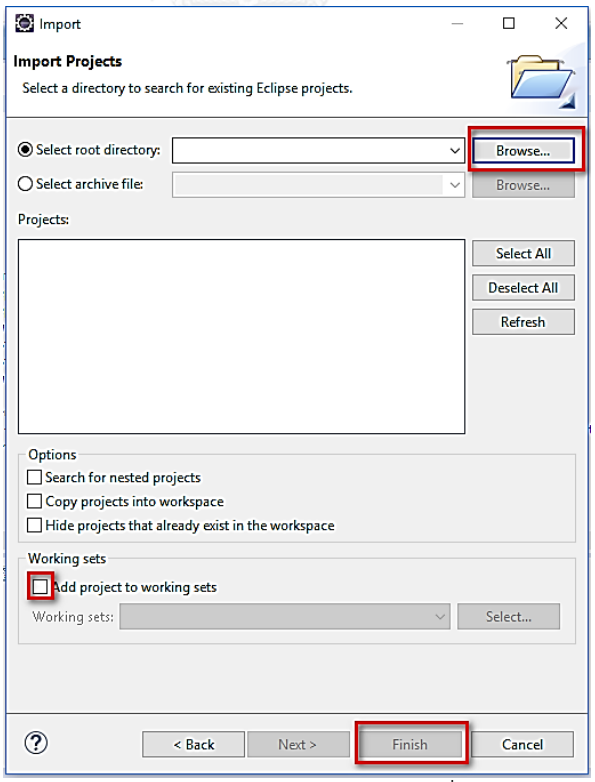
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



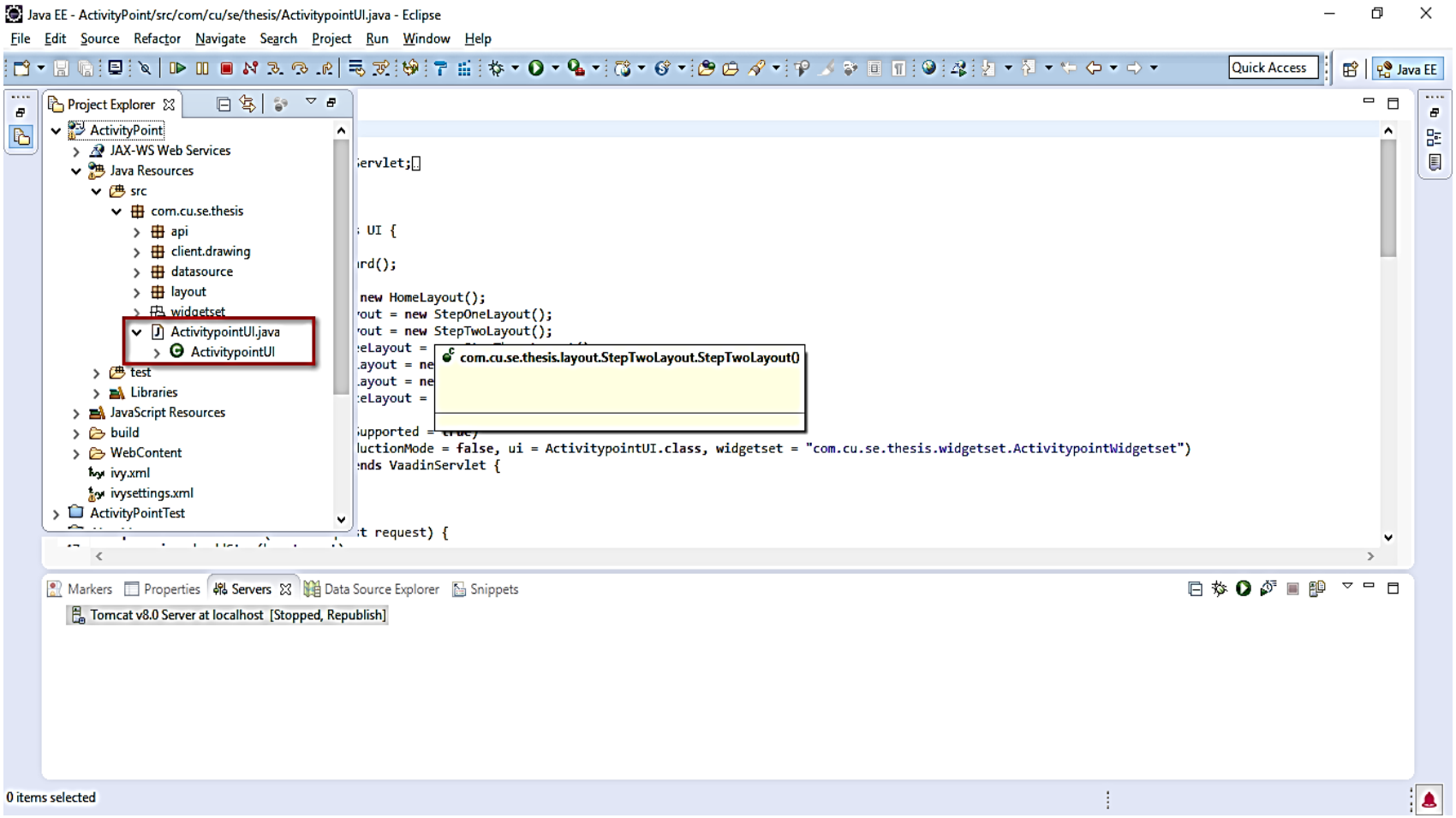
รูปที่ 38 ขั้นตอนการ Import ชุดรหัสโครงการมายังโปรแกรม Eclipse



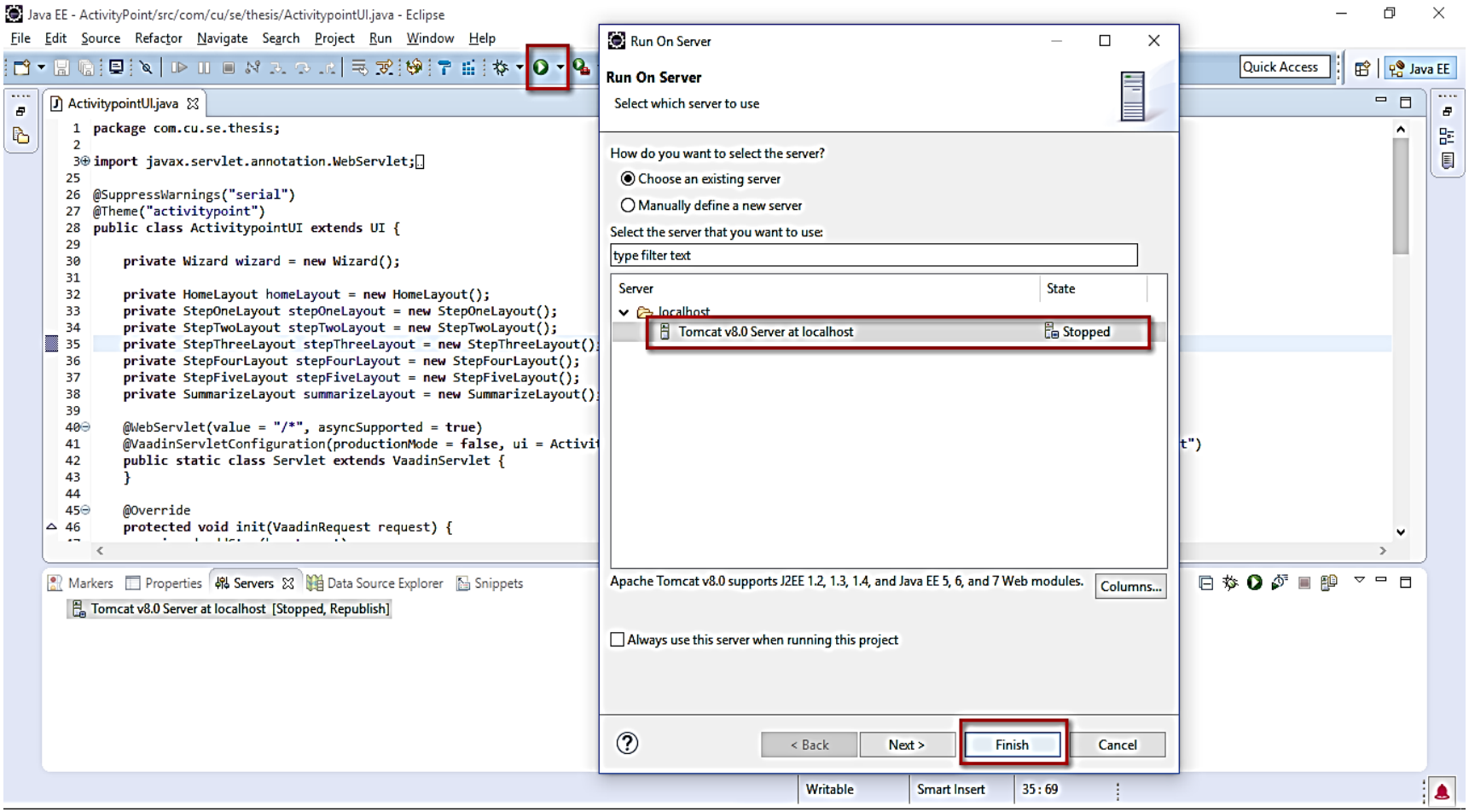
รูปที่ 39 ขั้นตอนการเลือกประเภทชุดรหัสข้อมูลของโครงการที่มีอยู่ในเครื่องผู้พัฒนา



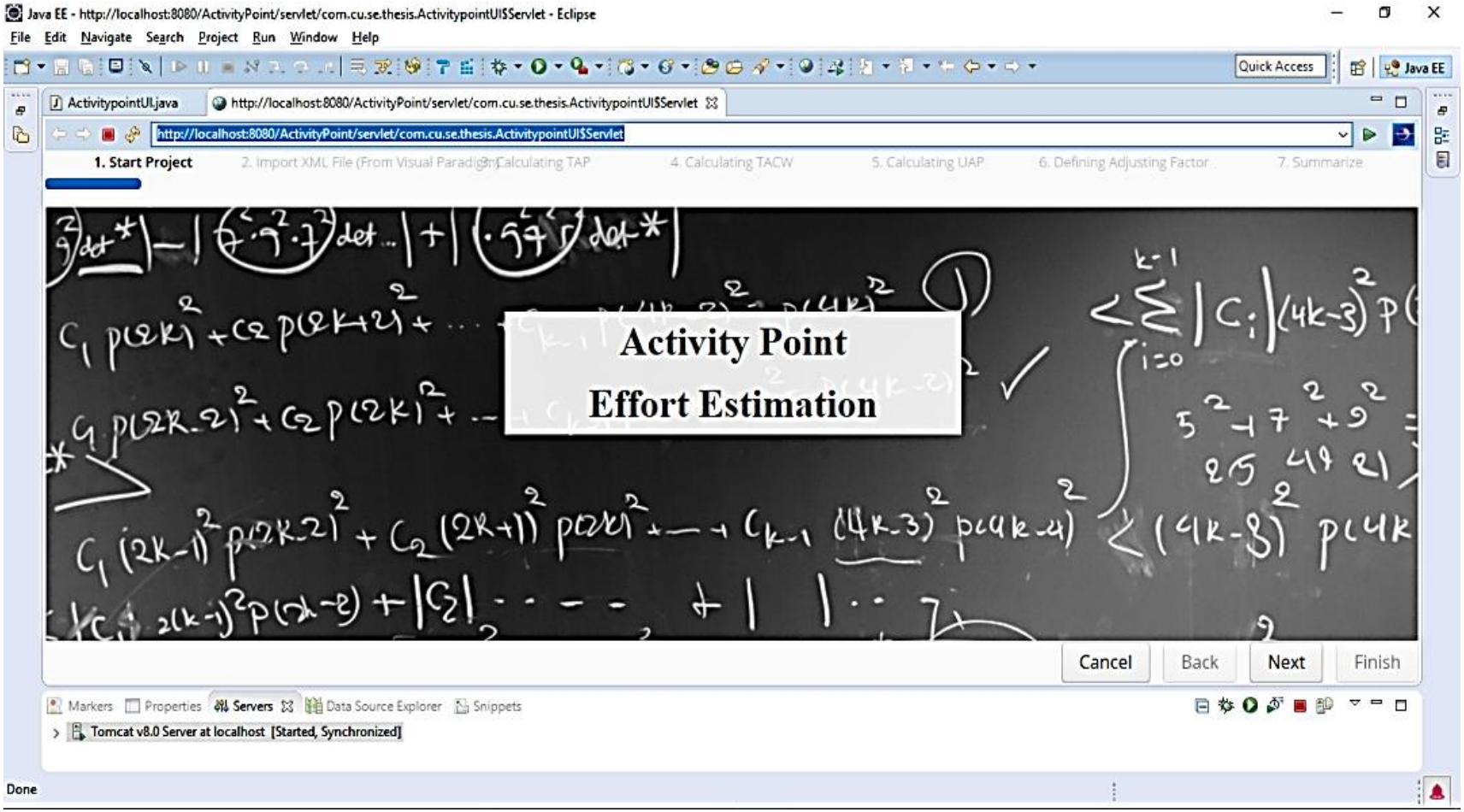
รูปที่ 40 ขั้นตอนการเลือกตำแหน่งของชุดรหัสโครงการเพื่อนำเข้าในโปรแกรม Eclipse



รูปที่ 41 ขั้นตอนการเปิดไฟล์หลักของโครงการวิจัยเพื่อทำการรัน



รูปที่ 42 ขั้นตอนการรันโครงการงานวิจัย



รูปที่ 43 ขั้นตอนผลลัพธ์การรันเครื่องมืองานวิจัย

ภาคผนวก ค
โครงสร้างชุดรหัสข้อมูลเครื่องมืองานวิจัย

ชุดรหัสเครื่องมือวิจัยที่พัฒนาด้วยภาษา Java นั้น ที่นำ Framework Vaadin มาใช้ โดยมีโครงสร้างชุดรหัสข้อมูลที่ประกอบด้วยองค์ประกอบหลัก ๆ ดังต่อไป

1. Folder SRC

โฟลเดอร์ SRC เป็นโฟลเดอร์ที่ใช้ในการเก็บชุดรหัสข้อมูลหลักในรูปแบบไฟล์ชนิด Java โดยแบ่งแพ็คเกจออกเป็น api, client.drawing, layout, widgetset และไฟล์ ActivityPointUI.java

1.1 API เป็นโฟลเดอร์ที่เก็บการทำงานเฉพาะของตัวโครงการเช่น การทำงานในส่วนของการอัพโหลด, การคำนวณเส้นทาง, การวาดแผนภาพกิจกรรมจากข้อมูลนำเข้า XML เป็นต้น

1.2 Client.drawing เป็นโฟลเดอร์ที่เก็บการทำงานที่ใช้ในการวาดแผนภาพกิจกรรมจากข้อมูลนำเข้า XML โดยแบ่งออกเป็น 2 ประเภทคือ การวาดเส้นเชื่อม และการวาดสัญลักษณ์ของแผนภาพกิจกรรม

1.3 Datasource เป็นโฟลเดอร์ที่เก็บข้อมูลคงที่ต่าง ๆ อาทิเช่น ค่าน้ำหนักของแต่ละปัจจัย คำอธิบายของแต่ละระดับของปัจจัยต่าง ๆ เช่น คำอธิบายของระดับปัจจัย TAW ค่าน้ำหนักของ TAW เป็นต้น

1.4 Layout เป็นโฟลเดอร์ที่เก็บหน้าเว็บไซต์ต่าง ๆ ของส่วนเชื่อมต่อประสานในแต่ละการทำงาน เพื่อนำข้อมูลต่าง ๆ ที่ได้แสดงไปยังส่วนเชื่อมต่อประสานไปยังผู้ใช้

1.5 Widgetset เป็นโฟลเดอร์ที่ใช้ในการตั้งค่าของตัวโครงการที่พัฒนาด้วยเฟรมเวิร์ค Vaadin โดยจะมีไฟล์ ActivitypointWidgetset.gwt.xml ที่ใช้ในการระบุค่าต่าง ๆ ที่จะถูกใช้ในโครงการนั้น ๆ

1.6 ActivityPointUI.java เป็นไฟล์หลักที่ใช้ในการเป็นไฟล์เริ่มต้นที่เปรียบเสมือน index.html

2. Folder WebContent

โฟลเดอร์ WebContent เป็นส่วนที่ใช้เก็บไฟล์หลักที่ผ่านการคอมไพล์เสร็จสิ้นโดย ไฟล์ Java ที่อยู่ภายในโฟลเดอร์ SRC จะถูกคอมไพล์ให้อยู่ในรูปของไฟล์ javascript โดยในส่วนนี้จะใช้ในการเก็บไฟล์ css รูปภาพ หรืออื่น ๆ ที่ใช้ในการประกอบหน้าเว็บนั้น ๆ โดยแสดงอยู่ในรูปที่ 45

2.1 META-INF เป็นโฟลเดอร์ที่เก็บข้อมูลไฟล์ทั้งหมดใน jar และเช็คการทำงานของไฟล์ jar ทุกครั้งเมื่อ jar ถูกใช้งาน หากมีสิ่งผิดปกติภายในนั้น META-INF จะทำการระงับการทำงานของไฟล์ทันที

2.2 VAADIN เป็นโฟลเดอร์ที่เก็บไฟล์รูปภาพ css หรือฟอนต์ที่ใช้สำหรับการแสดงหน้าเว็บ โดยจะบรรจุอยู่ภายในโฟลเดอร์ themes/activitypoint/ อีกทั้งโฟลเดอร์ widgetsets จะเป็นโฟลเดอร์ที่เก็บไฟล์จาวาสคริป (javascript) ที่ได้จากการคอมไพล์ไฟล์การทำงานหลักที่อยู่ภายในโฟลเดอร์ SRC

2.3 WEB-INF เป็นโฟลเดอร์ที่ใช้ในการเก็บไฟล์ Jar ต่าง ๆ ภายในโฟลเดอร์ lib ที่ถูกใช้เป็นไลบรารี่เสริมเพื่อทำงานใดงานหนึ่งของตัวโครงการ อีกทั้งยังเก็บไฟล์ web.xml เพื่อใช้ในการตั้งค่าหน้าเว็บว่าต้องการให้หน้าเว็บแสดงผลหน้าแรกบนไฟล์ใด เช่น index.html, index.jsp เป็นต้น

3. Folder Dataset

โฟลเดอร์ dataSet เป็นโฟลเดอร์ที่ใช้ในการเก็บข้อมูลดิบต่าง ๆ ที่ใช้ในการสร้างแบบจำลองในงานวิจัยนี้ ซึ่งประกอบด้วย แผนภาพกิจกรรมของแต่ละโครงการ ข้อมูลดิบที่ใช้ในการหาค่าน้ำหนักของแต่ละปัจจัย รวมถึงไฟล์ชุดรหัสข้อมูลที่ใช้ในการคำนวนหาค่าน้ำหนักที่ใกล้เคียงกับค่าความพยายามในการพัฒนาซอฟต์แวร์จริงที่สุด โดยรายละเอียดประกอบดังนี้

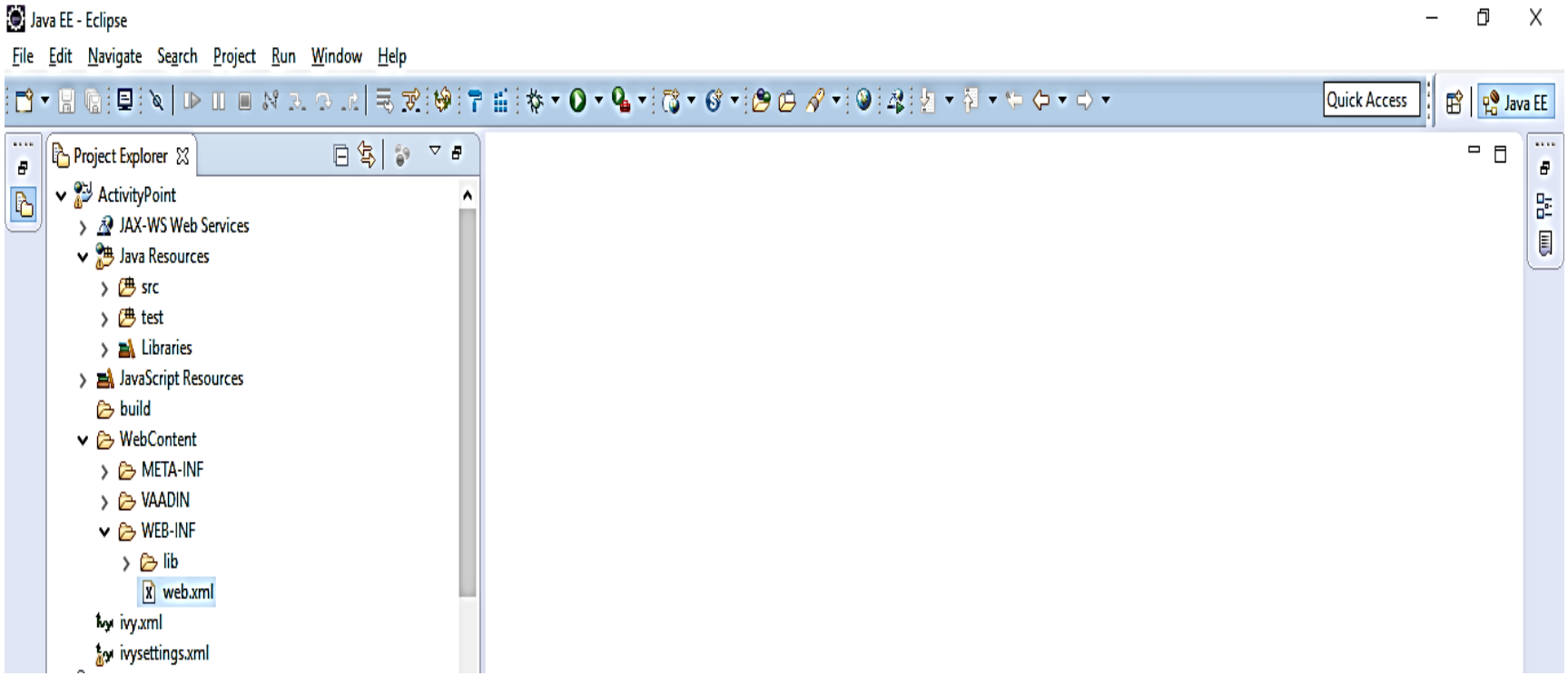
3.1 Evaluation เป็นโฟลเดอร์ที่เก็บไฟล์ Excel ที่ใช้ในการประมาณขนาดและประเมินการประมาณด้วยวิธีการแอคทิวิตีพอยท์และยูสเคสพอยท์

3.**2 Pseudocode** เป็นโฟลเดอร์ที่ใช้ในการเก็บข้อมูลดิบในการคำนวนค่าน้ำหนักที่ใกล้เคียงกับค่าความพยายามในการพัฒนาซอฟต์แวร์จริง โดยประกอบด้วย project.csv และ WeighNumberFinder.java

3.3 SPSS เป็นโฟลเดอร์ที่ใช้ในการเก็บไฟล์ SPSS ในการหาค่า Correlation และแบบจำลองการประมาณ Effort

3.4 ชื่อโครงการ.vpp เป็นไฟล์ Visual Paradigm ที่ประกอบด้วยแผนภาพยูสเคสและแผนภาพกิจกรรมในแต่ละโครงการ

3.5 System.docx เป็นไฟล์ที่ใช้ในการอธิบายถึงรายละเอียดของยูสเคสในแต่ละโครงการ โดยรวมถึงแผนภาพกิจกรรมในมุมมองระดับบน



รูปที่ 44 โครงสร้างไฟล์ของชุดรหัสเครื่องมือวิจัย

ภาคผนวก ง
การใช้งานเครื่องมืองานวิจัย

เครื่องมืองานวิจัยนี้พัฒนาขึ้นเพื่อช่วยผู้ที่ต้องการนำเครื่องมือไปใช้ในการประมาณขนาดและความพยายามจากแผนภาพกิจกรรมที่ได้ออกแบบไว้อัตโนมัติ โดยมีวัตถุประสงค์ในการอำนวยความสะดวกในการประมาณและลดระยะเวลาในการประมาณให้ลดลงไปสู่การนำไปใช้ที่รวดเร็วและมีประสิทธิภาพสูงยิ่งขึ้น ซึ่งการใช้งานจะประกอบด้วยขั้นตอนดังต่อไปนี้

1. นำออกข้อมูลแผนภาพกิจกรรมจาก Visual Paradigm

เบื้องต้น ผู้ใช้งานควรออกแบบแผนภาพกิจกรรมให้ครบถ้วนตั้งแต่มุมมองระดับบนและมุมมองระดับล่าง โดยโปรแกรม Visual Paradigm เป็นโปรแกรมที่ใช้ในการออกแบบแผนภาพ UML ต่าง ๆ โดยผู้ใช้สามารถออกแบบแผนภาพได้หลายแผนภาพในโครงการตัวเดียวกัน ซึ่งเมื่อผู้ใช้งานออกแบบแผนภาพกิจกรรมครบถ้วนแล้วนั้น ให้นำข้อมูลออกโดยการคลิกที่ Export แล้วทำการเลือกแผนภาพกิจกรรมทั้งหมดที่ออกแบบไว้ โดยชนิดไฟล์ที่ได้จากการนำออกข้อมูลจะอยู่ในรูปแบบไฟล์ XML ดังตัวอย่างในรูปที่ 46

2. หน้าแรก

เมื่อทำการเปิดเครื่องมืองานวิจัย ระบบจะเข้ามาในหน้าแรกเพื่อให้ผู้ใช้เริ่มเข้าสู่ขั้นตอนการประมาณขนาด โดยข้างล่างของหน้าเว็บไซต์จะประกอบด้วยปุ่ม Next ซึ่งเป็นปุ่มที่เริ่มเข้าสู่ขั้นตอนของการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์จากเครื่องมืองานวิจัยนี้ โดยทางด้านบนจะระบุถึงขั้นตอนปัจจุบันของการประมาณ และขั้นตอนถัดไปที่ผู้ใช้ต้องทำในการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยเครื่องมือนี้ ดังแสดงตัวอย่างในรูปที่ 47

3. หน้าจำแนกแผนภาพกิจกรรม

เมื่อผู้ใช้ทำการกดปุ่ม Next เพื่อเข้าสู่ขั้นตอนในการประมาณ ระบบจะเริ่มเข้าสู่ขั้นตอนในการจำแนกแผนภาพกิจกรรมโดยเริ่มต้นผู้ใช้จะต้องทำการนำเข้าไฟล์ XML ที่ได้จากการนำออกของโปรแกรม Visual Paradigm ซึ่งเมื่อการอัพโหลดเสร็จสิ้น ระบบจะทำการแสดงแผนภาพกิจกรรมทั้งหมดที่มีในไฟล์ XML นี้ โดยผู้ใช้ต้องทำการเลือกแผนภาพจากรายการที่แสดงว่าแผนภาพใดเป็นแผนภาพกิจกรรมในมุมมองระดับล่าง ก็ให้ทำการดับเบิ้ลคลิกแผนภาพนั้นเพื่อย้ายไปอยู่ในกล่องทางขวา หรือกดปุ่มลูกศร เพื่อย้ายได้เช่นกัน

สำหรับมุมมองระดับบนก็ทำเช่นเดียวกับการจำแนกมุมมองระดับล่าง โดยให้ทำการเลือกแผนภาพจากรายการที่แสดงว่าแผนภาพใดเป็นแผนภาพที่อยู่ในมุมมองของระดับบน

หากผู้ใช้ทำการเลือกแผนภาพมุมมองระดับบน และมุมมองระดับล่างเสร็จสิ้นแล้วก็ให้ทำการกดปุ่ม Next เพื่อไปยังขั้นตอนถัดไป แต่หาผู้ใช้ทำการเลือกแผนภาพผิดก็สามารถยกเลิกการเลือกแผนภาพนั้น ๆ ได้โดยการดับเบิลคลิกในแผนภาพที่แสดงในกล่องทางขวามือหรือทำการกดลูกศร ย้อนกลับได้เช่นกัน โดยตัวอย่างในการใช้งานถูกอธิบายในรูปที่ 48

4. หน้าคำนวณ TAW, TPW และ TAP

ต่อมาเมื่อผู้ใช้ทำการไปยังขั้นตอนการคำนวณ TAP โดยระบบจะทำการนับจำนวนของจำนวนกิจกรรมและคำนวณเส้นทางทั้งหมดของแต่ละแผนภาพเพื่อทำการคำนวณหาค่า TAW และ TPW จากการนำจำนวนกิจกรรมและเส้นทางที่ได้มาทำการเปรียบเทียบกับเงื่อนไขของค่าน้ำหนักว่าอยู่ในช่วงใดเพื่อทำการถ่วงน้ำหนักและนำผลรวมจากค่าที่ถ่วงน้ำหนักที่ได้มารวมเป็นค่า TAP โดยในหน้านี้จะแสดงถึงจำนวนของกิจกรรมที่ระบบนับได้ในแต่ละแผนภาพ (Number of Activity) ค่าน้ำหนักของกิจกรรมที่อยู่ในช่วงที่กำหนด (NA Level Weight) ค่าผลรวมของน้ำหนักกิจกรรม (TAW) ค่าจำนวนของเส้นทาง (Number of Path) ค่าน้ำหนักของเส้นทางที่อยู่ในช่วงที่กำหนด (NP Level Weight) และ ค่าผลรวมน้ำหนักเส้นทาง (TPW)

ค่า TAW และ TPW ที่ได้จากการนับข้างต้นจะถูกนำมารวมกันเพื่อแสดงถึงค่าของผลรวมของกิจกรรม (TAP) ของแต่ละกิจกรรมในมุมมองระดับล่าง โดยตัวอย่างของการคำนวณนี้จะถูกแสดงดังรูปที่ 49

Chulalongkorn University

5. หน้าคำนวณ McCabe Complexity และ TACW

เมื่อผู้ใช้ทราบถึงค่า TAP ในแต่ละแผนภาพกิจกรรมแล้ว ผู้ใช้ทำการกดปุ่ม Next เพื่อไปยังขั้นตอนของการคำนวณค่า McCabe Complexity และ TACW ในส่วนของแผนภาพกิจกรรมในมุมมองระดับบน โดยค่าของ McCabe Complexity ที่ได้จะถูกนำมาเทียบกับช่วงของความซับซ้อนเพื่อทำการกำหนดค่าน้ำหนักของความซับซ้อนนั้น (AC Level Weight)

เมื่อระบบได้ค่าของ McCabe Complexity และ ค่าน้ำหนักของความซับซ้อน ระบบจะนำสองค่าที่ได้มาทำการคำนวณค่า TACW เพื่อแสดงถึงแทนค่าสำหรับแผนภาพกิจกรรมในมุมมองระดับบนต่อไป โดยตัวอย่างของการคำนวณในหน้านี้ ถูกอธิบายดังรูปที่ 50

6. หน้าคำนวณ UAP

เมื่อผู้ใช้ทำการคำนวณในขั้นตอนของแผนภาพกิจกรรมในมุมมองระดับล่าง และ แผนภาพกิจกรรมในมุมมองระดับบนเสร็จสิ้น ระบบก็จะเข้าสู่ขั้นตอนการคำนวณค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง (UAP) โดยระบบจะแสดงถึงค่าของ TAP ที่แสดงถึงตัวแทนของแผนภาพกิจกรรมในมุมมองระดับล่าง และค่าของ TACW ที่แสดงถึงตัวแทนของแผนภาพกิจกรรมในมุมมองระดับบน พร้อมกับค่าของขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง เพื่อไปยังขั้นตอนการกำหนดปัจจัยในการปรับค่าต่อไป โดยตัวอย่างของหน้าคำนวณ UAP ถูกอธิบายดังรูปที่ 51

7. หน้ากำหนดปัจจัยในการปรับค่า TCF, ECF และ PRF

เมื่อระบบคำนวณค่าของขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุงเสร็จสิ้น ก็จะเข้าสู่ขั้นตอนการกำหนดค่าปัจจัยในการปรับค่า โดยระบบจะแสดงรายละเอียดของปัจจัยในการปรับค่าทั้งหมด ที่ประกอบด้วย ค่าปัจจัยความซับซ้อนทางเทคนิค (TCF) ค่าปัจจัยความซับซ้อนทางสภาพแวดล้อม (ECF) และค่าปัจจัยความเสี่ยงของมนุษย์ (PRF)

ข้างล่างรายละเอียดปัจจัยในการปรับค่า ระบบจะแสดงแบบฟอร์มให้ผู้ใช้สามารถกำหนดคะแนนผลกระทบของแต่ละปัจจัย โดยมีค่าเริ่มต้นที่ 0 ซึ่งหมายถึงปัจจัยดังกล่าวไม่มีผลกระทบต่อขนาด ไปจนถึงค่าผลกระทบเท่ากับ 5 ที่แสดงถึงปัจจัยดังกล่าวมีผลกระทบสูงสุดต่อการเปลี่ยนแปลงของขนาดของซอฟต์แวร์ ซึ่งผู้ใช้งานต้องทำการกำหนดคะแนนในทุกปัจจัยที่ถูกระบุไว้ ซึ่งหากระบุคะแนนไม่ครบทุกปัจจัย ระบบจะไม่ทำการอนุญาตให้ไปยังขั้นตอนถัดไป เนื่องจากผลของการไม่เลือกปัจจัยบางปัจจัยนั้น ส่งผลต่อขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ จึงทำให้ค่าที่ได้อาจจะมีความคลาดเคลื่อนที่สูงขึ้น โดยตัวอย่างของการคำนวณค่าปัจจัยในการปรับปรุงขนาดจะถูกแสดงดังรูปที่ 52

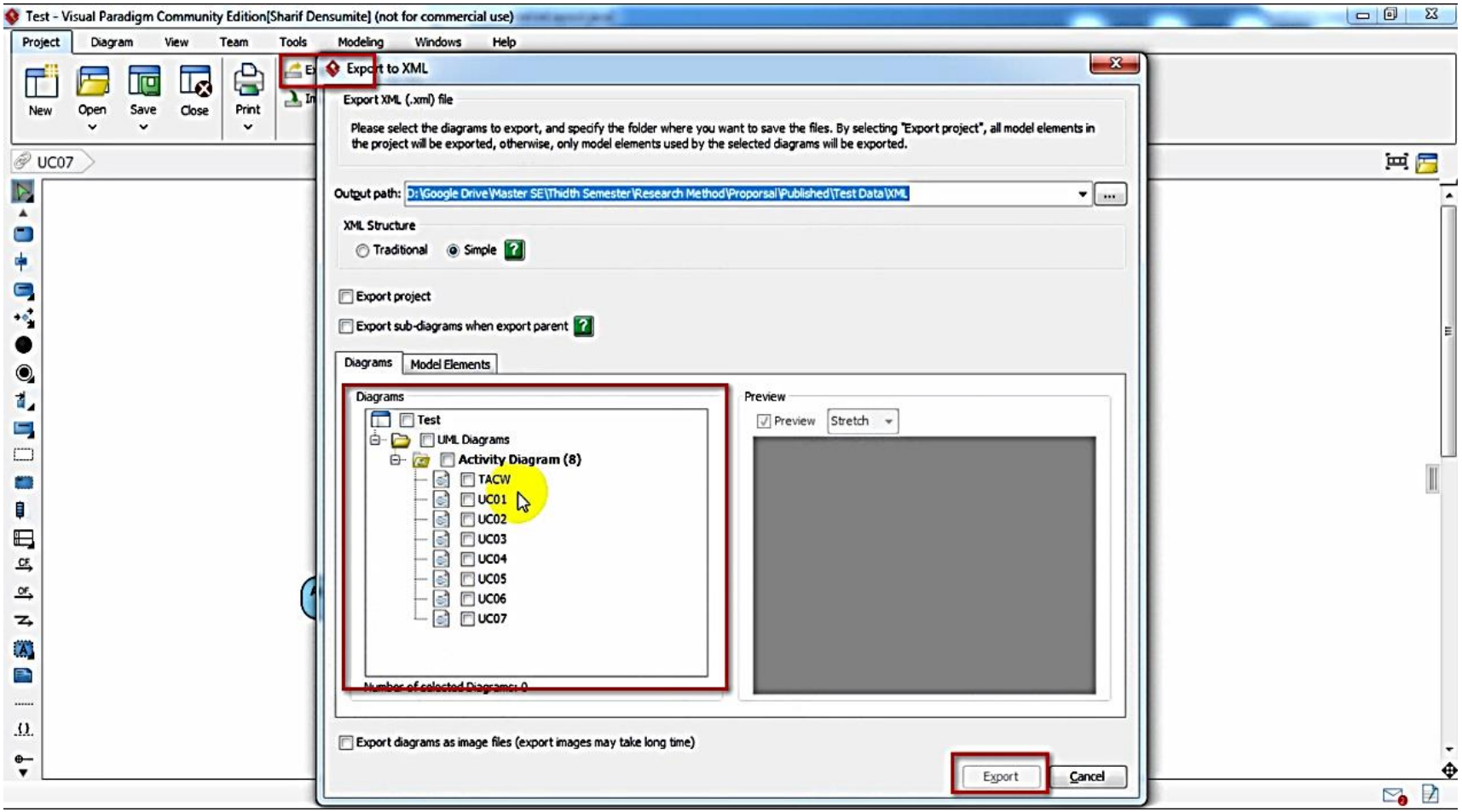
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

8. หน้าคำนวณ AP และ Effort

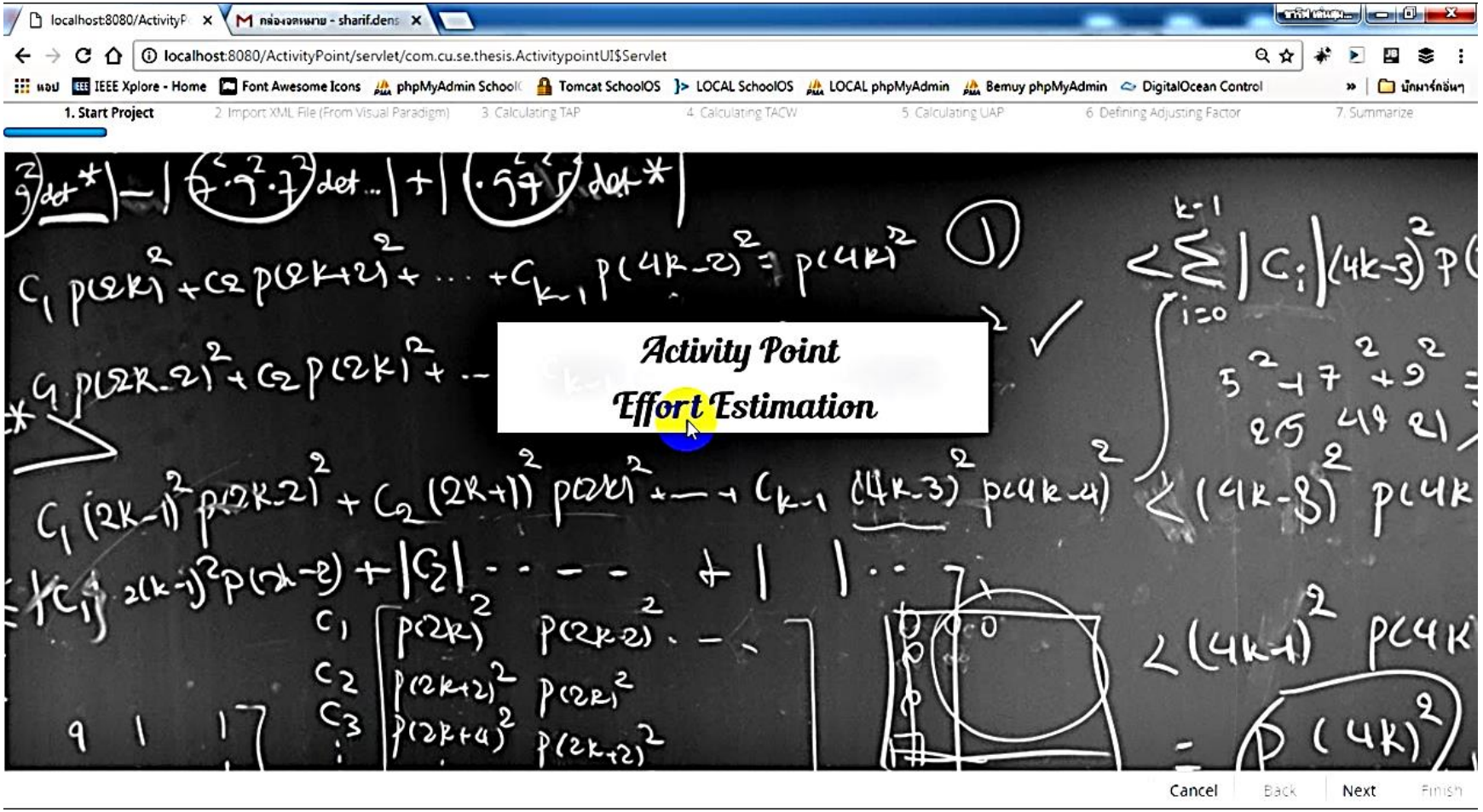
ในขั้นตอนสุดท้ายของการประมาณขนาด ระบบจะนำค่าขนาดของแอคทิวิตีพอยท์ก่อนการปรับปรุง ค่าปัจจัยที่ใช้ในการปรับปรุง เพื่อทำการคำนวณค่าของขนาดซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์

ค่าของขนาดของซอฟต์แวร์ในหน่วยของแอคทิวิตีพอยท์จะถูกนำมาใช้ในการประมาณความพยายามในการพัฒนาซอฟต์แวร์ที่ได้จากการนำค่าของแอคทิวิตีพอยท์ที่ได้จากชุดข้อมูลทดลอง (Training Data Sets) มาใช้ในการสร้างแบบจำลองการประมาณความพยายามการพัฒนาซอฟต์แวร์ด้วยวิธีสมการถดถอย (Regression Model) โดยใช้ค่าความพยายามในการพัฒนาซอฟต์แวร์จริงในการแสดงถึงตัวแปรอิสระในการสร้างแบบจำลองนี้ โดยตัวอย่างของการคำนวณค่าขนาดของซอฟต์แวร์ในหน่วยแอคทิวิตีพอยท์ และความพยายามในการพัฒนาซอฟต์แวร์ ดังแสดงในรูปที่ 53

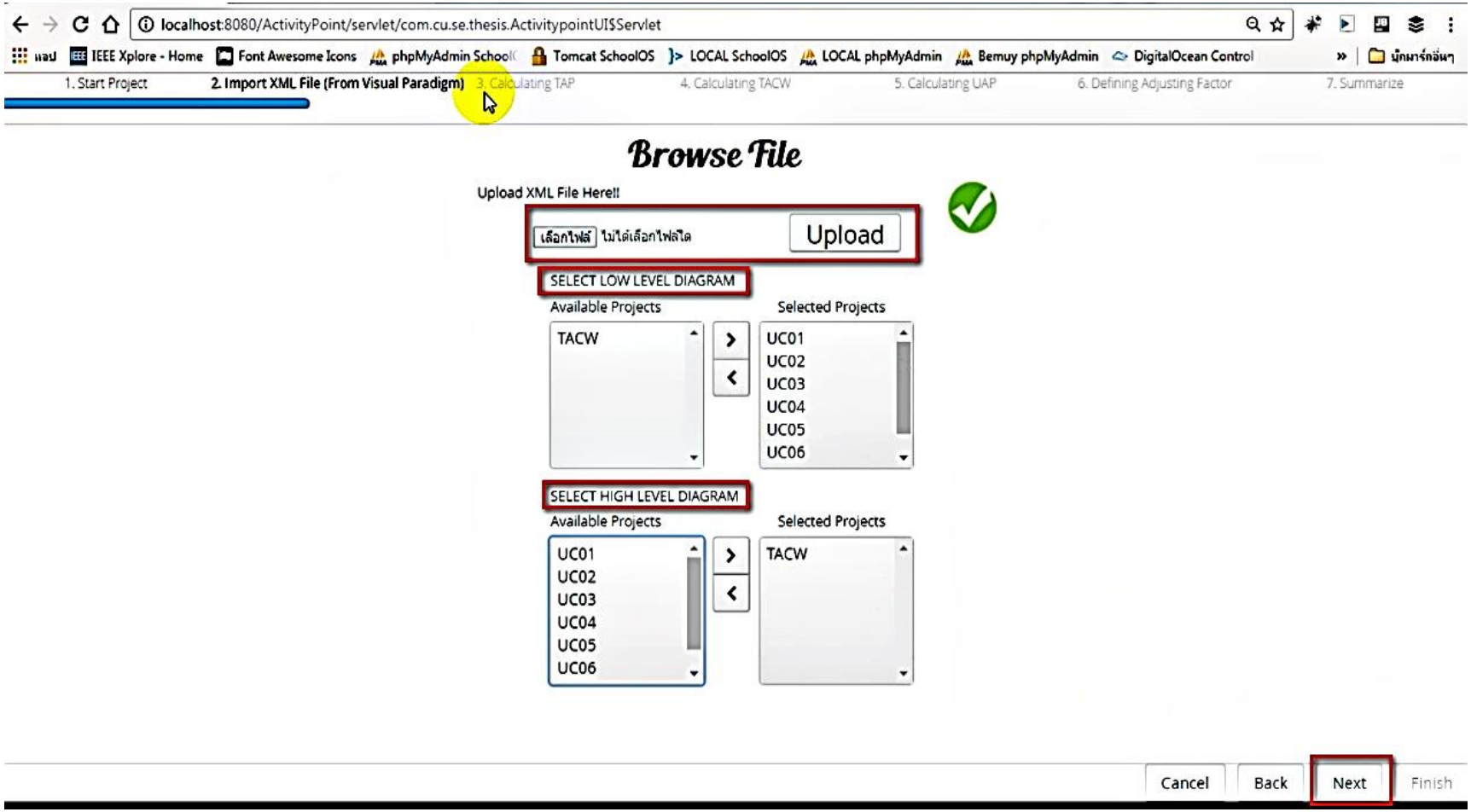
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



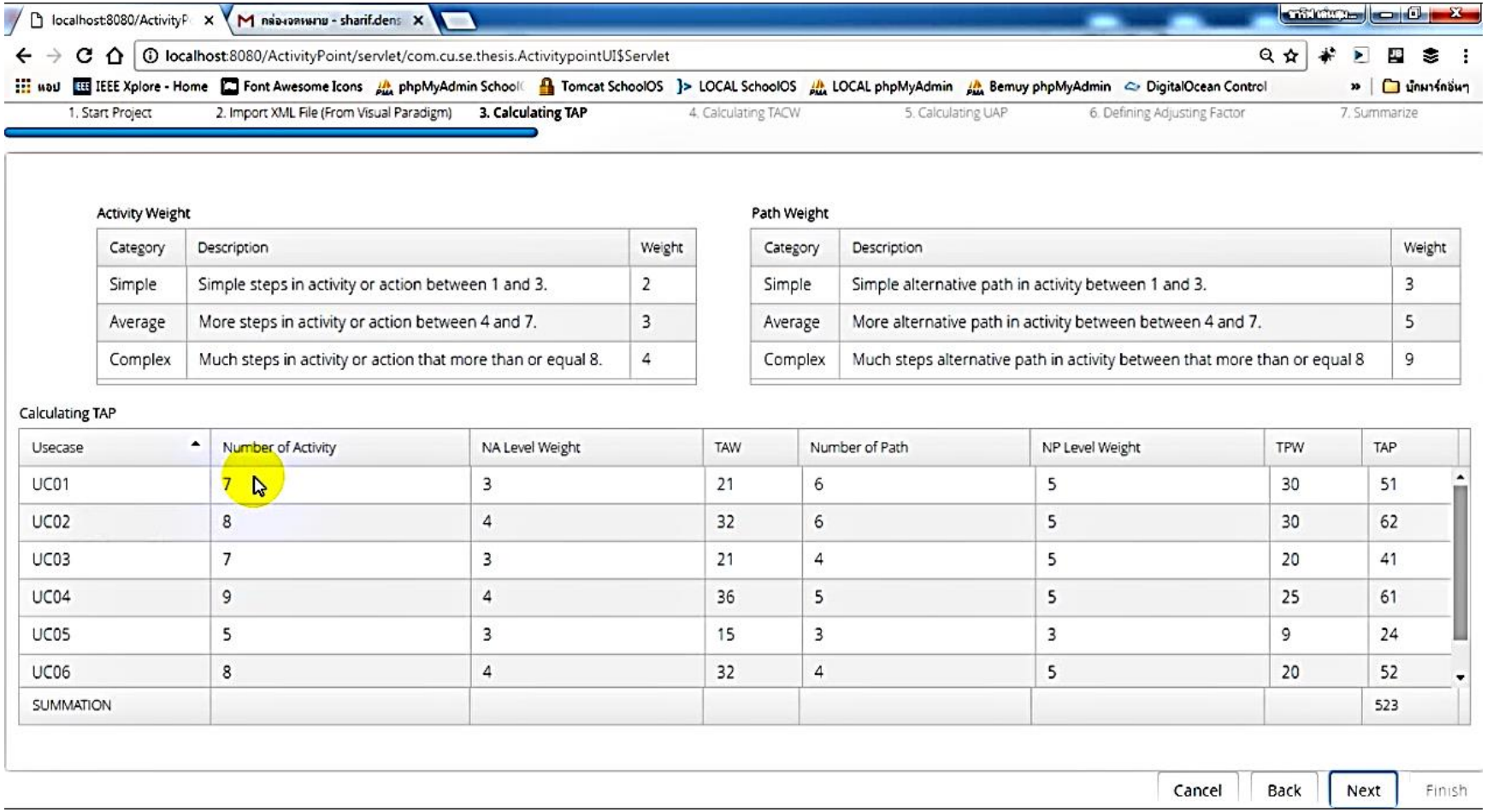
รูปที่ 45 การนำข้อมูลแผนภาพกิจกรรมออกในโปรแกรม Visual Paradigm



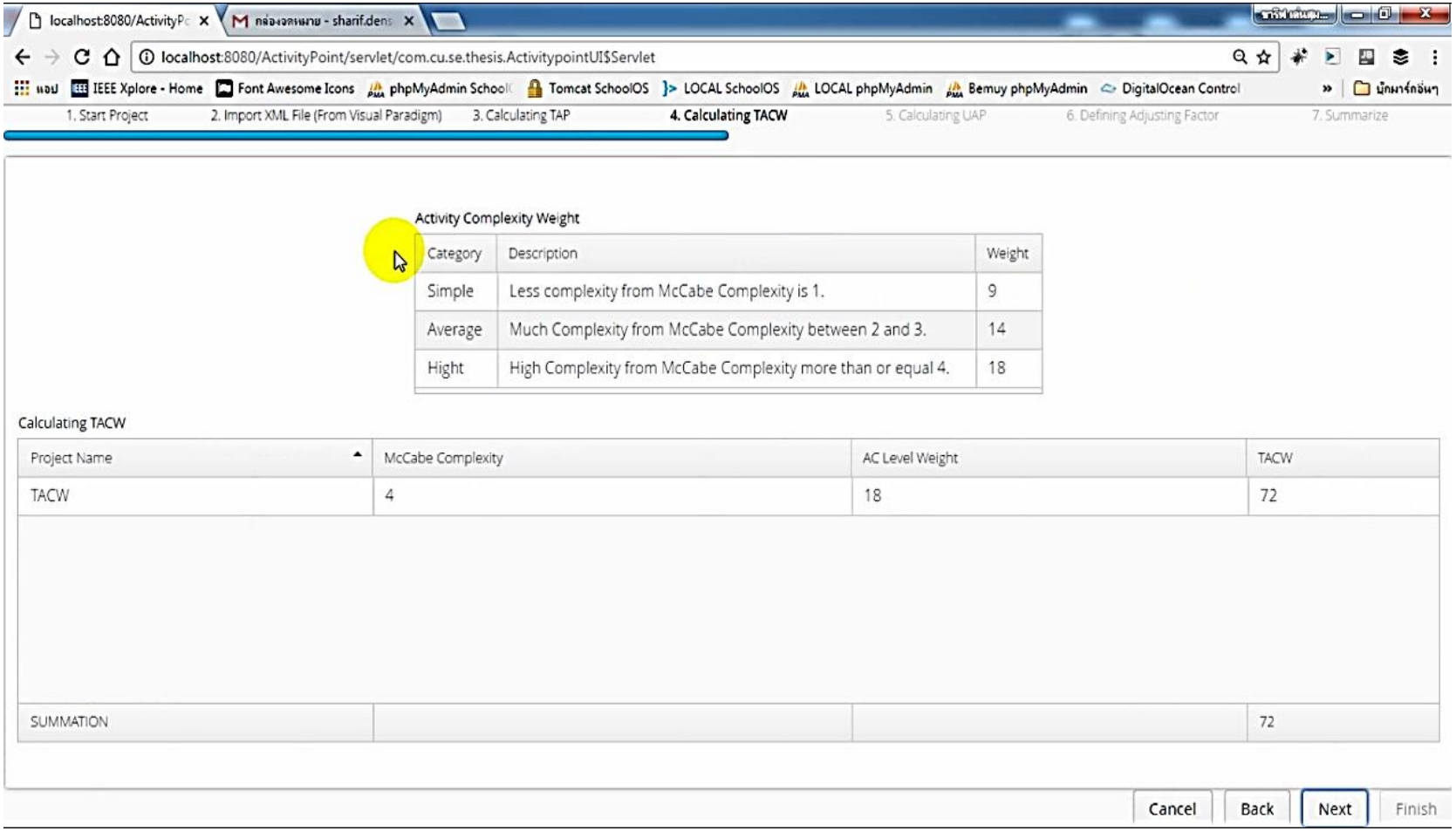
รูปที่ 46 หน้าแรกของเครื่องมือการประมาณขนาดด้วยวิธีแอคทิวิตีพอยท์



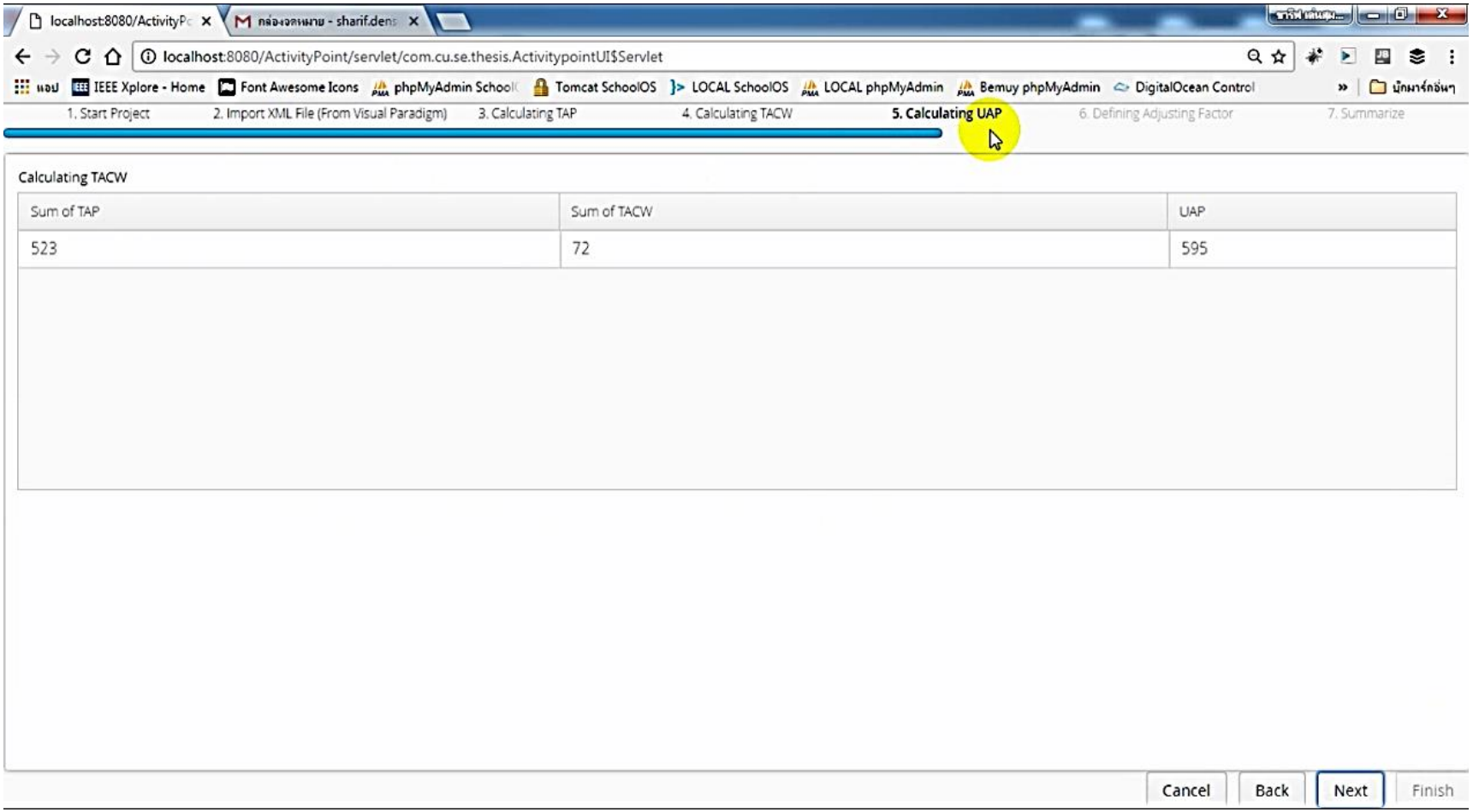
รูปที่ 47 ขั้นตอนการใช้งานในส่วนของการจำแนกแผนภาพกิจกรรม



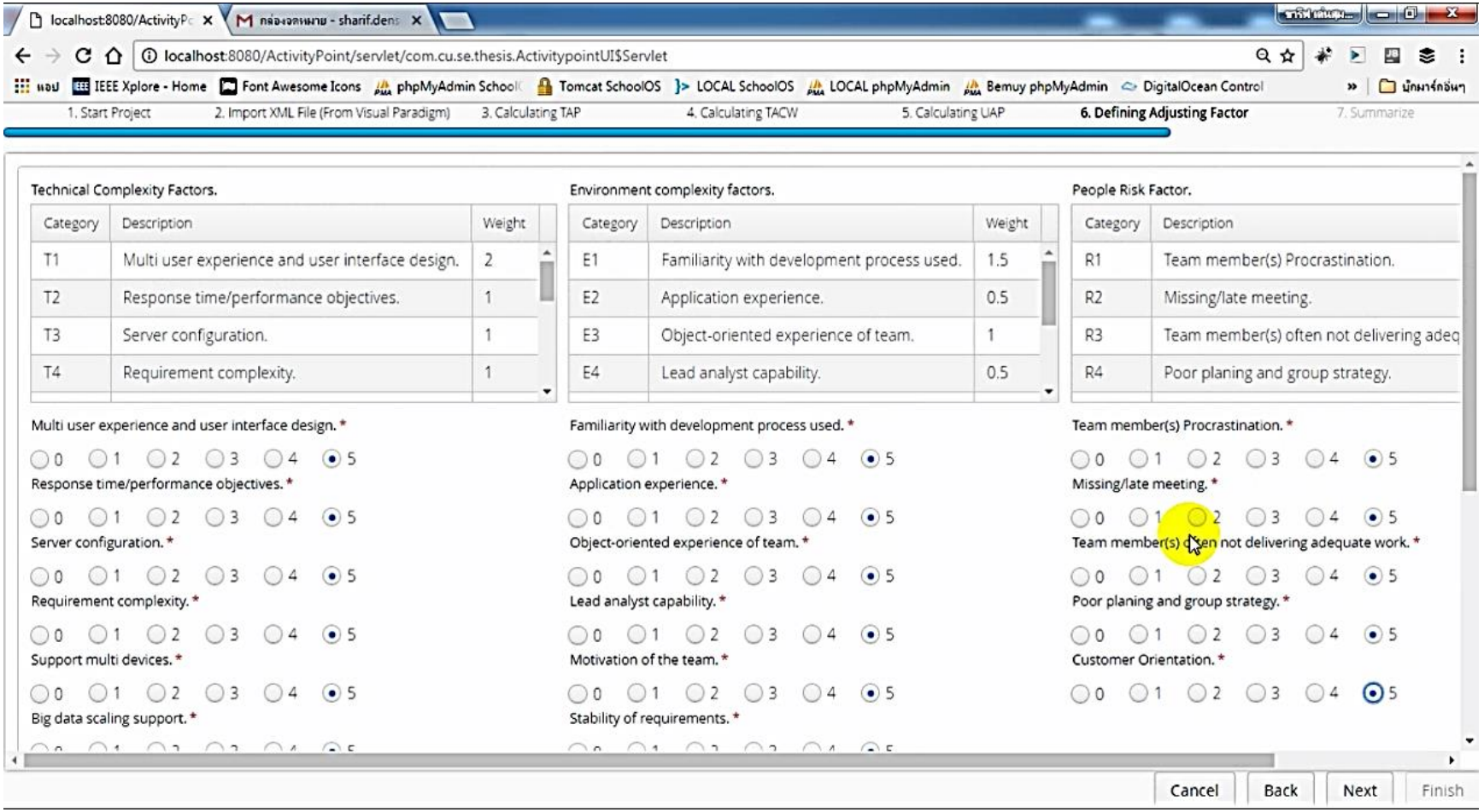
รูปที่ 48 ขั้นตอนหน้าการคำนวณค่า TAW TPW และ TAP



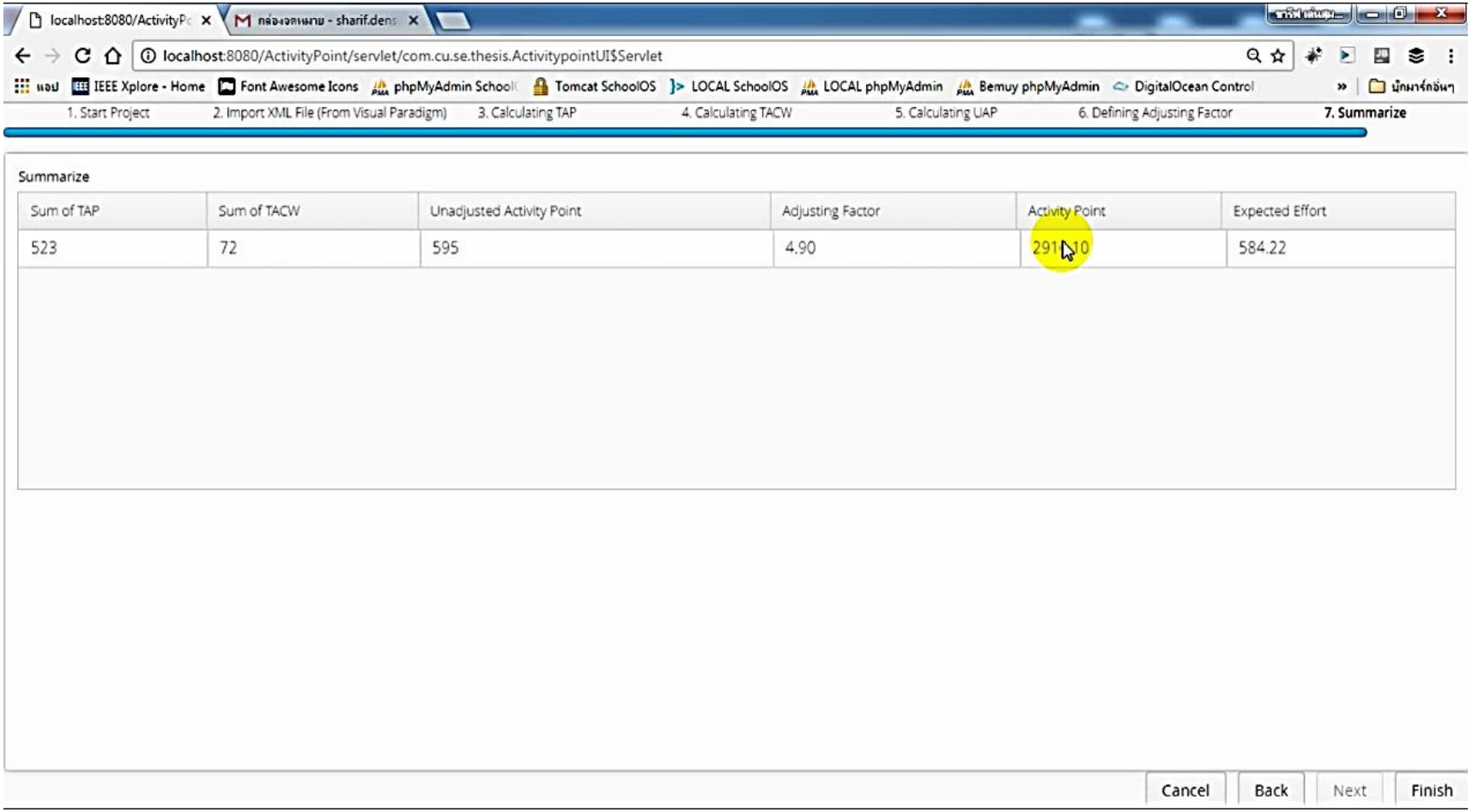
รูปที่ 49 ขั้นตอนการทำงานหน้าการคำนวณค่า TACW



รูปที่ 50 ขั้นตอนการใช้งานหน้าการคำนวณค่า UAP



รูปที่ 51 ขั้นตอนการทำงานหน้าการกำหนดค่าของปัจจัยในการปรับค่า



รูปที่ 52 ขั้นตอนการคำนวณค่าขนาดแอคทิวิตีพอยท์ และความพยายามในการพัฒนาซอฟต์แวร์

ภาคผนวก จ

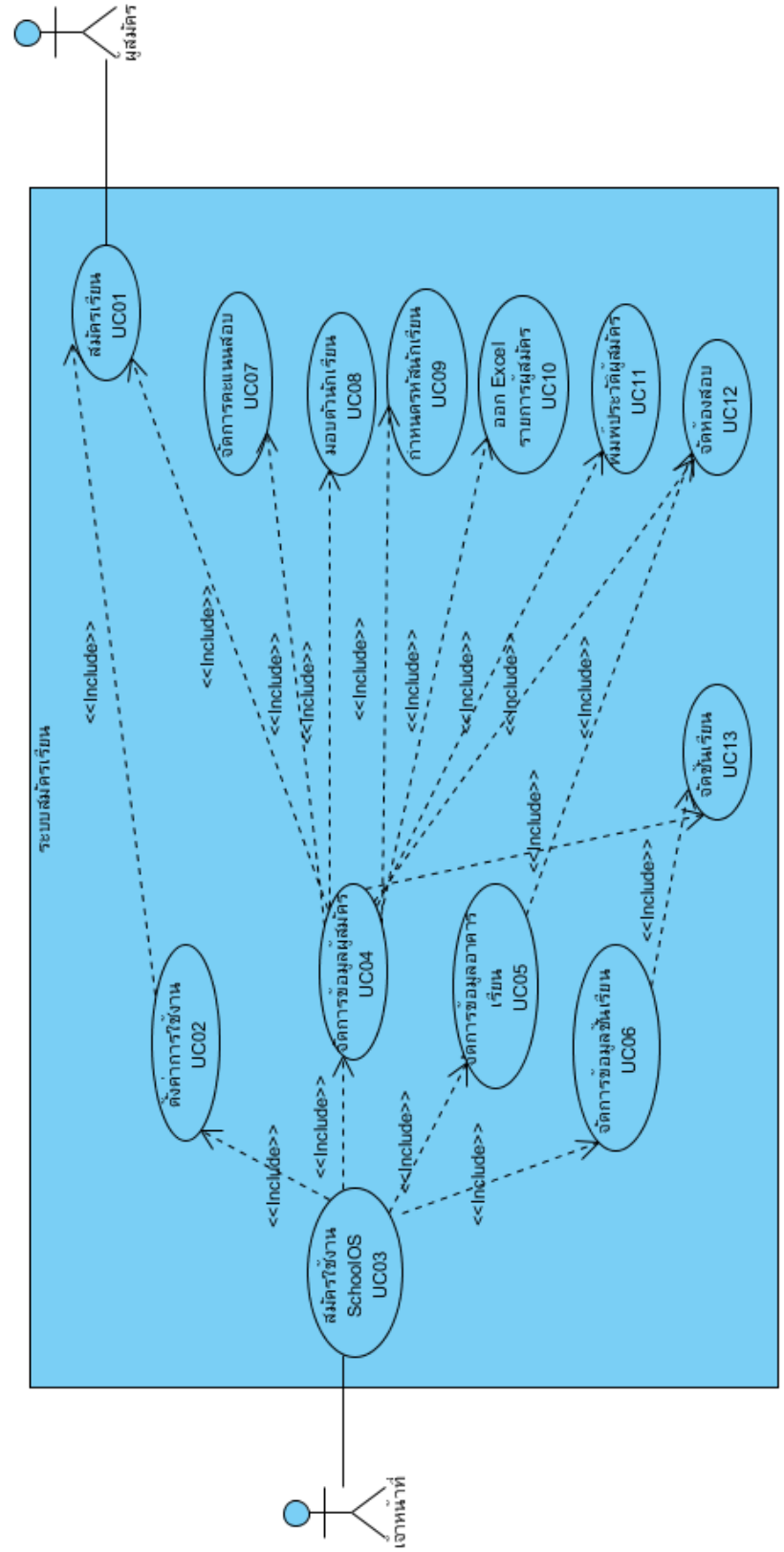
ตัวอย่างการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์

ตัวอย่างการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีแอคทิวิตีพอยท์นี้ ระบบการสมัครเรียนซึ่งเป็นหนึ่งในโครงการที่ถูกใช้ในการประเมินแบบจำลองนี้ในบทที่ 6 โดยมีแผนภาพยูสเคสของระบบดังรูปที่ 54 แผนภาพกิจกรรมในมุมมองระดับบนดังรูปที่ 55 และแผนภาพควบคุมกระแสที่ได้จากการแปลงจากแผนภาพกิจกรรมในมุมมองระดับบนในรูปที่ 56

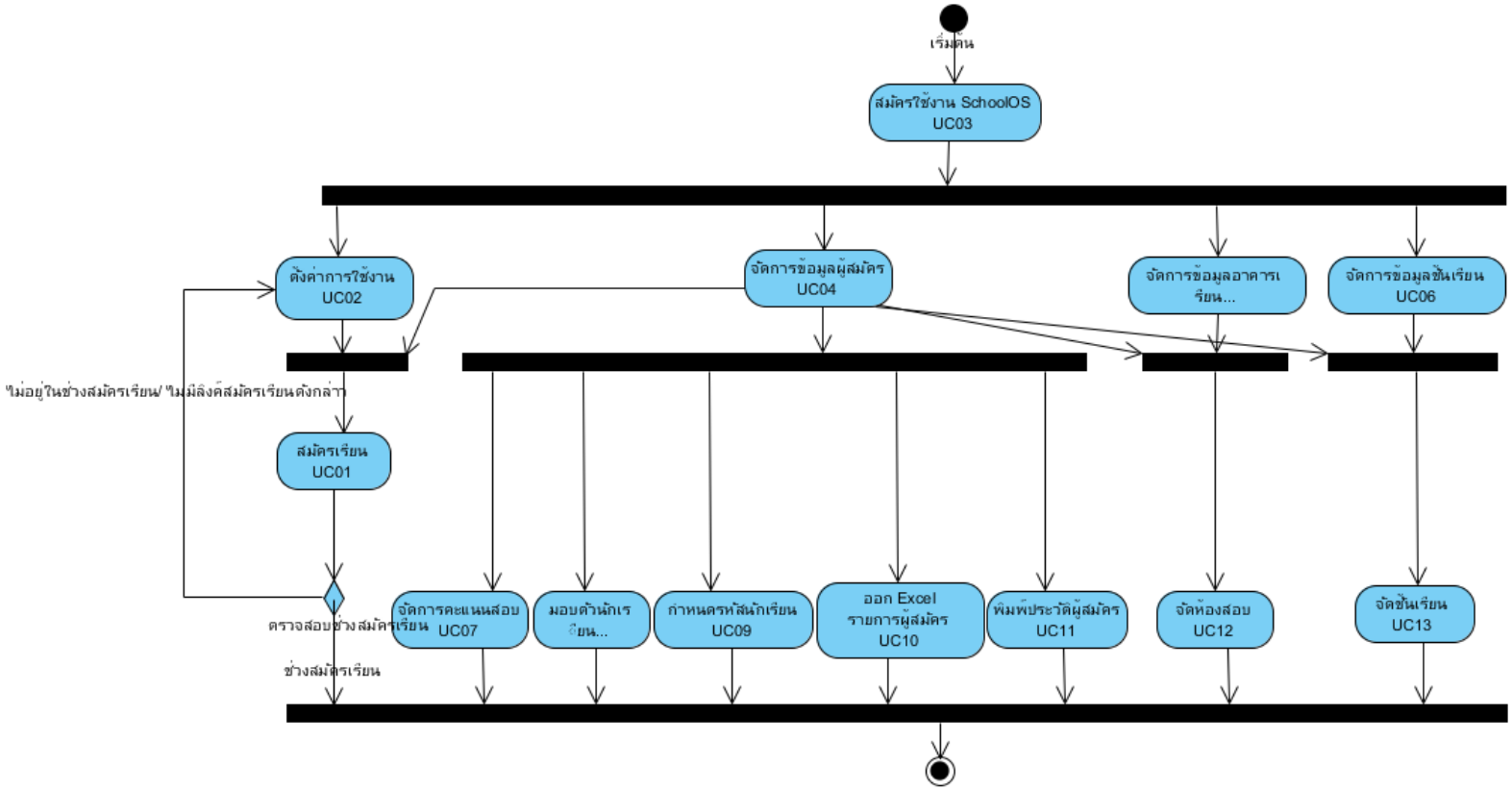
ในการประมาณขนาดในมุมมองระดับล่างจะนำรายละเอียดของยูสเคสของระบบสมัครเรียนมาทำการนับจำนวนของทรานแซคชัน และจำนวนของเส้นทางที่ได้จากจำนวนของซีเนริโอ โดยรายละเอียดของยูสเคสนี้ถูกแสดงดังตารางที่ 26

ผลลัพธ์ของการประมาณขนาดและความพยายามในการพัฒนาซอฟต์แวร์ด้วยวิธีการแอคทิวิตีพอยท์ ของระบบสมัครเรียนที่ถูกนำมาใช้เป็นตัวอย่างซึ่งเป็นโครงการเดียวกับโครงการที่ 1 ของบทที่ 6 ที่เกี่ยวกับการประเมินผล โดยมีรายละเอียดของแต่ละปัจจัยและผลของการประมาณในขั้นตอนต่าง ๆ ดังแสดงภายในตารางที่ 27 – 34 ตามลำดับ

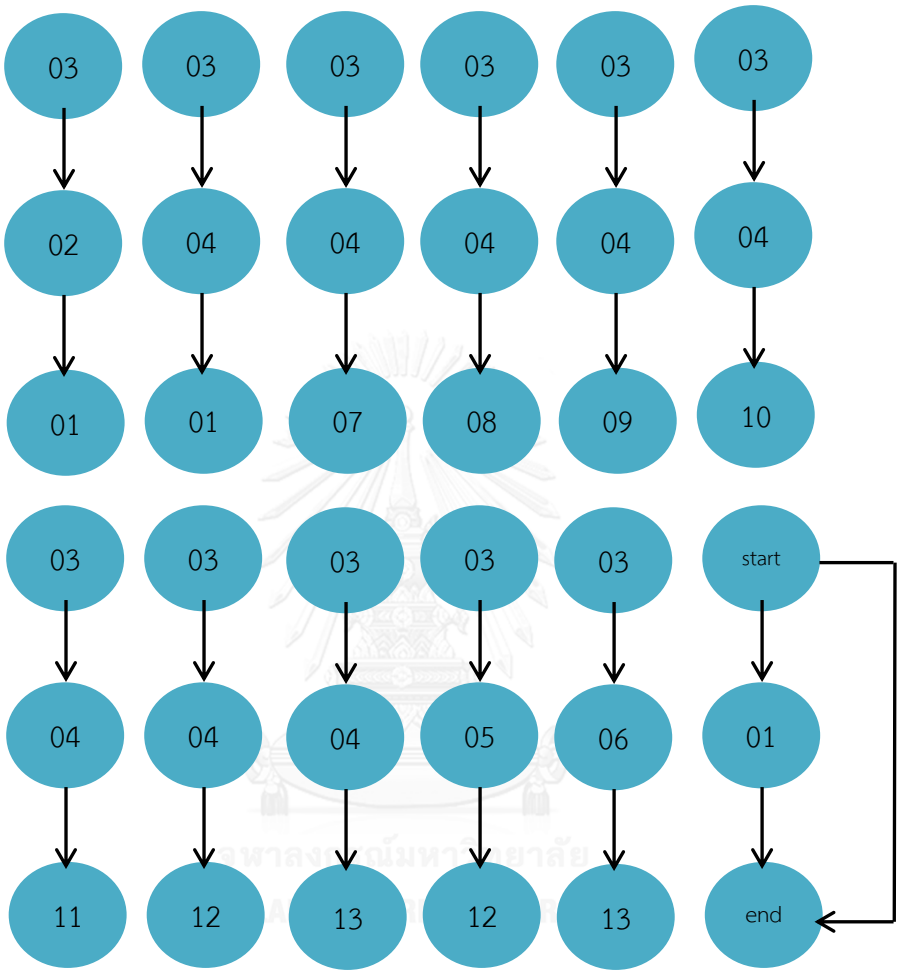
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 53 แผนภาพยูสเคสของระบบสมัครเรียน



รูปที่ 54 แผนภาพกิจกรรมในมุมมองระดับบนของระบบสมัครเรียน



รูปที่ 55 แผนภาพควบคุมกระแสมุมมองระดับบนของระบบสมัครเรียน

ตารางที่ 26 รายละเอียดยูสเคสของระบบสมัครเรียน

Use case name: สมัครเรียน	ID: UC01	Important Level : High
Primary Actor : ผู้สมัคร	Use case Type : Detail , Essential	
Stakeholders and Interests : เจ้าหน้าที่		
Brief Description : ส่วนที่ผู้สมัครใช้ในการสมัครเรียนด้วยตนเองผ่านระบบออนไลน์		
Association : สมัครเรียน		
Basic Flow of Events		
Actor Action	System Response	
1. เข้าระบบด้วย Url โดย parameter id โรงเรียน	2. ระบบตรวจสอบ URL	
	3. ระบบตรวจสอบเงื่อนไข ช่วงการสมัคร	
	4. ระบบแสดงหน้าต่างเข้าสมัครเรียน	
5. ผู้สมัครเลือก สมัครเรียน ออนไลน์	6. ระบบแสดงแบบฟอร์มการรับสมัคร	
7. ผู้สมัครกรอกข้อมูลส่วนตัว และ ประวัติการเรียน		
8. ผู้สมัครกรอกข้อมูลบิดา		
9.ผู้สมัครกรอกข้อมูลมารดา		
10.ผู้สมัครกรอกข้อมูลผู้ปกครอง		
11.ผู้สมัครยืนยันการสมัคร		
	12.ระบบตรวจสอบข้อมูลส่วนตัวถูกต้อง	
	13.ระบบตรวจสอบข้อมูลบิดาถูกต้อง	
	14.ระบบตรวจสอบข้อมูลมารดาถูกต้อง	
	15.ระบบตรวจสอบข้อมูลผู้ปกครองถูกต้อง	
	16. ระบบบันทึกข้อมูลการสมัคร	
	17. ระบบสร้างเอกสารการสมัคร	
	18. ระบบส่ง เอกสาร ทางอีเมล์	
	19. ระบบแสดงหน้าพิมพ์ข้อมูลการสมัคร	
Alternate/Exceptional Flows: 2.a ไม่พบ URL 2.a1 ระบบซ่อนปุ่มสมัคร 3.a ไม่ได้อยู่ในเงื่อนไขช่วงการรับสมัคร		

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.a1 ระบบซ่อนเมนูการสมัคร กรณีไม่ได้อยู่ช่วงการสมัคร 5.b ผู้สมัครยกเลิกการสมัคร 7.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 7.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขบัตรประจำตัวประชาชนใหม่ 7.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 7.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 8.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 8.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 8.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 8.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 9.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำประชาชนซ้ำ 9.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 9.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 9.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 10.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 10.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 10.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 10.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 12.a ข้อมูลไม่ครบถ้วน 12.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require 13.a ข้อมูลไม่ครบถ้วน 13.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require 14.a ข้อมูลไม่ครบถ้วน 14.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require 15.a ข้อมูลไม่ครบถ้วน 15.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require

Use case name: ตั้งค่าการใช้งาน	ID: UC02	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Stakeholders and Interests : ผู้สมัคร		
Brief Description : ส่วนที่เจ้าหน้าที่ใช้ในการตั้งค่าข้อมูลการใช้งานระบบในส่วนต่าง ๆ		

Association : ตั้งค่า		
Basic Flow of Events		
Actor Action	System Response	
1. เลือกเมนูตั้งค่าการใช้งาน	2. ระบบแสดงแบบฟอร์มการตั้งค่า	
3. กรอกรายละเอียดการสมัคร พร้อมบันทึก	4. ระบบบันทึกข้อมูลการสมัคร	
Alternate/Exceptional Flows: 3.a กรณี Url โรงเรียนซ้ำ 3.a1 ระบบขึ้นเตือนให้กรอกใหม่อีกครั้ง 3.b ผู้ใช้ยกเลิกการสมัคร		
Use case name: สมัครใช้งาน SchoolOS	ID: UC03	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : ส่วนที่เจ้าหน้าที่ใช้ในการสมัครใช้งาน SchoolOS		
Association สมัครใช้งาน		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้เข้าระบบ SchoolOS	2. ระบบแสดงแบบฟอร์มการสมัคร	
3. ผู้ใช้กรอกรายละเอียด		
4. ผู้ใช้บันทึกข้อมูลการสมัคร	5. ระบบบันทึกข้อมูลการสมัคร	
	6. ระบบส่ง Email Username & Password	
	7. ระบบสอบถามการเข้าระบบอัตโนมัติ	
	8. ระบบเข้าสู่ระบบภายใน	
Alternate/Exceptional Flows: 3.b ยกเลิกการเข้าสู่ระบบอัตโนมัติ 5.a ไม่อนุญาตสมัคร 5.a1 ระบบขึ้นเตือนกรณี Email สมัครซ้ำ ให้พิมพ์ Email ใหม่		
Use case name: จัดการข้อมูลผู้สมัคร	ID: UC04	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : ส่วนที่เจ้าหน้าที่ใช้ในการเพิ่ม แก้ไข ลบข้อมูลผู้สมัคร		

Association จัดการ	
Basic Flow of Events	
Actor Action	System Response
1. เลือกเมนูจัดการข้อมูลผู้สมัคร	2. ระบบแสดงรายการผู้สมัคร
Basic Flow of Events (เพิ่ม)	
Actor Action	System Response
1. ผู้ใช้เลือกเมนูเพิ่มข้อมูล	2. ระบบแสดงแบบฟอร์มการรับสมัคร
3. ผู้ใช้กรอกข้อมูลส่วนตัว และ ประวัติการเรียน	
4. ผู้ใช้กรอกข้อมูลบิดา	
5. ผู้ใช้กรอกข้อมูลมารดา	
6. ผู้ใช้กรอกข้อมูลผู้ปกครอง	
7.ผู้ใช้ยืนยันการสมัคร	
	8.ระบบตรวจสอบข้อมูลส่วนตัวถูกต้อง
	9.ระบบตรวจสอบข้อมูลบิดาถูกต้อง
	10.ระบบตรวจสอบข้อมูลมารดาถูกต้อง
	11.ระบบตรวจสอบข้อมูลผู้ปกครองถูกต้อง
	12. ระบบบันทึกข้อมูลการสมัคร
	13. ระบบสร้างเอกสารการสมัคร
	14. ระบบส่ง เอกสาร ทางอีเมล์
	15. ระบบแสดงหน้าพิมพ์ข้อมูลการสมัคร
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกการเพิ่มข้อมูล 3.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 3.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 3.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 3.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 4.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 4.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 4.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 4.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 5.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 5.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 5.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 6.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ 6.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่ 6.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน 6.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ 8.a ข้อมูลไม่ครบถ้วน 8.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require 9.a ข้อมูลไม่ครบถ้วน 9.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require 10.a ข้อมูลไม่ครบถ้วน 10.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require 11.a ข้อมูลไม่ครบถ้วน 11.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require

Basic Flow of Events (แก้ไข)	
Actor Action	System Response
1. เลือกเมนูแก้ไขข้อมูลจากรายการผู้สมัคร	2. ระบบแสดงแบบฟอร์มแก้ไขการรับสมัคร
	3. ระบบแทนค่าข้อมูลแก้ไขลงในฟอร์ม
4. ผู้ใช้กรอกข้อมูลส่วนตัว และ ประวัติการเรียน	
5. ผู้ใช้กรอกข้อมูลบิดา	
6. ผู้ใช้กรอกข้อมูลมารดา	
7. ผู้ใช้กรอกข้อมูลผู้ปกครอง	
8. ผู้ใช้ยืนยันการสมัคร	
	9.ระบบตรวจสอบข้อมูลส่วนตัวถูกต้อง
	10.ระบบตรวจสอบข้อมูลบิดาถูกต้อง
	11.ระบบตรวจสอบข้อมูลมารดาถูกต้อง
	12.ระบบตรวจสอบข้อมูลผู้ปกครองถูกต้อง
	13. ระบบบันทึกข้อมูลการสมัคร

Alternate/Exceptional Flows:

1.a ผู้ใช้ยกเลิกการเพิ่มข้อมูล

4.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ

4.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่

4.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน

4.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ

5.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ

5.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่

5.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน

5.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ

6.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ

6.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่

6.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน

6.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ

7.a ไม่สามารถสมัคร เนื่องจากหมายเลขประจำตัวประชาชนซ้ำ

7.a1 ทำการปิดฟอร์ม (Disabled) เพื่อให้กรอกหมายเลขประชาชนใหม่

7.aa ผู้ใช้เลือกนำข้อมูลที่มีมาใช้งาน

7.aa1 ระบบนำข้อมูลที่มีมากรอกในฟอร์มอัตโนมัติ

9.a ข้อมูลไม่ครบถ้วน

9.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require

10.a ข้อมูลไม่ครบถ้วน

10.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require

11.a ข้อมูลไม่ครบถ้วน

11.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require

12.a ข้อมูลไม่ครบถ้วน

12.a1 ระบบขึ้นเตือนเครื่องหมายตกใจ ในช่วงข้อมูลที่ Require

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Basic Flow of Events (ลบ)	
Actor Action	System Response
1. ผู้ใช้เลือกลบข้อมูลจากรายการผู้สมัคร	2. ระบบขึ้นข้อความยืนยัน
3. ผู้ใช้ยืนยันการลบข้อมูลการสมัคร	4. ระบบขึ้นข้อความเตือนยืนยันการลบ
5. ผู้ใช้ยืนยันการลบข้อมูล	6. ระบบลบข้อมูลการสมัคร
Alternate/Exceptional Flows: 5.a ผู้ใช้ยกเลิกการลบข้อมูลการสมัคร	

Use case name: จัดการข้อมูลอาคารเรียน	ID: UC05	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : ส่วนที่เจ้าหน้าที่ใช้ในการเพิ่ม แก้ไข ลบข้อมูลอาคารเรียน		
Association จัดการ		

Basic Flow of Events	
Actor Action	System Response
1. ผู้ใช้เลือกเมนูจัดการข้อมูลอาคารเรียน	2. ระบบแสดงรายการอาคารเรียน

Basic Flow of Events (เพิ่ม)	
Actor Action	System Response
1. ผู้ใช้เลือกเมนูเพิ่มอาคารเรียน	2. ระบบแสดงแบบฟอร์มเพิ่มอาคารเรียน
3. ผู้ใช้กรอกข้อมูลตามแบบฟอร์ม	
4. ผู้ใช้ยืนยันการเพิ่มข้อมูล	5. ระบบบันทึกข้อมูลการสมัคร
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกการเพิ่มอาคารเรียน	

Basic Flow of Events (แก้ไข)	
Actor Action	System Response
1. ผู้ใช้เลือกเมนูแก้ไขข้อมูลอาคารเรียน	2. ระบบแสดงแบบฟอร์มแก้ไขข้อมูลอาคารเรียน
	3. ระบบแทนค่าข้อมูลการแก้ไขบนฟอร์ม
4. ผู้ใช้แก้ไขข้อมูลตามแบบฟอร์ม	
5. ผู้ใช้ยืนยันการแก้ไขข้อมูล	6. ระบบบันทึกข้อมูลอาคารเรียน
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกข้อมูลการแก้ไขข้อมูลอาคารเรียน	

Basic Flow of Events (ลบ)	
Actor Action	System Response
1. ผู้ใช้เลือกลบข้อมูลจากรายการอาคารเรียน	2. ระบบขึ้นข้อความยืนยัน
3. ผู้ใช้ยืนยันการลบข้อมูลอาคารเรียน	
	4. ระบบลบข้อมูลอาคารเรียน
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกการลบข้อมูลอาคารเรียน	

Use case name: จัดการข้อมูลชั้นเรียน	ID: UC06	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : จัดการข้อมูลชั้นเรียน		
Association จัดการ		

Basic Flow of Events	
Actor Action	System Response
1. เลือกเมนูจัดการข้อมูลชั้นเรียน	2. ระบบแสดงรายการข้อมูลชั้นเรียน

Basic Flow of Events (เพิ่ม)	
Actor Action	System Response
1. ผู้ใช้เลือกเมนูเพิ่มชั้นเรียน	2. ระบบแสดงแบบฟอร์มเพิ่มชั้นเรียน
3. ผู้ใช้กรอกข้อมูลตามแบบฟอร์ม	
4. ผู้ใช้ยืนยันการเพิ่มข้อมูล	5. ระบบบันทึกข้อมูลชั้นเรียน
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกการเพิ่มข้อมูลชั้นเรียน	

Basic Flow of Events (แก้ไข)	
Actor Action	System Response
1. เลือกเมนูแก้ไขข้อมูลชั้นเรียน	2. ระบบแสดงแบบฟอร์มแก้ไขข้อมูลชั้นเรียน
	3. ระบบแทนค่าข้อมูลการแก้ไขบนฟอร์ม
4. แก้ไขข้อมูลตามแบบฟอร์ม	
5. ยืนยันการแก้ไขข้อมูล	6. ระบบบันทึกข้อมูลชั้นเรียน
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกข้อมูลการแก้ไขข้อมูลชั้นเรียน	

Basic Flow of Events (ลบ)		
Actor Action	System Response	
1. เลือกลบข้อมูลจากรายการชั้นเรียน	2. ระบบขึ้นข้อความยืนยัน	
3. ยืนยันการลบข้อมูลชั้นเรียน	4. ระบบลบข้อมูลการชั้นเรียน	
Alternate/Exceptional Flows: 3.a ผู้ใช้ยกเลิกการลบข้อมูลชั้นเรียน		
Use case name: จัดการคะแนนสอบ	ID: UC07	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : จัดการคะแนนสอบ		
Association จัดการ		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้เลือกเมนูจัดการคะแนนสอบ	2. ระบบแสดงรายการผู้สมัครที่มีสิทธิ์สอบ	
3. ผู้ใช้ค้นหาผู้สมัครที่ต้องการ		
4. ผู้ใช้เลือกผู้สมัคร	5. ระบบแสดงข้อมูลฟอร์มกรอกคะแนน	
6. ผู้ใช้ที่กรอกข้อมูล		
7. ผู้ใช้บันทึกข้อมูล	8. ระบบบันทึกจัดการคะแนนสอบ	
Alternate/Exceptional Flows: 1.a ผู้ใช้ยกเลิกการเพิ่มคะแนนสอบ		
Use case name: มอบตัวนักเรียน	ID: UC08	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : เจ้าหน้าที่ทำการยืนยันการมอบตัวนักเรียน ในวันมอบตัวนักเรียน		
Association มอบตัว		
Basic Flow of Events		
Actor Action	System Response	
1. เลือกเมนูจัดการมอบตัวนักเรียน	2. ระบบแสดงรายชื่อผู้สมัคร	
	3. ระบบแสดงรายชื่อผู้สมัครที่ยืนยันตัว	
4. ผู้ใช้ค้นรายชื่อผู้สมัคร		
5. ผู้ใช้เลือกผู้สมัครที่ต้องการ		

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Basic Flow of Events (เพิ่ม)	
Actor Action	System Response
1.ผู้ใช้ยืนยันการมอบตัวผู้สมัครที่เลือก	2. ระบบบันทึกรายชื่อผู้ยืนยันตัว
	3. ระบบอัพเดทรายชื่อผู้ที่ยังไม่ถูกยืนยันตัว
Alternate/Exceptional Flows: 1.a ผู้ใช้ยืนยันการมอบตัวผู้สมัครทั้งหมด 1.a1 ระบบยืนยันรายชื่อผู้สมัครจากรายการผู้สมัครที่เลือก	
Basic Flow of Events (ลบ)	
Actor Action	System Response
.1ผู้ใช้ยืนยันการลบข้อมูลการยืนยันตัว	2. ระบบลบรายชื่อผู้ยืนยันตัว
Alternate/Exceptional Flows: 1.a ผู้ใช้เลือกผู้สมัครที่มอบตัวทั้งหมด 1.a1 ระบบลบรายชื่อผู้สมัครที่มอบตัวจากรายการผู้สมัครที่เลือก	

Use case name: กำหนดรหัสนักเรียน	ID: UC09	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : เจ้าหน้าที่กำหนดรหัสนักเรียนในส่วนของนักเรียนที่มีการยืนยันตัว		
Association กำหนดรหัสนักเรียน		

Basic Flow of Events	
Actor Action	System Response
1. ผู้ใช้เลือกเมนูกำหนดรหัสนักเรียน	2. ระบบแสดงรายชื่อผู้สมัครที่มอบตัว
3. ผู้ใช้เลือก สร้างรหัสอัตโนมัติ	4. ระบบทำการตรวจสอบรูปแบบการสร้างรหัสนักเรียนจากส่วนตั้งค่าในการกำหนดค่ารหัสเริ่มต้น
	5. ระบบตรวจสอบรหัสล่าสุดที่มีการใช้งานโดยทำการเพิ่มลำดับใน 3 หลักสุดท้าย
	6. ระบบบันทึกข้อมูลบิดา
	7.ระบบบันทึกข้อมูลมารดา
	8.ระบบบันทึกข้อมูลผู้ปกครอง
	9. บันทึกข้อมูลประวัตินักเรียน
	10. บันทึกประวัติข้อมูลการศึกษานักเรียน

	11. ระบบนำรหัสล่าสุดมาบวกหนึ่งพร้อมบันทึกข้อมูลผู้สมัคร

Alternate/Exceptional Flows:

3.a กรณีการตั้งค่าเป็นอัตโนมัติ

3.a1 ระบบจะสร้างรหัสโดยใช้ พ.ศ. 2 หลัก รหัสช่วงชั้น และลำดับ 3 หลัก เช่น 581001

5.a กรณีมีรหัสที่ใช้งานก่อนหน้า (ใช้งานมาแล้ว)

5.a1 จะทำการเพิ่มลำดับจากรหัสล่าสุด

6.a กรณีมีข้อมูลบิดาอยู่ภายในระบบอยู่แล้ว อาจเนื่องจากมีพี่เรียนอยู่ภายใน

6.a1 ข้อมูลจะถูกแก้ไขให้ล่าสุดแทนการเพิ่มใหม่

6.b กรณีมีข้อมูลมารดาอยู่ภายในระบบอยู่แล้ว อาจเนื่องจากมีพี่เรียนอยู่ภายใน

6.b1 ข้อมูลจะถูกแก้ไขให้ล่าสุดแทนการเพิ่มใหม่

8.c กรณีมีข้อมูลผู้ปกครองอยู่ภายในระบบอยู่แล้ว อาจเนื่องจากมีพี่เรียนอยู่ภายใน

8.c1 ข้อมูลจะถูกแก้ไขให้ล่าสุดแทนการเพิ่มใหม่

9.a กรณีมีข้อมูลนักเรียนอยู่แล้ว ซึ่งอาจจะเนื่องจากการเลื่อนชั้น ด้วยการสอบเข้าใหม่ หรือโควต้า

9.a1 ข้อมูลจะถูกแก้ไขให้ล่าสุดแทนการเพิ่มใหม่

Use case name: ออก Excel รายการผู้สมัคร	ID: UC10	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : เจ้าหน้าที่ทำการออก Excel รายชื่อนักเรียนใหม่เพื่อใช้ในกรณีต่าง ๆ		
Association ออกเอกสาร		

Basic Flow of Events

Actor Action	System Response
1. ผู้ใช้เลือกเมนูกำหนดรหัสนักเรียน	2. ระบบแสดงรายชื่อผู้สมัคร
3. ผู้ใช้เลือกออก Excel รายการผู้สมัคร	4. ระบบแสดงสร้างเอกสารรายการผู้สมัครในรูปแบบ Excel
5. ผู้ใช้ที่บันทึกลงเครื่อง	

Alternate/Exceptional Flows:

3.a ผู้ใช้ยกเลิกการบันทึกลงเครื่อง

Use case name: พิมพ์ประวัติผู้สมัคร	ID: UC11	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	

Brief Description : เจ้าหน้าที่ทำการพิมพ์ประวัติผู้สมัครในรูปแบบ Pdf		
Association ออกเอกสาร		
Sub-flow 1. ก่อนเขียนการทำงานจะต้องสร้างต้นแบบการออกเอกสาร เพื่อนำต้นแบบที่สร้าง มาใช้ในการดึงข้อมูลจากฐานข้อมูลมาวางลงในต้นแบบ เพื่อแปลงเป็น Pdf อีกครั้ง		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้เลือกเมนูข้อมูลผู้สมัคร	2. ระบบแสดงรายชื่อผู้สมัคร	
3. ผู้ใช้ค้นหารายชื่อผู้สมัคร		
4. ผู้ใช้เลือกพิมพ์ประวัติผู้สมัครที่ต้องการ	5. ระบบแสดงสร้างเอกสารประวัติผู้สมัครในรูป Pdf	
6. ผู้ใช้บันทึกลงเครื่อง		
Alternate/Exceptional Flows: 5.a ระบบค้นหาแบบฟอร์มต้นแบบการสร้าง ประวัตินักเรียน ไม่พบ		
Use case name: จัดห้องสอบ	ID: UC12	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : เจ้าหน้าที่ทำการจัดห้องสอบแก่ผู้สมัครเรียน		
Association จัดห้องสอบ		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้เลือกเมนูจัดห้องสอบ		
2. ผู้ใช้เลือกอาคารที่ได้จากส่วนจัดการอาคารเรียน	3. ระบบแสดงรายชื่อผู้สมัครที่สอบในอาคารเรียนนั้น ๆ	
4. ผู้ใช้เลือกระดับชั้นที่ต้องการจัด	5. ระบบแสดงรายชื่อผู้สมัครจากระดับชั้นที่สมัครที่ยังไม่ได้กำหนดห้องสอบ	
Basic Flow of Events (เพิ่ม)		
Actor Action	System Response	
1. ผู้ใช้เลือกผู้สมัครที่ต้องการ		
2. ผู้ใช้ที่บันทึกข้อมูล	3. ระบบบันทึกข้อมูลห้องสอบ	

Alternate/Exceptional Flows: 2.a ผู้ใช้เลือกผู้สมัครทั้งหมด 2.a1 ระบบทำการบันทึกรายการผู้สมัครที่เลือกทั้งหมด		
Basic Flow of Events (ลบ)		
Actor Action	Actor Action	
1. ผู้ใช้เลือกผู้สมัครในห้องสอบที่ต้องการ		
2. ผู้ใช้ลบข้อมูล	3. ระบบลบข้อมูลห้องสอบ	
Alternate/Exceptional Flows: 2.a ผู้ใช้เลือกผู้สมัครในห้องสอบทั้งหมด 2.a1 ระบบทำการลบรายการผู้สมัครที่เลือกทั้งหมด		
Use case name: จัดชั้นเรียน	ID: UC13	Important Level : High
Primary Actor : เจ้าหน้าที่	Use case Type : Detail , Essential	
Brief Description : เจ้าหน้าที่ทำการจัดชั้นเรียนแก่ผู้สมัครเรียน		
Association จัดห้องสอบ		
Basic Flow of Events		
Actor Action	System Response	
1. ผู้ใช้เลือกเมนูจัดชั้นเรียน		
2. ผู้ใช้เลือกชั้นเรียนที่ได้จากส่วนจัดการชั้นเรียน	3. ระบบแสดงรายชื่อผู้สมัครที่สมัครในช่วงชั้นที่ถูกระบุในชั้นเรียน	
	4. ระบบแสดงรายชื่อผู้สมัครที่ถูกจัดในจัดชั้นเรียนนั้น ๆ	
Basic Flow of Events (เพิ่ม)		
Actor Action	System Response	
1. ผู้ใช้เลือกผู้สมัครที่ต้องการ		
2. ผู้ใช้บันทึกข้อมูล	3. ระบบบันทึกข้อมูลชั้นเรียน	
Alternate/Exceptional Flows: 3.a ผู้ใช้เลือกผู้สมัครทั้งหมด 3.a1 ระบบทำการบันทึกชั้นเรียนผู้สมัครที่เลือกทั้งหมด		
Basic Flow of Events (ลบ)		
Actor Action	Actor Action	

1. ผู้ใช้เลือกผู้สมัครในชั้นเรียนที่ต้องการ	
2. ผู้ใช้ลบข้อมูล	3. ระบบลบข้อมูลชั้นเรียน
Alternate/Exceptional Flows: 1.a ผู้ใช้เลือกผู้สมัครในชั้นเรียนทั้งหมด 1.a1 ระบบทำการลบชั้นเรียนผู้สมัครที่เลือกทั้งหมด	

ตารางที่ 27 ตารางสรุปจำนวนของระดับทรานแซคชัน เส้นทาง และจำนวนความซับซ้อนของกิจกรรมของระบบสมัครเรียน

ปัจจัยในการประมาณ	**จำนวนของระดับเรียบง่าย**	**จำนวนของระดับปานกลาง**	**จำนวนของระดับซับซ้อน**
ทรานแซคชัน	0	3	10
ซีเนริโอ	5	5	3
ความซับซ้อนของกิจกรรม	0	0	1

ตารางที่ 28 จำนวนของระดับปัจจัยเส้นทางของระบบสมัครเรียน

ระดับของจำนวนปัจจัยทรานแซคชัน	**จำนวนเส้นทาง**	**น้ำหนัก**	**ค่าน้ำหนักของเส้นทาง (TPW)**
เรียบง่าย	0	2	0
ปานกลาง	3	3	9
ซับซ้อน	10	4	40
ผลรวมค่าน้ำหนักของเส้นทาง (TPW)			49

ตารางที่ 29 จำนวนของระดับปัจจัยกิจกรรมของระบบสมัครเรียน

ระดับของจำนวนปัจจัยกิจกรรม	**จำนวนเส้นทาง**	**น้ำหนัก**	**ค่าน้ำหนักของกิจกรรม (TAW)**
เรียบง่าย	5	1	5
ปานกลาง	5	4	20
ซับซ้อน	3	9	27
ผลรวมค่าน้ำหนักของเส้นทาง (TAW)			52

ตารางที่ 30 ค่าผลรวมของแอคทิวีตีพอยท์ของระบบสมัครเรียน

ระบบ	ค่าผลรวมค่าน้ำหนักของเส้นทาง (TPW)	ค่าผลรวมค่าน้ำหนักของกิจกรรมหรือการกระทำ (TAW)	ผลรวมของแอคทิวีตีพอยท์ (TAP)
สมัครเรียน	49	52	101

ตารางที่ 31 ค่าผลรวมของค่าน้ำหนักความซับซ้อนกิจกรรมของระบบสมัครเรียน

แผนภาพกิจกรรม	ค่า McCabe Complexity	ระดับ	น้ำหนัก	ผลรวมของค่าน้ำหนักความซับซ้อนของกิจกรรม (TACW)
มุมมองระดับบน	13	ซับซ้อนมาก	20	20

ตารางที่ 32 ขนาดของแอคทิวีตีพอยท์ก่อนการปรับปรุงของระบบสมัครเรียน

ระบบ	ผลรวมของแอคทิวีตีพอยท์ (TAP)	TACW	ขนาดของแอคทิวีตีพอยท์ก่อนการปรับปรุง (UAP)
สมัครเรียน	101	20	121

ตารางที่ 33 ขนาดของแอคทิวีตีพอยท์ของระบบสมัครเรียน

ระบบ	ขนาดของแอคทิวีตีพอยท์ก่อนการปรับปรุง (UAP)	ปัจจัยความซับซ้อนทางเทคนิค (TCF)	ปัจจัยความซับซ้อนทางด้านสิ่งแวดล้อม (ECF)	ปัจจัยความเสี่ยงของคน (PRF)	ขนาดซอฟต์แวร์ในหน่วยแอคทิวีตีพอยท์ (AP)
สมัครเรียน	121	1.12	0.62	3.28	275.59

ตารางที่ 34 ความพยายามในการพัฒนาซอฟต์แวร์ของระบบสมัครเรียน

ระบบ	แอคทิวิตีพอยท์ (AP)	ความพยายามที่คาดหวัง (Expected Effort)	ความพยายามจริง (Actual Effort)	ความคลาดเคลื่อนสัมพัทธ์ (RE)
สมัครเรียน	275.59	332.32	301	0.09

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ-นามสกุล นาย ชารีฟ เด่นสุมิตร

วัน/เดือน/ปี เกิด วันที่ 22 กรกฎาคม 2531

ประวัติการศึกษา

ระดับประถมศึกษา โรงเรียนอนุบาลยะลา

ระดับมัธยมศึกษาตอนต้น โรงเรียนคณะราษฎรบำรุงจังหวัดยะลา

ระดับมัธยมศึกษาตอนปลาย โรงเรียนสาธิตมหาวิทยาลัยสงขลานครินทร์ จังหวัดปัตตานี

ระดับอุดมศึกษา มหาวิทยาลัย สงขลานครินทร์
สาขาวิศวกรรมซอฟต์แวร์
คณะเทคโนโลยีและสิ่งแวดล้อม

ระดับปริญญาบัณฑิต สถาบันบัณฑิตพัฒนบริหารศาสตร์ (นิด้า)
สาขา บริหารธุรกิจ (Young Executive MBA)

ประวัติการทำงาน

2554 - 2555 บริษัท อินเตอร์เนชันแนล เมดิคอล ซอฟต์แวร์ จำกัด
ตำแหน่ง Java Developer

2556 - 2558 บริษัท อินเทลลิเจนท์ เอ็นเตอร์ไพรส์ ซอฟต์แวร์ จำกัด
ตำแหน่ง กรรมการผู้จัดการ

2559 - 2560 นักพัฒนาอาชีพอิสระ และ เจ้าของโครงการ
Pinsouq Online Halal Marketplace

ประวัติการอบรม

ASIAN Software Alliance

Business Plan for fund by SIPA

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University