การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

นายปวินท์ อยู่พวง

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 2558
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Time Estimation for Fault Reparation in Software Maintenance Phase

Mr. Pawin Youpoung

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 2015

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์
โดย	นายปวินท์ อยู่พวง
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

ปวินท์ อยู่พวง : การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ (Time Estimation for Fault Reparation in Software Maintenance Phase) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. พรศิริ หมื่นไชยศรี, 134 หน้า.

ขั้นตอนการบำรุงรักษาซอฟต์แวร์เป็นขั้นตอนที่เกี่ยวข้องกับการปรับเปลี่ยนและเปลี่ยนแปลงซอฟต์แวร์เนื่องจากมีการเปลี่ยนแปลงความต้องการ และมีข้อผิดพลาดเกิดขึ้นซึ่งผู้ใช้งานได้รายงานเข้ามา ในขั้นตอนนี้เมื่อข้อผิดพลาดถูกตรวจพบ ผู้ใช้งานทำการแจ้งการแก้ไขข้อผิดพลาดให้กับทีมผู้บำรุงรักษา ทีมผู้บำรุงรักษาจะทำการแก้ไขข้อผิดพลาดในซอฟต์แวร์และทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้น อย่างไรก็ตามทีมผู้บำรุงรักษาและผู้ใช้งานมีความจำเป็นต้องทราบระยะเวลาที่จะใช้ในการแก้ไขข้อผิดพลาดนั้นๆ ก่อนการแก้ไขจริงเพื่อใช้ในการวางแผนการแก้ไขและการทำงานในส่วนอื่นๆ และช่วยให้องค์กรหรือหน่วยงานสามารถจัดการทรัพยากรได้อย่างมีประสิทธิภาพ ดังนั้นนักวิจัยจึงจำเป็นต้องหาวิธีการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

วิทยานิพนธ์นี้นำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งจะทำการสร้างแบบจำลองด้วยวิธีที่แตกต่างกัน คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ ผลการทดลองพบว่าแบบจำลองที่สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ มีประสิทธิภาพและประมาณได้ใกล้เคียงกับค่าจริงมากที่สุด ซึ่งสามารถอธิบายได้ว่าความไม่เสถียรของซอฟต์แวร์มีผลต่อความถูกต้องของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และหากทำการพิจารณาประเภทของข้อผิดพลาดจะทำให้แบบจำลองนั้นมีความถูกต้องมากขึ้น อันดับสองคือแบบจำลองที่สร้างด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล และอันดับสุดท้ายคือแบบจำลองที่สร้างด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

5670275721 : MAJOR SOFTWARE ENGINEERING
KEYWORDS: ESTIMATION / FAULT REPARATION TIME / SOFTWARE MAINTENANCE / EXPONENTIAL DISTRIBUTION / PROBABILITY DENSITY FUNCTION / INSTABILITY METRIC

PAWIN YOUPOUNG: Time Estimation for Fault Reparation in Software Maintenance Phase. ADVISOR: ASSOC. PROF. PORNSIRI MUENCHAISRI, Ph.D., 134 pp.

Software maintenance phase involves modifying and making changes to software due to requirements changes and/or defects reported by users. At this phase, when defects are uncovered, users will inform the problem to software maintenance team. The team will fix the errors and then will know actual fault reparation time of the error. However, the team and users may need to know fault reparation time before the errors are corrected since they can use the estimated time for maintenance plan and the allocation of human resources. Therefore, researcher needs to find an approach to estimate fault reparation time.

In this research, we present an estimation model for fault reparation time. The models are constructed by using 3 different methods: Model 1 constructed by mean of exponential probability density function, Model 2 constructed by applying theory of exponential distribution and Model 3 constructed by applying theory of exponential distribution and using software instability metric. The experiment result shows that Model 3 constructed by applying theory of exponential distribution and using software instability metric is better than other models. This can be explained that software instability metric affects the accuracy of estimate and when considering fault types, the model has better accuracy. Model 1 constructed by mean of exponential probability density function is the second best model and Model 2 constructed by applying theory of exponential distribution comes last.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Department: Computer Engineering Student's Signature .....
Field of Study: Software Engineering Advisor's Signature .....
Academic Year: 2015

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี ที่ได้กรุณาสละเวลาให้คำแนะนำและช่วยเหลือในด้านความรู้ทางวิชาการที่เป็นประโยชน์ต่อการทำวิจัยที่ดีมาโดยตลอด รวมถึงให้กำลังใจและแนวทางการแก้ไขปัญหาที่เกิดขึ้นในระหว่างการทำวิจัย ทั้งยังสนับสนุนและผลักดันการทำวิจัยของผู้วิจัยด้วยความห่วงใยและเอาใจใส่อย่างเต็มกำลังมาโดยตลอด ทำให้ผู้วิจัยสามารถพัฒนางานวิจัยออกมาได้อย่างมีคุณภาพและสำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณอาจารย์เป็นอย่างสูง

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านได้แก่ รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา กรรมการผู้ทรงคุณวุฒิภายในมหาวิทยาลัย และผู้ช่วยศาสตราจารย์ ดร.มฑุปายาส ทองมาก กรรมการผู้ทรงคุณวุฒิภายนอกมหาวิทยาลัย ที่ได้กรุณาสละเวลาให้คำแนะนำอันเป็นประโยชน์ต่อการทำวิจัย รวมถึงขัดเกลา และพิจารณาเนื้อหาให้มีความถูกต้องและครบถ้วนสมบูรณ์ เพื่อให้วิทยานิพนธ์ฉบับนี้มีคุณภาพและมีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณคณาจารย์และบุคลากรภาควิชาวิศวกรรมคอมพิวเตอร์ และเพื่อนๆ ทุกคนในสาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ความรู้ กำลังใจในการทำวิจัย และคำแนะนำในการเรียนและการทำวิจัย รวมถึงที่ได้ให้ความช่วยเหลือในทุกๆ เรื่องอย่างเต็มใจเสมอมา และขอขอบพระคุณทุนสนับสนุนนิสิตระดับปริญญาเอกและโทไปเสนอผลงานวิชาการในต่างประเทศของทางบัณฑิตวิทยาลัยร่วมกับภาควิชาวิศวกรรมคอมพิวเตอร์ ที่สนับสนุนค่าใช้จ่ายในการนำเสนองานในต่างประเทศ

สุดท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา และสมาชิกทุกคนในครอบครัว ที่มอบโอกาสที่ดีในการศึกษาต่อในระดับปริญญาโท และสนับสนุนผู้วิจัยในทุกๆด้าน ทั้งยังให้คำแนะนำและกำลังใจอย่างดียิ่งมาโดยตลอดจนสำเร็จการศึกษา ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะสร้างแรงบันดาลใจและเป็นแรงผลักดันให้เกิดงานวิจัยที่มีคุณภาพและการพัฒนาด้านวิศวกรรมซอฟต์แวร์ต่อไป

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฐ

สารบัญภาพ ..... ณ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของงานวิจัย 3

1.3 ขอบเขตของงานวิจัย 3

1.4 ขั้นตอนและวิธีดำเนินการวิจัย 3

1.5 ประโยชน์ของงานวิจัย 4

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6

2.1 องค์ความรู้และทฤษฎีที่เกี่ยวข้อง 6

2.1.1 การบำรุงรักษาซอฟต์แวร์ (Software Maintenance) 6

2.1.2 ประเภทของข้อผิดพลาด (Fault types) 7

2.1.3 มาตรวัดซอฟต์แวร์ (Software Metrics) 14

2.1.4 พื้นฐานเกี่ยวกับความน่าเชื่อถือ (Basics of reliability theory) 15

2.1.5 แบบจำลองความน่าเชื่อถือของ Jelinski-Moranda 16

2.2 งานวิจัยที่เกี่ยวข้อง 19

2.2.1 การทำนายระยะเวลาในการแก้ไขข้อผิดพลาด (Predicting the Fix Time of Bugs) 19

หน้า

2.2.2 การทำนายระยะเวลาในการแก้ไขข้อผิดพลาด สำหรับการศึกษาเชิงประจักษ์ของโครงการซอฟต์แวร์เชิงพาณิชย์ (Predicting Bug-Fixing Time: An Empirical Study of Commercial Software Projects)..... 20

2.2.3 แบบจำลองการทำนายระยะเวลาในการแก้ไขข้อผิดพลาด โดยใช้การจำแนกแบบเบย์ (Bug Fix-Time Prediction Model Using Naïve Bayes Classifier)..... 20

2.2.4 การทำนายช่วงเวลาของการพัฒนาบนพื้นฐานของแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไป (Predicting Time Range of Development Based on Generalized Software Reliability Model)..... 20

บทที่ 3 การสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ 21

3.1 ภาพรวมแนวคิดของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ 21

3.2 ขั้นตอนการเตรียมชุดโปรแกรม..... 24

3.3 ขั้นตอนการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด..... 25

3.3.1 ขั้นตอนการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของโปรแกรม..... 26

3.3.2 ขั้นตอนการวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด..... 26

3.4 ขั้นตอนการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด..... 27

3.4.1 สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 1)..... 29

3.4.2 สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 2)..... 31

หน้า

3.4.3 สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ (แบบจำลองที่ 3) 32

3.5 ขั้นตอนการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 33

3.5.1 การเปรียบเทียบแบบจำลองโดยใช้การทดสอบนัยสำคัญด้วยสถิติทดสอบที (Paired t-test)..... 34

3.5.2 การประเมินผลแบบจำลองโดยการหาค่าความผิดพลาดสัมพัทธ์และการทำนายที่ระดับแอล 35

บทที่ 4 การทดลองและการประเมินผล แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 37

4.1 การเตรียมข้อมูลสำหรับทดสอบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 37

4.2 วัตถุประสงค์ของการทดลองประมาณค่าและประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 40

4.3 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 40

4.3.1 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล 40

4.3.2 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล 43

4.3.3 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ 44

4.4 การประเมินผลและการเปรียบเทียบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 46

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

4.4.1 การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบที 46

4.4.2 การประเมินผลแบบจำลองโดยการคำนวณค่าความผิดพลาดสัมพัทธ์และการทำนายที่ระดับแอล 48

4.4.2.1 ผลลัพธ์ของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล 48

4.4.2.2 ผลลัพธ์ของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล 52

4.4.2.3 ผลลัพธ์ของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ 55

4.4.3 การเปรียบเทียบการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 59

4.4.3.1 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 2 กรณีไม่พิจารณาประเภทข้อผิดพลาด 61

4.4.3.2 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 2 กรณีพิจารณาประเภทข้อผิดพลาด 61

4.4.3.3 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 3 กรณีไม่พิจารณาประเภทข้อผิดพลาด 61

4.4.3.4 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 3 กรณีพิจารณาประเภทข้อผิดพลาด 61

4.4.3.5 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 2 กับแบบจำลองที่ 3 กรณีไม่พิจารณาประเภทข้อผิดพลาด 62

4.4.3.6 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 2 กับแบบจำลองที่ 3 กรณีพิจารณาประเภทข้อผิดพลาด 62

หน้า

4.4.3.7 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองทั้งหมด กรณีไม่พิจารณาประเภทข้อผิดพลาด 62
4.4.3.8 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองทั้งหมด แยกตามประเภทของข้อผิดพลาด 63
บทที่ 5 การออกแบบและพัฒนาเครื่องมือ สนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ 64
5.1 ความต้องการเชิงหน้าที่ 64
5.1.1 การนำเข้าไฟล์ข้อมูล 64
5.1.2 การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 64
5.1.3 การแสดงผลและการส่งออกข้อมูล 65
5.2 การออกแบบเครื่องมือ 66
5.2.1 แผนภาพยูสเคส 66
5.2.2 แผนภาพคลาส 70
5.2.3 แผนภาพลำดับ 73
5.2.4 แผนภาพกิจกรรม 75
5.2.5 เครื่องมือสนับสนุนในการพัฒนา 76
5.3 การทำงานและส่วนต่อประสานกับผู้ใช้ 77
5.4 การทดสอบเครื่องมือ 85
บทที่ 6 สรุปผลการวิจัย 98
6.1 สรุปผลการวิจัย 98
6.2 ข้อจำกัดของงานวิจัย 100
6.3 งานวิจัยในอนาคต 100
6.4 ผลงานการตีพิมพ์จากวิทยานิพนธ์ 100
รายการอ้างอิง 102

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

ภาคผนวก 105
ภาคผนวก ก การแก้สมการเพื่อใช้ในการหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 106
ภาคผนวก ข การแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดของแบบจำลองที่ 2 109
ภาคผนวก ค การแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดของแบบจำลองที่ 3 111
ภาคผนวก ง การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบที 113
ภาคผนวก จ การคำนวณค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 127
ประวัติผู้เขียนวิทยานิพนธ์ 134

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 1 ตัวอย่างของการรายงานข้อผิดพลาด 12
ตารางที่ 2 มาตรวัดที่ใช้ในงานวิจัย 14
ตารางที่ 3 ตารางแสดงแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 33
ตารางที่ 4 ตารางแสดงรายละเอียดของแต่ละโปรแกรมที่ใช้สำหรับทดสอบแบบจำลองแบบไม่มีการแยกประเภทข้อผิดพลาด 38
ตารางที่ 5 ตารางแสดงจำนวนข้อผิดพลาดของแต่ละประเภทของแต่ละโปรแกรม 39
ตารางที่ 6 ตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 42
ตารางที่ 7 ตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 44
ตารางที่ 8 ตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 45
ตารางที่ 9 ตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 1 49
ตารางที่ 10 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 ของแต่ละโปรแกรมในชุดข้อมูลที่ 1 51
ตารางที่ 11 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 ของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมของชุดข้อมูลที่ 2 51
ตารางที่ 12 ตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 2 . 53
ตารางที่ 13 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2 ของแต่ละโปรแกรมในชุดข้อมูลที่ 1 54
ตารางที่ 14 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2 ของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมของชุดข้อมูลที่ 2 55
ตารางที่ 15 ตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 3 . 57
ตารางที่ 16 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 3 ของแต่ละโปรแกรมในชุดข้อมูลที่ 3 58
ตารางที่ 17 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 3 ของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมของชุดข้อมูลที่ 4 58

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 18 การเปรียบเทียบการประเมินผลแบบจำลองทั้งสามแบบจำลอง ในกรณีไม่ได้พิจารณาประเภทของข้อผิดพลาด 59

ตารางที่ 19 การเปรียบเทียบการประเมินผลแบบจำลองทั้งสามแบบจำลอง ในกรณีพิจารณาประเภทของข้อผิดพลาด 60

ตารางที่ 20 ขั้นตอนการทำงานของยูสเคส Import File 67

ตารางที่ 21 ขั้นตอนการทำงานของยูสเคส Estimate Fault Reparation Time 68

ตารางที่ 22 ขั้นตอนการทำงานของยูสเคส Result Models for Fault Reparation Time 69

ตารางที่ 23 ขั้นตอนการทำงานของยูสเคส Export Files 70

ตารางที่ 24 กรณีทดสอบการนำเข้าไฟล์ข้อมูลในกรณีพิจารณามาตรวัด 87

ตารางที่ 25 กรณีทดสอบการนำเข้าไฟล์ข้อมูลในกรณีไม่พิจารณามาตรวัด 88

ตารางที่ 26 กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 89

ตารางที่ 27 กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 90

ตารางที่ 28 กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 91

ตารางที่ 29 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้น 92

ตารางที่ 30 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้น 93

ตารางที่ 31 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้น 94

ตารางที่ 32 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของการเปรียบเทียบแบบจำลองในรูปแบบตารางและกราฟเส้น 95

ตารางที่ 33 กรณีทดสอบการนำออกหรือบันทึกไฟล์ในรูปแบบตารางและรูปแบบรูปภาพ 96

ตารางที่ 34 สรุปผลการทดสอบเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไข
ข้อผิดพลาด 97

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญภาพ

หน้า

รูปที่ 1 ภาพเวลาระหว่างการเกิดข้อผิดพลาดและเวลาที่แก้ไขข้อผิดพลาดเสร็จหลังส่งมอบซอฟต์แวร์ 7

รูปที่ 2 ภาพการเกิดความล้มเหลวแต่ละครั้งและได้รับการแก้ไขแล้วทุกครั้ง 17

รูปที่ 3 ภาพจำลองการเกิดข้อผิดพลาดโดยมีระยะเวลาระหว่างการเกิดข้อผิดพลาด (t) และระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (r) 18

รูปที่ 4 ภาพรวมแนวคิดของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ 22

รูปที่ 5 ภาพรวมของกิจกรรมและขั้นตอนของงานวิจัย 23

รูปที่ 6 ภาพรวมการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดและการวัดค่าคุณลักษณะของโปรแกรม 25

รูปที่ 7 ขั้นตอนการวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 27

รูปที่ 8 ภาพระหว่างการเกิดความล้มเหลวในแต่ละครั้งและแต่ละครั้งมีการแก้ไขข้อผิดพลาดแล้ว ... 28

รูปที่ 9 ภาพรวมของกิจกรรมการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 29

รูปที่ 10 การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 41

รูปที่ 11 การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 4 41

รูปที่ 12 ผลการเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7 46

รูปที่ 13 ลำดับขั้นตอนของการประเมินผลแบบจำลองที่ 2 52

รูปที่ 14 ลำดับขั้นตอนของการประเมินผลแบบจำลองที่ 3 56

รูปที่ 15 แผนภาพยูสเคสของเครื่องมือสำหรับการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 67

รูปที่ 16 แผนภาพคลาสแสดงองค์ประกอบและความสัมพันธ์ของเครื่องมือ 71

รูปที่ 17 แผนภาพลำดับแสดงการทำงานของระบบ 74

รูปที่ 18 แผนภาพกิจกรรมของเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 75
รูปที่ 19 หน้าจอหลักของเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 77
รูปที่ 20 หน้าจอสำหรับการนำเข้าไฟล์ 78
รูปที่ 21 หน้าจอแสดงข้อมูลนำเข้าในกรณีที่พิจารณามาตรวัด 79
รูปที่ 22 หน้าจอแสดงข้อมูลนำเข้าในกรณีที่ไม่พิจารณามาตรวัด 79
รูปที่ 23 หน้าจอของข้อความแจ้งเตือนเมื่อเลือกไฟล์นำเข้าไม่ถูกต้อง 80
รูปที่ 24 หน้าจอแสดงสถานะของการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 80
รูปที่ 25 หน้าจอแสดงสถานะของการคำนวณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเสร็จสิ้น 81
รูปที่ 26 ตัวอย่างผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 82
รูปที่ 27 ตัวอย่างผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 82
รูปที่ 28 ตัวอย่างผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 83
รูปที่ 29 ตัวอย่างการเปรียบเทียบผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของทั้ง 3 แบบจำลอง 83
รูปที่ 30 ตัวอย่างการบันทึกข้อมูลผลลัพธ์การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 84
รูปที่ 31 ข้อความแจ้งเตือนเมื่อบันทึกข้อมูลสำเร็จ 84
รูปที่ 32 ข้อความแจ้งเตือนเมื่อตั้งชื่อไฟล์ซ้ำ 85
รูปที่ 33 ไฟล์ที่ได้จากการบันทึกผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด 85
รูปที่ 34 ส่วนของการแสดงรายละเอียดของเครื่องมือ 85
รูปที่ 35 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 1 113
รูปที่ 36 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 2 114
รูปที่ 37 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 3 116
รูปที่ 38 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 4 117
รูปที่ 39 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 5 118

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

รูปที่ 40 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 6 119
รูปที่ 41 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7 121
รูปที่ 42 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 8 122
รูปที่ 43 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 9 123
รูปที่ 44 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 10 125
รูปที่ 45 หน้าต่างหลักของโปรแกรมเน็ตบีนส์ 127
รูปที่ 46 หน้าต่างของการนำเข้าไฟล์รหัสต้นฉบับ 128
รูปที่ 47 หน้าต่างของการเลือกนำเข้าไฟล์รหัสต้นฉบับจากที่ได้รวบรวมไว้ 128
รูปที่ 48 หน้าต่างแสดงการเลือกที่อยู่ไฟล์รหัสต้นฉบับ 129
รูปที่ 49 หน้าต่างการเลือกโฟลเดอร์ของรหัสต้นฉบับ 129
รูปที่ 50 หน้าต่างเลือกที่อยู่ของโฟลเดอร์รหัสต้นฉบับ 130
รูปที่ 51 หน้าต่างแสดงรายการที่อยู่ของโฟลเดอร์รหัสต้นฉบับ 130
รูปที่ 52 หน้าต่างแสดงรายการของไฟล์รหัสต้นฉบับ 131
รูปที่ 53 หน้าต่างแสดงการนำเข้าไฟล์รหัสต้นฉบับสำเร็จ 131
รูปที่ 54 หน้าต่างแสดงการเลือกคำสั่งการคำนวณค่ามาตรวัด 132
รูปที่ 55 หน้าต่างแสดงการประมวลผลการคำนวณค่ามาตรวัด 132
รูปที่ 56 หน้าต่างแสดงค่ามาตรวัดต่างๆ 133
รูปที่ 57 หน้าต่างแสดงค่ามาตรวัดในระดับโปรแกรม 133

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงภาพรวมของงานวิจัย ซึ่งประกอบด้วย ที่มาและความสำคัญของปัญหา วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย ขั้นตอนและวิธีดำเนินการวิจัย ประโยชน์ของงานวิจัย และโครงสร้างของเนื้อหาในวิทยานิพนธ์ ซึ่งมีเนื้อหาดังต่อไปนี้

1.1 ที่มาและความสำคัญของปัญหา

ในวงจรชีวิตของซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์จะเกิดขึ้น เมื่อมีข้อผิดพลาดจากการทำงานของตัวซอฟต์แวร์นั้นๆ หรือเกิดจากความต้องการของผู้ใช้งานที่มีการเปลี่ยนแปลง หรือมีความต้องการเพิ่มเติมจากที่เคยได้ระบุไว้ในกระบวนการระบุความต้องการของซอฟต์แวร์ จึงปฏิเสธไม่ได้ว่าการบำรุงรักษาซอฟต์แวร์เป็นกระบวนการที่มีความสำคัญเป็นอย่างยิ่ง การบำรุงรักษาซอฟต์แวร์สามารถเกิดขึ้นหลังจากการส่งมอบซอฟต์แวร์ให้กับผู้ใช้งาน เช่น มีข้อผิดพลาดเกิดขึ้นในผลิตภัณฑ์ซอฟต์แวร์ ผู้บำรุงรักษาจึงจำเป็นต้องเร่งแก้ไขให้ซอฟต์แวร์สามารถใช้งานได้ตามปกติ ดังนั้นจึงเกิดการบำรุงรักษาหลังการส่งมอบผลิตภัณฑ์ซอฟต์แวร์ให้กับผู้ใช้งาน เป็นต้น

ข้อผิดพลาดที่เกิดขึ้นในซอฟต์แวร์สามารถเกิดจากข้อผิดพลาดในการออกแบบและพัฒนาของมนุษย์หรือผู้พัฒนาซอฟต์แวร์ เช่น การไม่เข้าใจในสัญลักษณ์ที่ใช้ในการออกแบบ การวนลูปที่ไม่ถูกต้อง ข้อมูลแสดงออกทางหน้าจอไม่ถูกต้อง เป็นต้น ซึ่งส่งผลให้ซอฟต์แวร์ไม่สามารถทำงานได้ตรงตามความต้องการของผู้ใช้งาน ข้อผิดพลาดต่างๆ เหล่านี้สามารถทำให้ซอฟต์แวร์เกิดการล้มเหลว ก็ต่อเมื่อซอฟต์แวร์ส่วนที่มีข้อผิดพลาดถูกเรียกใช้งาน ซึ่งข้อผิดพลาดนั้นสามารถค้นพบได้ในเอกสารข้อกำหนดความต้องการโดยวิธีการทบทวน (Review) และข้อผิดพลาดที่ถูกค้นพบในรหัสต้นฉบับ (Source Code) จะเรียกว่าข้อผิดพลาดที่เกิดขึ้นในซอฟต์แวร์หรือบั๊ก (Bugs) ซึ่งในขั้นตอนก่อนการส่งมอบซอฟต์แวร์นั้น ได้มีการทดสอบซอฟต์แวร์เป็นอย่างดีจากผู้ทดสอบ แต่ว่าเมื่อส่งมอบซอฟต์แวร์ให้กับผู้ใช้งานแล้ว ซอฟต์แวร์มีโอกาสเกิดข้อผิดพลาดขึ้นได้อีก เนื่องจากผู้ทดสอบอาจจะทดสอบได้ไม่ครอบคลุมตามฟังก์ชันการทำงานทั้งหมดของซอฟต์แวร์นั้น นอกจากนี้ข้อผิดพลาดที่ได้รับการแก้ไขไปแล้ว บางครั้งข้อผิดพลาดที่เกิดขึ้นใหม่อาจเกิดจากข้อผิดพลาดเดิมที่ได้รับการแก้ไขแล้วก็เป็นได้ เนื่องจากทีมทดสอบหรือทีมบำรุงรักษาทำการแก้ไขได้ไม่ถูกต้องตามการทำงานของซอฟต์แวร์ หรือเกิดจากข้อผิดพลาดอื่นๆ ที่ส่งผลกระทบให้ข้อผิดพลาดที่ได้รับการแก้ไขไปแล้ว กลับมาเกิดขึ้นซ้ำอีกครั้งได้

โดยปกติเมื่อเกิดข้อผิดพลาดเกิดขึ้นในซอฟต์แวร์ ผู้ใช้งานทำการแจ้งการแก้ไขข้อผิดพลาดให้กับทีมผู้บำรุงรักษา ผู้บำรุงรักษาซอฟต์แวร์จะทำการแก้ไขข้อผิดพลาดในซอฟต์แวร์ หลังจากนั้นจะทำให้ผู้บำรุงรักษาสามารถทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นๆ เสร็จสิ้น เมื่อผู้ใช้งานแจ้งการแก้ไขข้อผิดพลาดมายังทีมผู้บำรุงรักษา ผู้ใช้งานต้องการทราบระยะเวลาที่จะใช้ในการแก้ไขข้อผิดพลาดนั้นๆ เพื่อนำไปใช้ในการวางแผนการทำงานในส่วนอื่นๆ หรือใช้ในการทำสัญญาการบำรุงรักษาซอฟต์แวร์ แต่ผู้บำรุงรักษาจะทราบระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดก็ต่อเมื่อแก้ไขข้อผิดพลาดนั้นเสร็จสิ้น ดังนั้นจึงจำเป็นต้องหาวิธีการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งหากผู้บำรุงรักษาสามารถบอกระยะเวลาประมาณที่ใช้ในการแก้ไขข้อผิดพลาดได้ จะช่วยให้องค์กรหรือหน่วยงานจัดสรรทรัพยากรและงบประมาณไปใช้ในการบำรุงรักษาหรือใช้งานในด้านอื่นๆ ได้อย่างเหมาะสม

ในปัจจุบันส่วนใหญ่งานวิจัยที่เกี่ยวกับระยะเวลานั้นจะสร้างแบบจำลองประมาณระยะเวลาในการพัฒนาซอฟต์แวร์ [1-3] ในส่วนของงานวิจัยที่เกี่ยวกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีการใช้วิธีการและขั้นตอนที่แตกต่างกันออกไป เช่น แบบจำลองของมาร์คอฟ (Markov Model), วิธีการของมอนติคาร์โล (Monte Carlo method) หรือ การเรียนรู้ของเครื่องจักร (Machine Learning Algorithms) เช่น ต้นไม้ตัดสินใจ (Decision trees), ขั้นตอนวิธีการค้นหาเพื่อนบ้านที่ใกล้ที่สุด (K-Nearest Neighbors Algorithm) หรือ การจำแนกประเภทแบบเบย์อย่างง่าย (Naïve Bayes Classification) [4-9] และในขณะเดียวกันนั้นแบบจำลองความน่าเชื่อถือได้มีการใช้ข้อมูลเกี่ยวกับระยะเวลาที่เกิดความล้มเหลวและเกิดข้อผิดพลาด ซึ่งจะส่งผลต่อความน่าเชื่อถือของซอฟต์แวร์ [10-12]

ดังนั้นงานวิจัยนี้จึงนำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ โดยใช้ข้อมูลของการเกิดข้อผิดพลาดและมีการแก้ไขข้อผิดพลาดเสร็จเรียบร้อย มาร่วมวิเคราะห์ในการสร้างแบบจำลองเพื่อใช้ในการคำนวณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล นอกจากนี้ผู้วิจัยต้องการศึกษาผลกระทบที่ส่งผลต่อระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยใช้ค่ามาตรวัดที่ได้นำเสนอโดย Robert Martin [13] คือมาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric) เพื่อศึกษาว่าความเสถียรของซอฟต์แวร์นั้นมีผลกระทบให้ระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นเพิ่มขึ้นหรือน้อยลง ในการนำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด มีวิธีในการสร้างแบบจำลองที่แตกต่างกัน คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3

สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ พร้อมทั้งพัฒนาเครื่องมือที่สนับสนุนการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษา

1.2 วัตถุประสงค์ของงานวิจัย

1) นำเสนอแนวคิดในการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

2) ออกแบบและพัฒนาเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดตามแบบจำลองที่ได้นำเสนอ

1.3 ขอบเขตของงานวิจัย

1) งานวิจัยนี้ทำการเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด จากข้อมูลของการเกิดข้อผิดพลาดและมีการแก้ไขข้อผิดพลาดเสร็จเรียบร้อย ซึ่งมีทั้งหมด 3 แบบจำลอง คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์

2) มาตรวัดที่ใช้ในแบบจำลองมี 1 มาตรวัด คือมาตรวัดความไม่เสถียรของซอฟต์แวร์เท่านั้น ซึ่งได้อธิบายรายละเอียดในวิทยานิพนธ์ฉบับนี้

3) แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในงานวิจัยนี้ ใช้สำหรับซอฟต์แวร์ที่พัฒนาด้วยภาษาจาวา

4) ค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้นั้น มีหน่วยเป็นวัน

5) พัฒนาเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งพัฒนาด้วยภาษาจาวา และทดสอบบนระบบปฏิบัติการวินโดวส์ (Windows)

6) แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ใช้ชุดโปรแกรมสำหรับทดสอบแบบจำลองจากซอฟต์แวร์จำนวน 10 ซอฟต์แวร์

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1) ศึกษาองค์ความรู้และนิยามทฤษฎีพื้นฐานของการบำรุงรักษา ประเภทของการบำรุงรักษาซอฟต์แวร์ ประเภทของข้อผิดพลาด มาตรวัดซอฟต์แวร์ ทฤษฎีพื้นฐานเกี่ยวกับการแจกแจงแบบเอกซ์

โปแนนเชียล ทฤษฎีพื้นฐานเกี่ยวกับความน่าเชื่อถือ แบบจำลองความน่าเชื่อถือของ Jelinski-Moranda และการประเมินผลแบบจำลอง โดยศึกษาจากหนังสือและบทความทางวิชาการที่เกี่ยวข้อง

2) วิเคราะห์ทฤษฎีที่จะนำมาใช้ในการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

3) รวบรวมรายละเอียดและข้อมูลเกี่ยวกับข้อผิดพลาดที่ได้รับการแก้ไขเรียบร้อยแล้ว เพื่อใช้เป็นค่าจริงและนำมาใช้ในการประเมินผลแบบจำลอง

4) รวบรวมโปรแกรมภาษาจาวาจากเว็บไซต์ของซอฟต์แวร์โอเพนซอร์ซ เพื่อใช้หาค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์

5) ออกแบบรูปแบบและวิธีการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ทั้งหมด 3 วิธี คือการสร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล การสร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และการสร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์

6) ประเมินผลความถูกต้องของแบบจำลองด้วยการทดลองกับซอฟต์แวร์ เปรียบเทียบผลจากการประเมินผลแบบจำลอง และวิเคราะห์ผลการทดลอง

7) ออกแบบและพัฒนาเครื่องมือที่สนับสนุนการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

8) สรุปผล ข้อเสนอแนะ งานวิจัยในอนาคต และเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ของงานวิจัย

1) ได้แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด เพื่อเป็นแนวทางให้ทีมผู้พัฒนาซอฟต์แวร์พิจารณาความสามารถในการบำรุงรักษาซอฟต์แวร์ของทีม ซึ่งจะใช้ในการวางแผนโครงการและใช้ประกอบกับการพิจารณาเพื่อทำสัญญาการบำรุงรักษาซอฟต์แวร์

2) ได้เครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์

โครงสร้างของเนื้อหาในวิทยานิพนธ์ฉบับนี้ถูกแบ่งออกเป็น 6 บทดังนี้

บทที่ 1 เป็นบทนำ โดยจะกล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย ขั้นตอนและวิธีดำเนินการวิจัย และประโยชน์ของงานวิจัย

บทที่ 2 เป็นบทของทฤษฎีและงานวิจัยที่เกี่ยวข้อง ซึ่งจะกล่าวถึงองค์ความรู้และทฤษฎีที่เกี่ยวข้อง และงานวิจัยที่เกี่ยวข้อง

บทที่ 3 เป็นบทของการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งจะกล่าวถึงภาพรวมของแนวคิด ขั้นตอนการเตรียมชุดโปรแกรม ขั้นตอนการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ขั้นตอนการสร้างแบบจำลอง และขั้นตอนในการประเมินผลแบบจำลอง

บทที่ 4 เป็นบทของการทดลองและการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งจะกล่าวถึงการทดลองตั้งแต่เตรียมข้อมูลสำหรับทดสอบแบบจำลอง วัตถุประสงค์ของการทดลอง การประเมินผลแบบจำลอง การประเมินผลและเปรียบเทียบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

บทที่ 5 เป็นบทของการออกแบบและพัฒนาเครื่องมือสนับสนุนการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ ซึ่งจะกล่าวถึงความต้องการเชิงหน้าที่ การออกแบบเครื่องมือ การทำงานและส่วนต่อประสานกับผู้ใช้ และการทดสอบเครื่องมือ

บทที่ 6 เป็นบทของสรุปผลการวิจัย ซึ่งจะกล่าวถึงการสรุปผลการวิจัย ข้อจำกัดของงานวิจัย งานวิจัยในอนาคต และผลงานการตีพิมพ์จากวิทยานิพนธ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้กล่าวถึงองค์ความรู้พื้นฐาน ทฤษฎีที่เกี่ยวข้องและงานวิจัยที่เกี่ยวข้อง ซึ่งเกี่ยวข้องกับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

2.1 องค์ความรู้และทฤษฎีที่เกี่ยวข้อง

ในหัวข้อนี้อธิบายเกี่ยวกับองค์ความรู้และทฤษฎีที่เกี่ยวข้องกับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ ซึ่งประกอบด้วยนิยามของการบำรุงรักษาซอฟต์แวร์ ประเภทของข้อผิดพลาด มาตรวัดซอฟต์แวร์ พื้นฐานเกี่ยวกับความน่าเชื่อถือและแบบจำลองความน่าเชื่อถือของ Jelinski-Moranda โดยอธิบายรายละเอียดดังนี้

2.1.1 การบำรุงรักษาซอฟต์แวร์ (Software Maintenance)

การบำรุงรักษาซอฟต์แวร์ [14, 15] คือกิจกรรมทั้งหมดที่จำเป็นในสนับสนุน ต้นทุนและประสิทธิภาพ ให้กับซอฟต์แวร์ ซึ่งกิจกรรมต่างๆ เป็นกิจกรรมที่ดำเนินในระหว่างขั้นตอนก่อนการส่งมอบ รวมทั้งหลังการส่งมอบ โดยกิจกรรมก่อนการส่งมอบรวมทั้งการวางแผนสำหรับหลังการส่งมอบ, การดำเนินงาน,การสนับสนุน และการกำหนดการขนส่ง และกิจกรรมหลังการส่งมอบรวมทั้งการปรับเปลี่ยนแก้ไขซอฟต์แวร์,การฝึกอบรม และการดำเนินงานให้ความช่วยเหลือ การบำรุงรักษาสามารถแบ่งออกเป็น 4 ประเภทคือ

1) การบำรุงรักษาเพื่อความถูกต้อง (Corrective maintenance) คือการบำรุงรักษาที่เกี่ยวข้องกับการค้นหาและแก้ไขข้อผิดพลาดต่างๆ ของซอฟต์แวร์หลังจากการส่งมอบ โดยจะแก้ไขปัญหาที่พบให้ถูกต้องมากที่สุด เพื่อตอบสนองความต้องการของผู้ใช้

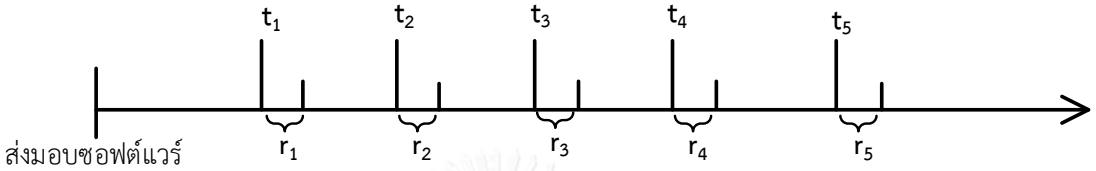
2) การบำรุงรักษาเพื่อปรับเปลี่ยนหรือดัดแปลง (Adaptive maintenance) คือการปรับเปลี่ยนหรือดัดแปลงซอฟต์แวร์หลังจากการส่งมอบ เพื่อให้ซอฟต์แวร์สามารถใช้งานได้เมื่อมีการเพิ่มฟังก์ชันการใช้งานขึ้นมา หรือสามารถใช้งานได้เมื่อมีการเปลี่ยนแปลงสภาพแวดล้อมที่ใช้

3) การบำรุงรักษาเพื่อป้องกัน (Preventive maintenance) คือการค้นหา,ตรวจสอบและแก้ไขก่อนที่จะกลายเป็นข้อผิดพลาดแล้วแฝงอยู่ในซอฟต์แวร์หลังจากการส่งมอบ เพื่อเป็นการป้องกันความผิดพลาดที่อาจจะเกิดขึ้นได้ เมื่อผู้ใช้นำไปใช้งาน

4) การบำรุงรักษาเพื่อความสมบูรณ์ (Perfective maintenance) คือการค้นหา,ตรวจสอบและแก้ไขข้อผิดพลาดที่แฝงอยู่หลังการส่งมอบ ซึ่งเป็นการแก้ไขก่อนจะแสดงอาการของข้อผิดพลาด ที่

อาจจะทำให้เกิดการเข้าใจผิดได้ โดยอาจจะเปลี่ยนแปลงชื่อตัวแปร เปลี่ยนคำอธิบายโปรแกรม หรือชื่อของฟังก์ชัน เพื่อให้สามารถเข้าใจโปรแกรมได้ง่ายขึ้น

งานวิจัยนี้จะนำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งเป็นการบำรุงรักษาเพื่อความถูกต้อง โดยจะทำการแก้ไขข้อผิดพลาดก็ต่อเมื่อผู้ใช้งานแจ้งการเกิดข้อผิดพลาดมายังผู้บำรุงรักษาซอฟต์แวร์ เมื่อผู้บำรุงรักษาซอฟต์แวร์ทำการแก้ไขข้อผิดพลาดเสร็จสิ้นก็จะทำให้ทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด



รูปที่ 1 ภาพเวลาระหว่างการเกิดข้อผิดพลาดและเวลาที่แก้ไขข้อผิดพลาดเสร็จหลังส่งมอบซอฟต์แวร์

จากรูปที่ 1 ภาพเวลาการเกิดข้อผิดพลาดและเวลาที่แก้ไขข้อผิดพลาดเสร็จ สามารถอธิบายได้ว่าเมื่อซอฟต์แวร์ได้มีการส่งมอบให้กับลูกค้า แล้วลูกค้านำไปใช้งาน อาจจะเกิดข้อผิดพลาดเกิดขึ้นในระหว่างที่ใช้งานอยู่ ทำให้ต้องมีการหยุดการใช้งานซอฟต์แวร์ และแจ้งให้กับผู้บำรุงรักษา เพื่อให้ทีมผู้บำรุงรักษาได้ทำการแก้ไขให้ซอฟต์แวร์สามารถใช้งานได้ตามปกติ ซึ่งหลังการส่งมอบจะมีระยะเวลาที่สำคัญสองค่าคือ ระยะเวลาระหว่างการเกิดข้อผิดพลาด (t) และระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (r) ซึ่งระยะเวลา t นั้นสามารถนำไปใช้ในการสร้างแบบจำลองการประมาณความน่าเชื่อถือของซอฟต์แวร์ได้ ซึ่งได้อธิบายไว้ในหัวข้อที่ 2.1.4 และ หัวข้อที่ 2.1.5 โดยงานวิจัยนี้ได้สนใจระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (r) ซึ่งคือระยะเวลาที่สูญเสียไปในการที่หยุดใช้งานซอฟต์แวร์ โดยระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีความสำคัญที่จะแสดงให้เห็นถึงประสิทธิภาพของทีมผู้บำรุงรักษาซอฟต์แวร์ ซึ่งหากผู้บำรุงรักษาสามารถบอกระยะเวลาประมาณที่ใช้ในการแก้ไขข้อผิดพลาดได้ จะช่วยให้องค์กรหรือหน่วยงานจัดสรรทรัพยากรและงบประมาณไปใช้ในการบำรุงรักษาหรือใช้งานในด้านอื่นๆ ได้อย่างเหมาะสม

2.1.2 ประเภทของข้อผิดพลาด (Fault types)

ข้อผิดพลาด(Fault) ในมาตรฐานไอทริปเปิลอี 610.12 [16] มีการนิยามความหมายของข้อผิดพลาดไว้ 2 ความหมาย ดังต่อไปนี้

1) ข้อผิดพลาดในอุปกรณ์ฮาร์ดแวร์หรือส่วนประกอบฮาร์ดแวร์ เช่น ไฟฟ้าลัดวงจร

2) ขั้นตอน กระบวนการ หรือนิยามข้อมูลที่ไม่ถูกต้องในโปรแกรมคอมพิวเตอร์ เช่น คำสั่งในโปรแกรมคอมพิวเตอร์ผิด

งานวิจัยนี้จะนำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งเป็นการบำรุงรักษาเพื่อความถูกต้อง โดยข้อผิดพลาดที่เกิดขึ้นมีหลายประเภท และในบางประเภทเกิดขึ้นในส่วนของเอกสาร หรือส่วนของการออกแบบส่วนต่างๆ ของซอฟต์แวร์ โดยในไอทริปเปิลอี 1044 [17] ได้แบ่งประเภทของข้อผิดพลาดเป็น 13 ประเภทดังนี้

1) ข้อผิดพลาดด้านตรรกะ (Logic problem) เป็นข้อผิดพลาดที่เกิดขึ้นจากการพัฒนาซอฟต์แวร์ แล้วมีการใช้ตรรกะผิด, ตรรกะบางส่วนที่ถูกข้ามไป หรือตรรกะถูกนำไปใช้ไม่ถูกต้องในซอฟต์แวร์ เป็นต้น

2) ข้อผิดพลาดที่เกิดจากอัลกอริทึมบางส่วนที่อยู่ในโค้ดไม่ถูกต้อง (Computation problem) เช่น ตัวถูกดำเนินการในสมการไม่ถูกต้อง หรือเครื่องหมายในวงเล็บใช้ไม่ถูกต้อง เป็นต้น

3) ข้อผิดพลาดที่มีลักษณะของซอฟต์แวร์อินเตอร์เฟซหรือฮาร์ดแวร์อินเตอร์เฟซบางส่วนไม่สามารถทำงานได้อย่างถูกต้อง (Interface/timing problem) เช่น อินพุต/เอาต์พุต ไม่ถูกต้อง หรือความผิดพลาดของเวลาที่ทำให้ข้อมูลสูญหาย เป็นต้น

4) ข้อผิดพลาดด้านการจัดการข้อมูล (Data handling problem) ยกเว้นการจัดการคำนวณถูกจัดการอย่างไม่ถูกต้อง เช่น ข้อมูลเริ่มต้นไม่ถูกต้อง หรือจัดเก็บข้อมูลไม่ถูกต้อง เมื่อเรียกใช้ทำให้ข้อมูลผิดพลาด เป็นต้น

5) ข้อผิดพลาดด้านข้อมูล (Data problem) ข้อมูลที่ใช้ในซอฟต์แวร์บางส่วนไม่ถูกต้องหรือขาดหายไป เช่น ข้อมูลภายนอกไม่ถูกต้องหรือสูญหาย

6) ข้อผิดพลาดด้านเอกสาร (Documentation problem) เป็นข้อผิดพลาดที่เกิดขึ้นเฉพาะเอกสาร มากกว่าที่จะเกิดขึ้นในรหัสต้นฉบับ เช่น คำอธิบายที่จะต้องรวมอยู่ในเอกสารขาดหายไป หรือคำอธิบายในเอกสารนั้นเกิดความขัดแย้งกัน เป็นต้น

7) ข้อผิดพลาดด้านคุณภาพของเอกสาร (Document quality problem) เช่น เอกสารไม่เป็นไปตามซอฟต์แวร์ หรือเอกสารที่มีข้อมูลไม่สอดคล้องกับข้อมูลในเอกสารอื่นๆ เป็นต้น

8) ข้อผิดพลาดเมื่อมีการเพิ่มประสิทธิภาพ (Enhancement) การเพิ่มประสิทธิภาพเป็นข้อเสนอแนะที่มีการเปลี่ยนแปลงตามความต้องการสำหรับซอฟต์แวร์ปัจจุบัน หรือที่มีการเปลี่ยนแปลงโค้ดในผลิตภัณฑ์ปัจจุบัน เพื่อเพิ่มประสิทธิภาพการทำงานของซอฟต์แวร์ให้ดีขึ้น เช่น มี

การเพิ่มความสามารถใหม่ให้กับซอฟต์แวร์ หรือปรับปรุงแก้ไขความสามารถของซอฟต์แวร์ที่ใช้ในปัจจุบัน

9) ข้อผิดพลาดที่เกิดจากการแก้ไขก่อนหน้า (Failure caused by a previous fix) เป็นข้อผิดพลาดที่เกิดจากการแก้ไขปัญหาอื่นๆ หรือการเพิ่มการเปลี่ยนแปลงเข้าไป

10) ข้อผิดพลาดด้านประสิทธิภาพ (Performance problem) เช่น ความเร็วของซอฟต์แวร์ลดลง หรือส่วนประกอบของซอฟต์แวร์ที่ไม่สามารถรับได้เมื่อมีการเปลี่ยนแปลง

11) ข้อผิดพลาดที่เกิดจากการทำงานร่วมกัน (Interoperability problem) เป็นข้อผิดพลาดที่เกิดจากซอฟต์แวร์ หรือส่วนประกอบของซอฟต์แวร์ เข้ากันไม่ได้กับซอฟต์แวร์อื่นๆ

12) ข้อผิดพลาดที่เกิดจากซอฟต์แวร์หรือส่วนประกอบของซอฟต์แวร์ไม่สอดคล้องกับมาตรฐาน (Standards conformance problem) คือซอฟต์แวร์หรือส่วนประกอบของซอฟต์แวร์ไม่เป็นไปตามมาตรฐานที่สอดคล้องกับมาตรฐานเฉพาะ ที่ได้ระบุไว้ในข้อกำหนดความต้องการ

13) ข้อผิดพลาดอื่นๆ (Other problem) เป็นข้อผิดพลาดที่ไม่ตรงกับข้อต่างๆ ในข้างต้น

โดยในงานวิจัยนี้เป็นการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ ซึ่งเป็นการบำรุงรักษาเพื่อความถูกต้อง ดังนั้นจึงสนใจในกรณีที่ข้อผิดพลาดเกิดขึ้นในรหัสต้นฉบับเท่านั้น โดยทำการพิจารณาจากสาเหตุที่ทำให้เกิดข้อผิดพลาด หลังจากพิจารณาแล้วจะได้ประเภทของข้อผิดพลาดที่เกิดขึ้นในรหัสต้นฉบับ 5 ประเภทดังนี้

1) ข้อผิดพลาดด้านตรรกะ (Logic fault) เป็นข้อผิดพลาดที่เกิดขึ้นจากการพัฒนาซอฟต์แวร์บางส่วนที่ถูกข้ามหรือนำไปใช้ไม่ถูกต้องในผลิตภัณฑ์ ซึ่งมีสาเหตุมาจากหลายสาเหตุ เช่น

- ลืมทำขั้นตอนใดขั้นตอนหนึ่ง
- ตรรกะที่ซ้ำกัน
- เงื่อนไขที่มีความรุนแรงถูกละเลย
- ฟังก์ชันการทำงานที่ไม่จำเป็น
- การเข้าใจผิด
- ขาดการทดสอบเงื่อนไขต่างๆ
- ตรวจสอบตัวแปรที่ไม่ถูกต้อง
- การวนลูปไม่ถูกต้อง

2) ข้อผิดพลาดที่เกิดจากอัลกอริทึมบางส่วนที่อยู่ในโค้ดไม่ถูกต้อง (Computation fault) ซึ่งมีสาเหตุมาจากหลายสาเหตุ เช่น

- สมการไม่เพียงพอหรือไม่ถูกต้อง
- ขาดการคำนวณ
- ตัวถูกดำเนินการในสมการไม่ถูกต้อง
- ตัวดำเนินการในสมการไม่ถูกต้อง
- เครื่องหมายวงเล็บใช้ไม่ถูกต้อง
- การสูญเสียความแม่นยำ
- การปัดเศษและตัดทอนของค่าความผิดพลาด
- รูปแบบผสม
- ข้อตกลงเครื่องหมายของข้อผิดพลาด

3) ข้อผิดพลาดที่มีลักษณะของซอฟต์แวร์อินเตอร์เฟซหรือฮาร์ดแวร์อินเตอร์เฟซบางส่วนไม่สามารถทำงานได้อย่างถูกต้อง (Interface/timing fault) ซึ่งมีสาเหตุมาจากหลายสาเหตุ เช่น

- การจัดการการขัดจังหวะไม่ถูกต้อง
- เวลาของอินพุต/เอาต์พุต ไม่ถูกต้อง
- ความผิดพลาดของเวลาทำให้ข้อมูลสูญหาย
- โปรแกรมย่อย/มอดูล ไม่ตรงกัน
- การเรียกโปรแกรมย่อยผิด
- การเรียกที่อยู่ของโปรแกรมย่อยไม่ถูกต้อง
- การเรียกโปรแกรมย่อยที่ไม่มีอยู่จริง
- อาร์กิวเมนต์ของโปรแกรมย่อยไม่สอดคล้องกัน

4) ข้อผิดพลาดด้านการจัดการข้อมูล (Data handling fault) ยกเว้นการจัดการคำนวณ ถูกจัดการอย่างไม่ถูกต้อง ซึ่งมีสาเหตุมาจากหลายสาเหตุ เช่น

- ข้อมูลเริ่มต้นไม่ถูกต้อง
- การเข้าถึงหรือการจัดเก็บข้อมูลไม่ถูกต้อง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

- ตัวบ่งชี้ไม่ถูกต้อง
- การบรรจุข้อมูลไม่ถูกต้อง
- การอ้างอิงตัวแปรของข้อมูลไม่ถูกต้อง
- ข้อมูลที่อ้างอิงอยู่นอกขอบเขต
- การปรับขนาดหรือหน่วยของข้อมูลไม่ถูกต้อง
- ขนาดของข้อมูลไม่ถูกต้อง
- ชนิดของตัวแปรไม่ถูกต้อง
- ตัวแปรที่จองไว้ไม่ถูกต้อง
- ขอบเขตของข้อมูลที่ไม่ถูกต้อง

5) ข้อผิดพลาดด้านข้อมูล (Data fault) ข้อมูลที่ใช้ในซอฟต์แวร์บางส่วนไม่ถูกต้องหรือขาดหายไป ซึ่งมีสาเหตุมาจากหลายสาเหตุ เช่น

- ข้อมูลของเซ็นเซอร์ไม่ถูกต้องหรือสูญหาย
- ข้อมูลของตัวดำเนินการไม่ถูกต้องหรือสูญหาย
- ข้อมูลที่อยู่ในตารางไม่ถูกต้องหรือสูญหาย
- ข้อมูลภายนอกไม่ถูกต้องหรือสูญหาย
- ข้อมูลเอาต์พุตไม่ถูกต้องหรือสูญหาย
- ข้อมูลอินพุตไม่ถูกต้องหรือสูญหาย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 1 ตัวอย่างของการรายงานข้อผิดพลาด

ลำดับ	เรื่อง	คำอธิบายข้อผิดพลาด	ประเภทข้อผิดพลาด
1	Incorrect syntax near the keyword 'convert'	When executing a select statement against a table valued function that takes a date parameter in SQL Server 2008 I get the following error message: Incorrect syntax near the keyword 'convert' select * from dbo.my_table_valued_function(ts '2015-04-22') SQL Server 2012 allows you to use convert in a table valued function parameter but 2008 does not. I'm not sure about 2008 R2 as I have not tested it with that version.	Logic fault
2	Value of alpha miscalculate	The value of alpha from the equation is invalid. I need you to check equation in your program again.	Computation fault
3	CVS module missing	The CVS server module is missing. Please put it back.	Interface/ timing fault

ตารางที่ 1 ตัวอย่างของการรายงานข้อผิดพลาด (ต่อ)

ลำดับ	เรื่อง	คำอธิบายข้อผิดพลาด	ประเภทข้อผิดพลาด
4	Out of Memory error generating sample requests	When generating sample requests for a WSDL with XSDs, SOAP-UI runs out of memory and can't generate the sample requests. Java version: 1.7.0_40 Steps to reproduce 1. Unpack the zip file 2. In SOAP-UI, load the WSDL and click to generate sample requests 3. SOAP-UI will peg the CPU and then run out of memory with the error below Tue Jul 08 09:58:29 PDT 2014:ERROR:java.lang.OutOfMemoryError: GC overhead limit exceeded java.lang.OutOfMemoryError: GC overhead limit exceeded	Data handling fault
5	Incorrect Constant Use in FindNulDeref.java	The analyzeMethod() function checks if a method is volatile and returns, which can't happen. I'm pretty sure it means to check whether it's a bridge method. In truth the constants have the same value, but I was slightly confused by this when I first saw it. $find -iname "*java" | xargs grep -H "method" | grep "ACC_V" ./edu/umd/cs/findbugs/detect/FindNullDeref.java: if ((method.getAccessFlags() & Constants.ACC_VOLATILE) != 0)	Data fault

2.1.3 มาตรวัดซอฟต์แวร์ (Software Metrics)

การวัด (Measurement) [18] คือการกำหนดค่าตัวเลขหรือตัวอักษรให้กับคุณลักษณะของสิ่งที่สนใจ (Entity) เพื่ออธิบายคุณลักษณะ (Attribute) โดยสิ่งที่เราสนใจอาจจะเป็นวัตถุ เช่น คน หรือสิ่งของ หรืออาจจะเป็นเหตุการณ์ เช่น การวางแผนการเดินทาง หรือการออกแบบซอฟต์แวร์ เป็นต้น ซึ่งคุณลักษณะของสิ่งที่สนใจเป็นคุณสมบัติ (Property) ของสิ่งนั้น เช่น การวัดความสูงของคน เป็นหน่วยเซนติเมตร คนคือสิ่งที่สนใจ ความสูงเป็นคุณลักษณะของคนนั้น และค่าตัวเลขสามารถอธิบายคุณสมบัติของคนนั้นได้ว่าสูงหรือปกติหรือเตี้ย เป็นต้น ซึ่งคุณลักษณะของสิ่งที่สนใจจะเรียกว่าตัววัด (Metrics) หรือมาตรวัด จากตัวอย่างข้างต้นนั้นความสูงเป็นมาตรวัดของคน

การวัดนั้นมีความสำคัญต่อทุกขั้นตอนของการพัฒนาซอฟต์แวร์ เนื่องจากสามารถช่วยให้ผู้พัฒนาทราบว่าซอฟต์แวร์ที่พัฒนานั้น เป็นไปตามเป้าหมายและกำหนดการที่วางไว้หรือไม่ นอกจากนี้การวัดยังช่วยในการประมาณค่าใช้จ่าย ต้นทุน กำลังคน และระยะเวลาของการพัฒนาซอฟต์แวร์ การวัดจำนวนของข้อผิดพลาดซอฟต์แวร์ ช่วยให้ผู้พัฒนาทราบถึงความน่าเชื่อถือของซอฟต์แวร์ได้ และสามารถวางแผนการทดสอบได้อย่างเหมาะสม

คุณภาพของซอฟต์แวร์สามารถอธิบายได้ด้วยปัจจัยต่างๆ หลายๆ ปัจจัย นั่นคือสามารถคำนวณจากมาตรวัดต่างๆ ได้ เช่น ความซับซ้อนของซอฟต์แวร์ (Software Complexity) การเกาะกันเป็นก้อน (Cohesion) ความน่าเชื่อถือของซอฟต์แวร์ (Software Reliability) และการบำรุงรักษา (Maintainability) เป็นต้น ในปัจจุบันได้มีมาตรวัดต่างๆ หลายมาตรวัด งานวิจัยนี้สนใจคุณลักษณะที่มีผลกระทบต่อระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยในงานวิจัยนี้เลือกใช้คุณลักษณะความไม่เสถียรของซอฟต์แวร์ (Software Instability) จากการรวบรวมและศึกษาเกี่ยวกับมาตรวัดของคุณลักษณะที่กล่าวมาข้างต้นจากงานวิจัยที่ผ่านมา [13] ซึ่งได้เลือกมาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric) ดังตารางที่ 2

ตารางที่ 2 มาตรวัดที่ใช้ในงานวิจัย

ชื่อมาตรวัด	คำอธิบาย
มาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric: I)	มาตรวัดนี้เป็นตัวบ่งชี้ถึงความต้านทานการได้รับผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงของคลาสต่างๆ ในซอฟต์แวร์

จากตารางที่ 2 มาตรวัดที่ใช้ในงานวิจัย คือมาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric) ซึ่งมาตรวัดแต่ละตัวสามารถหาค่าได้ดังนี้

- มาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric) เป็นมาตรวัดที่ใช้เป็นเป็นตัวบ่งชี้ถึงความต้านทานการได้รับผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงของคลาสต่าง ๆ ในซอฟต์แวร์ ซึ่งสามารถหาได้จากสมการที่ (1)

$$Instability(I) = \frac{Ce}{Ce + Ca} \quad (1)$$

โดยที่

Ca Afferent Coupling คือจำนวนคลาสใดๆ ที่มีการเรียกใช้งานคลาสที่กำลังพิจารณา

Ce Efferent Coupling คือจำนวนคลาสใดๆ ที่ถูกเรียกใช้งานโดยคลาสที่กำลังพิจารณา

2.1.4 พื้นฐานเกี่ยวกับความน่าเชื่อถือ (Basics of reliability theory)

ปัญหาพื้นฐานของทฤษฎีความน่าเชื่อถือ [18] คือการประมาณว่าเมื่อใดซอฟต์แวร์หรือฮาร์ดแวร์ของระบบจะเกิดการล้มเหลว ซึ่งจะอธิบายด้วยความน่าจะเป็นที่ระบบจะเกิดความล้มเหลว ณ ช่วงเวลาหนึ่ง ซึ่งสามารถใช้ฟังก์ชันความน่าจะเป็นของ t (Probability density function(*pdf*) of t) หรือ $f(t)$ เป็นการบอกความน่าจะเป็นนั้นได้ ซึ่ง $f(t)$ ของการแจกแจงแบบเอกซ์โพเนนเชียล (Exponential) คือ $\lambda e^{-\lambda t}$ โดยฟังก์ชันนี้นิยมใช้กันมากในเรื่องของความน่าเชื่อถือ ซึ่งการกำหนดฟังก์ชันนั้นทำให้สามารถคำนวณค่าความน่าจะเป็นที่ซอฟต์แวร์จะไม่ทำงานในช่วงเวลาหนึ่งได้

โดยทั่วไปผู้ใช้งานซอฟต์แวร์ต้องการทราบว่าหลังจากใช้งานซอฟต์แวร์ไปแล้วนั้น จะเกิดความล้มเหลวเมื่อใด ซึ่งสามารถบอกได้ในรูปแบบของความน่าจะเป็นโดยความน่าจะเป็นที่ซอฟต์แวร์จะเกิดความล้มเหลวในช่วงเวลา 0 ถึงเวลา t ใดๆ คือฟังก์ชันความน่าจะเป็นสะสม (Cumulative density function หรือ distribution function) $F(t)$ คือความน่าจะเป็นที่ซอฟต์แวร์จะเกิดความล้มเหลวในช่วงเวลา 0 ถึงเวลา t

$$F(t) = \int_0^t f(t)dt$$

ซึ่งในทางกลับกันนั้นเราสามารถดูความน่าจะเป็นที่ซอฟต์แวร์สามารถทำงานได้ จนกระทั่งเกิดความล้มเหลวครั้งแรกที่เวลา t นั่นคือฟังก์ชันความน่าเชื่อถือของซอฟต์แวร์ (Reliability function) หรือ $R(t)$

$$R(t) = 1 - F(t)$$

ในการแจกแจงแบบเอกซ์โปแนนเชียลนั้น $f(t) = \lambda e^{-\lambda t}$ จะได้ฟังก์ชันความน่าจะเป็นสะสม $F(t)$ ดังนี้

$$F(t) = \int_0^t f(t)dt$$

$$F(t) = \int_0^t \lambda e^{-\lambda t} dt$$

$$F(t) = \left[-e^{-\lambda t}\right]_0^t$$

$$F(t) = 1 - e^{-\lambda t}$$

เวลาเฉลี่ยที่ซอฟต์แวร์ล้มเหลว (Mean time to failure, $MTTF$) คือค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นหรือเรียกอีกอย่างว่า ค่าเฉลี่ยของ T (Expected value of T, $E(T)$)

$$E(T) = \int tf(t)dt$$

ในการแจกแจงแบบเอกซ์โปแนนเชียลนั้น $f(t) = \lambda_i e^{-\lambda_i t_i}$ แล้วเวลาเฉลี่ยที่ซอฟต์แวร์ล้มเหลว สามารถหาได้จากสมการที่ (2), (3)

$$MTTF = E(T) = \int \lambda_i e^{-\lambda_i t_i} t dt = \frac{1}{\lambda_i} \quad (2)$$

$$MTTF = \frac{t_{i-2} + t_{i-1}}{2} \quad (3)$$

ดังนั้นอัตราการล้มเหลวของซอฟต์แวร์ระหว่างการเกิดข้อผิดพลาด (Hazard rate, h(t), λ) สามารถหาได้จากสมการที่ (4)

$$\lambda_i = \frac{2}{t_{i-2} + t_{i-1}} \quad (4)$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.1.5 แบบจำลองความน่าเชื่อถือของ Jelinski-Moranda

แบบจำลองความน่าเชื่อถือของ Jelinski-Moranda [12, 19, 20] หรือ แบบจำลองความน่าเชื่อถือของ JM เป็นแบบจำลองที่นิยมใช้มากที่สุดสำหรับแบบจำลองความน่าเชื่อถือ และถูกใช้อย่างแพร่หลาย เพื่ออธิบายพฤติกรรมของความล้มเหลวของระบบซอฟต์แวร์ โดย

1) สมมติฐานของแบบจำลอง (Model Assumptions) สมมติฐานของแบบจำลองความน่าเชื่อถือของ Jelinski-Moranda ประกอบด้วย

- จำนวนของข้อผิดพลาดเริ่มต้นของซอฟต์แวร์ไม่รู้จำนวนที่แน่นอน
- ข้อผิดพลาดของซอฟต์แวร์แต่ละอันมีความเป็นอิสระต่อกันและมีแนวโน้มเท่ากันที่จะทำให้เกิดความล้มเหลวในระหว่างการทดสอบ
- ช่วงเวลาระหว่างการเกิดขึ้นของข้อผิดพลาดนั้นมีความเป็นอิสระต่อกันและมีการแจกแจงแบบเอกซ์โพเนนเชียล

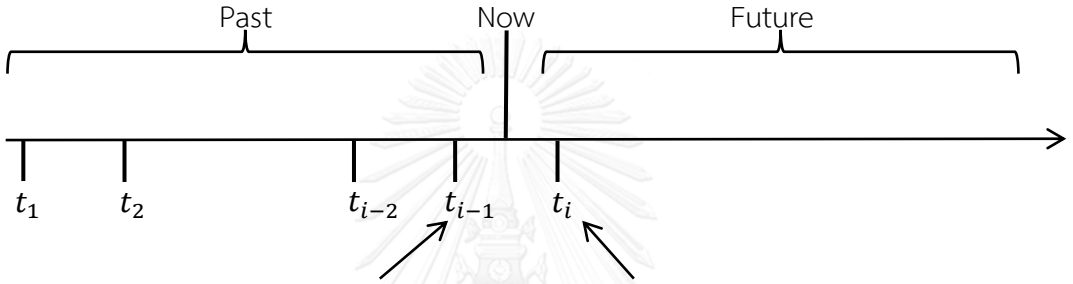
- อัตราความล้มเหลวของซอฟต์แวร์ยังคงที่ ตลอดช่วงเวลาที่เกิดข้อผิดพลาด

- อัตราความล้มเหลวเป็นสัดส่วนกับจำนวนของข้อผิดพลาดที่อยู่ในซอฟต์แวร์

- ข้อผิดพลาดที่ตรวจพบจะถูกลบออกทันทีและข้อผิดพลาดใหม่ที่เกิดขึ้น จะไม่ได้นำมาใช้ในช่วงการกำจัดข้อผิดพลาดที่ตรวจพบ

- เมื่อใดที่เกิดซอฟต์แวร์เกิดความล้มเหลว ข้อผิดพลาดที่สอดคล้องกันหรือเหมือนกันจะถูกลบออก

2) ข้อกำหนดของแบบจำลอง (Model Formulation)



รูปที่ 2 ภาพการเกิดความล้มเหลวแต่ละครั้งและได้รับการแก้ไขแล้วทุกครั้ง

จากรูปที่ 2 อธิบายได้ว่าซอฟต์แวร์มีความน่าจะเป็นในการเกิดความล้มเหลวเป็นครั้งแรก แต่บางครั้งซอฟต์แวร์อาจจะเกิดความล้มเหลวได้หลายๆ ครั้ง ซึ่งความน่าจะเป็นที่ซอฟต์แวร์เกิดความล้มเหลวเป็นครั้งที่ i นั่นคือมีการเกิดความล้มเหลวแล้ว $i-1$ ครั้ง โดยในแต่ละ i จะมี t_i แทนเวลาที่จะเกิดความล้มเหลวในครั้งที่ i โดยแต่ละ t_i มีฟังก์ชันความน่าจะเป็น $f(t_i)$ ถ้าเวลาระหว่างที่เกิดความล้มเหลวที่เกิดขึ้นเป็น $T_i = t_i - t_{i-1}$ โดยที่ $i = 1,2,\dots,N$

- อัตราการล้มเหลวของซอฟต์แวร์ระหว่างการเกิดข้อผิดพลาดที่ $i-1$ และ i สามารถหาได้จากสมการที่ (5)

$$\lambda_i = \big(N-(i-1)\big)\phi, \qquad i = 1,2,\dots,N \tag{5}$$

เมื่อ N คือจำนวนข้อผิดพลาดเริ่มต้นทั้งหมด

ϕ คือผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการเกิดความล้มเหลวทั้งหมด

- ฟังก์ชันความน่าจะเป็นของการแจกแจงแบบเอกซ์โพเนนเชียล เป็นดังสมการที่ (6)

$$F_i(t_i) = 1 - e^{-\lambda_i t_i} \tag{6}$$

เมื่อ t_i คือเวลาระหว่างการเกิดข้อผิดพลาดที่ $i-1$ และ i

- เวลาเฉลี่ยที่ซอฟต์แวร์ล้มเหลว (Mean Time To Failure: MTTF) สามารถหาได้จากสมการที่ (7)

$$MTTF = \frac{1}{\lambda_i} = \frac{1}{\big(N-(i-1)\big)\phi} \tag{7}$$

- ฟังก์ชันความน่าเชื่อถือของเวลาระหว่างการเกิดข้อผิดพลาดที่ $i-1$ และ i สามารถหาได้จากสมการที่ (8)

$$R(t_i) = 1 - F_i(t_i) = e^{-\lambda_i t_i} \tag{8}$$

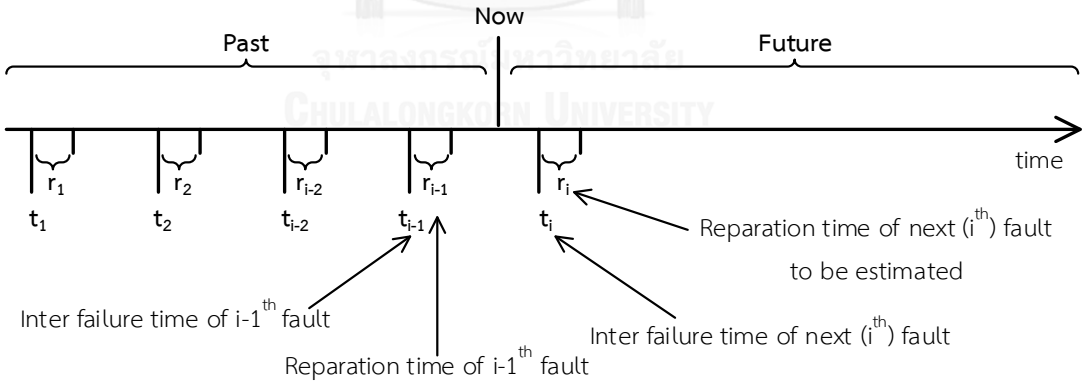
3) การประมาณค่าพารามิเตอร์ (Parameter Estimation)

จากสมการที่ (5) สามารถหาค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการเกิดความล้มเหลวทั้งหมด (ϕ) ได้โดยประมาณได้จากสมการที่ (9) และสามารถหาค่าจำนวนข้อผิดพลาดทั้งหมดของซอฟต์แวร์ที่มีโอกาสจะเกิดขึ้นได้จากสมการที่ (10)

$$\phi = \frac{n}{\widehat{N}(\sum_{i=1}^{n} t_i) - \sum_{i=1}^{n}(i-1)t_i} \tag{9}$$

$$\sum_{i=1}^{n} \frac{1}{\widehat{N}-(i-1)} = \frac{n}{\widehat{N} - \left(\frac{1}{\sum_{i=1}^{n} t_i}\right)(\sum_{i=1}^{n}(i-1)t_i)} \tag{10}$$

4) การประยุกต์แบบจำลองความน่าเชื่อถือของ Jelinski-Moranda



รูปที่ 3 ภาพจำลองการเกิดข้อผิดพลาดโดยมีระยะเวลาระหว่างการเกิดข้อผิดพลาด (*t*) และระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (*r*)

จากรูปที่ 3 อธิบายได้ว่าซอฟต์แวร์มีความน่าจะเป็นในการเกิดข้อผิดพลาด แต่บางครั้งซอฟต์แวร์อาจจะเกิดข้อผิดพลาดได้หลายๆ ข้อผิดพลาด ซึ่งเมื่อเกิดข้อผิดพลาดผู้บำรุงรักษาได้มีการแก้ไขข้อผิดพลาดเสร็จสิ้น ก็จะทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นๆ ซึ่งความน่าจะเป็นที่ซอฟต์แวร์เกิดข้อผิดพลาดเป็นครั้งที่ i นั่นคือมีการเกิดข้อผิดพลาดแล้ว $i-1$ ข้อผิดพลาด โดยแต่ละ

i จะมี t_i แทนระยะเวลาระหว่างที่ข้อผิดพลาดที่ i เกิดขึ้น และมี r_i แทนระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i โดยแต่ละ t_i มีฟังก์ชันความน่าจะเป็น $f(t_i)$ และแต่ละ r_i มีฟังก์ชันความน่าจะเป็น $f(r_i)$

ผู้วิจัยได้ศึกษาแบบจำลองความน่าเชื่อถือเพื่อมาประยุกต์ใช้กับงานวิจัย ซึ่งพบว่าแบบจำลองความน่าเชื่อถือนั้นได้มีการนำทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลมาประยุกต์ใช้ ผู้วิจัยจึงได้ศึกษาการแจกแจงข้อมูลของระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด เมื่อทำการศึกษาพบว่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดมีการแจกแจงแบบเอกซ์โปแนนเชียล ผู้วิจัยจึงได้นำทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลมาประยุกต์ใช้ในงานวิจัยนี้ โดยในงานวิจัยนี้จะศึกษาระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (r) ซึ่งคือระยะเวลาที่สูญเสียไปเพื่อแก้ไขข้อผิดพลาด เมื่อข้อผิดพลาดที่หนึ่งเกิดขึ้นด้วยเวลาระหว่างการเกิดข้อผิดพลาด t_1 ผู้ใช้งานจะทำการแจ้งการแก้ไขข้อผิดพลาดมายังผู้บำรุงรักษา จากนั้นผู้บำรุงรักษาจะทำการแก้ไขข้อผิดพลาดและเมื่อแก้ไขข้อผิดพลาดเสร็จเรียบร้อยจะทำให้ทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่หนึ่ง r_1 จากนั้นซอฟต์แวร์ก็จะทำงานต่อจนกว่าจะเกิดข้อผิดพลาดตัวถัดไป ผู้บำรุงรักษาก็จะทำการแก้ไขข้อผิดพลาดและทำให้ซอฟต์แวร์สามารถทำงานต่อได้ตามปกติ เมื่อข้อผิดพลาดที่ i เกิดขึ้น แบบจำลองที่ผู้วิจัยจะนำเสนอนี้จะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i

2.2 งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้อธิบายเกี่ยวกับการศึกษางานวิจัยที่ผ่านมา ซึ่งเกี่ยวข้องกับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดและการประมาณระยะเวลาในการพัฒนาซอฟต์แวร์ เพื่อเป็นการศึกษาแนวความคิดต่างๆ ของงานวิจัยเหล่านี้ และนำมาประยุกต์ใช้เป็นแนวทางสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่งานวิจัยนี้นำเสนอให้มีประสิทธิภาพมากขึ้น โดยจะแสดงรายละเอียดของแต่ละงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

2.2.1 การทำนายระยะเวลาในการแก้ไขข้อผิดพลาด (Predicting the Fix Time of Bugs)

งานวิจัยนี้ [4] ผู้วิจัยได้นำเสนอแบบจำลองในการทำนายระยะเวลาในการแก้ไขข้อผิดพลาดโดยใช้ข้อมูลคุณลักษณะต่างๆ และเวลาในการแก้ไขข้อผิดพลาดของรายงานข้อผิดพลาด (bug report) จาก Bugzilla จากนั้นสร้างแบบจำลองโดยใช้อัลกอริทึม Exhaustive CHAID ของต้นไม้ตัดสินใจ (Decision tree) ซึ่งได้ทำการจัดการค่าของเวลาในการแก้ไขข้อผิดพลาด โดยที่หากค่าระยะเวลาในการแก้ไขข้อผิดพลาดมากกว่าค่ามัธยฐานจะให้มีค่าเป็น Slow แต่หากค่าระยะเวลาในการแก้ไขข้อผิดพลาดน้อยกว่าหรือเท่ากับค่ามัธยฐานจะให้มีค่าเป็น Fast จากนั้นผู้วิจัยทำการตรวจสอบแบบจำลองโดยใช้ 10 fold cross validation และทดสอบประสิทธิภาพของแบบจำลอง

โดยใช้ Precision, Recall และ AUC ผู้วิจัยพบว่าการใช้ข้อมูลหลังจากการส่งรายงานข้อผิดพลาด (Post-submission) ทำให้การประมาณระยะเวลาในการแก้ไขข้อผิดพลาดนั้นมีถูกต้องมากขึ้น

2.2.2 การทำนายระยะเวลาในการแก้ไขข้อผิดพลาด สำหรับการศึกษาเชิงประจักษ์ของโครงการซอฟต์แวร์เชิงพาณิชย์ (Predicting Bug-Fixing Time: An Empirical Study of Commercial Software Projects)

งานวิจัยนี้ [5] ผู้วิจัยได้สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยใช้แบบจำลองของ Markov เพื่อทำนายจำนวนของข้อผิดพลาดที่จะเกิดในอนาคต จากนั้นใช้วิธี Monte Carlo เพื่อเป็นการทำนายเวลาทั้งหมดที่จำเป็นในการแก้ไขข้อผิดพลาดทั้งหมด จากนั้นใช้วิธีการ k-Nearest Neighbor เพื่อจำแนกประเภทของเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเป็น Slow หรือ Quick แล้วประเมินผลแบบจำลองโดยใช้ F-measure

2.2.3 แบบจำลองการทำนายระยะเวลาในการแก้ไขข้อผิดพลาด โดยใช้การจำแนกแบบเบย์ (Bug Fix-Time Prediction Model Using Naïve Bayes Classifier)

งานวิจัยนี้ [6] ผู้วิจัยได้นำเสนอแบบจำลองการทำนายระยะเวลาในการแก้ไขข้อผิดพลาด โดยใช้การจำแนกแบบเบย์ (Naïve Bayes Classifier) โดยใช้ข้อมูลของรายงานข้อผิดพลาดจาก 3 โปรแกรมคือ Mozilla, Eclipse และ Gnome จากนั้นใช้ 10 fold cross validation ในการตรวจสอบแบบจำลอง และใช้ Precision และ Recall สำหรับวัดประสิทธิภาพของแบบจำลอง

2.2.4 การทำนายช่วงเวลาของการพัฒนาบนพื้นฐานของแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไป (Predicting Time Range of Development Based on Generalized Software Reliability Model)

งานวิจัยนี้ [1] มีจุดประสงค์เพื่อทำนายช่วงเวลาของการพัฒนา โดยการนำเสนอแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไป (GSRM) บนพื้นฐานของกระบวนการสุ่ม โดยการจำลองการพัฒนาที่มีความไม่แน่นอนและมีการเปลี่ยนแปลง จากนั้นนำผลการทดลองไปเปรียบเทียบกับแบบจำลองความน่าเชื่อถือของซอฟต์แวร์อื่นๆ โดยใช้ค่าความไม่แน่นอนและการเปลี่ยนแปลงที่ได้จากแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไป ผู้วิจัยได้เปรียบเทียบแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไปกับแบบจำลองที่มีอยู่ โดยใช้ 2 ชุดข้อมูลเก่าที่เกิดขึ้นจริง และหนึ่งชุดข้อมูลใหม่ที่เกิดขึ้นจริงที่ผู้วิจัยได้เก็บรวบรวม และแสดงให้เห็นว่าแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไปมีความแม่นยำมากขึ้นกว่าแบบจำลองที่มีอยู่แล้ว และแบบจำลองความน่าเชื่อถือของซอฟต์แวร์ทั่วไปนี้สามารถนำมาใช้ในการทำนายช่วงเวลาของการพัฒนาเมื่อมีการเปลี่ยนแปลงได้

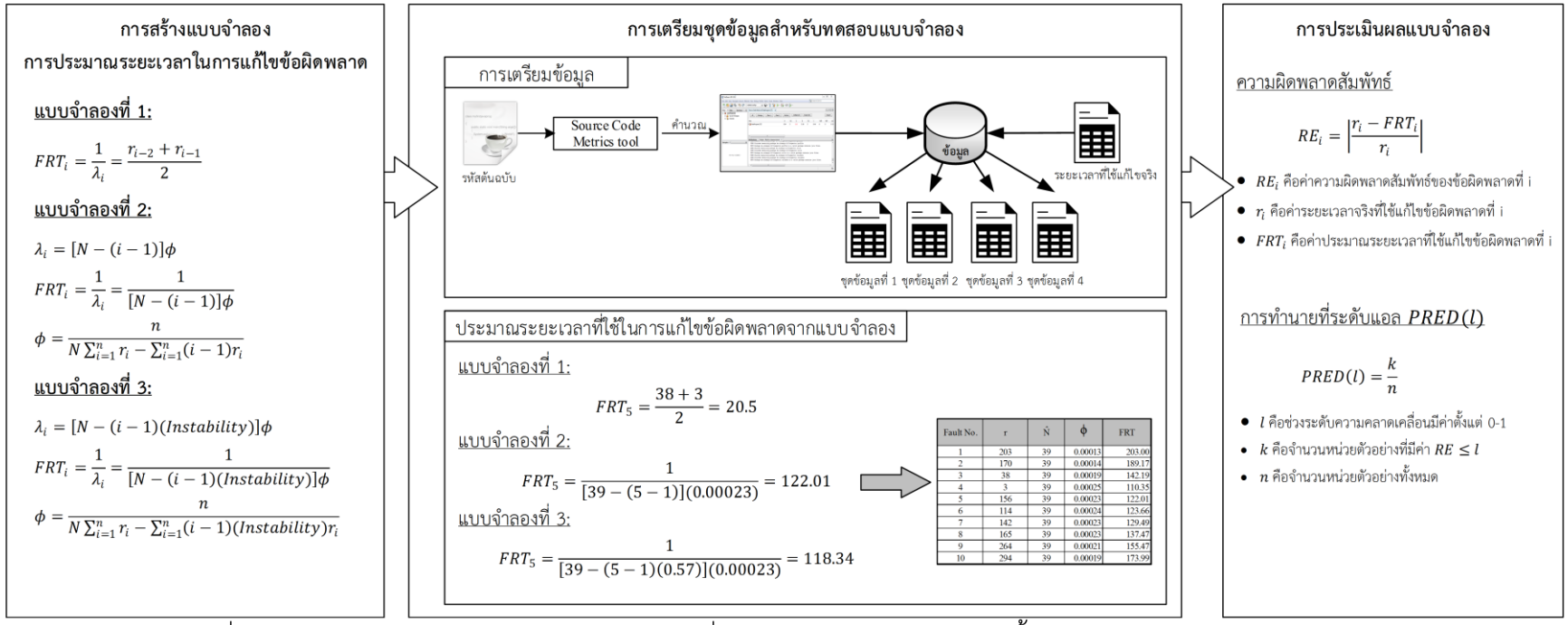
บทที่ 3
การสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

ในบทนี้กล่าวถึงรายละเอียดและขั้นตอนในการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ โดยจะกล่าวถึงภาพรวมในแนวคิดของงานวิจัย จากนั้นจึงกล่าวถึงวิธีการของแต่ละขั้นตอนอย่างละเอียด โดยมีรายละเอียดดังต่อไปนี้

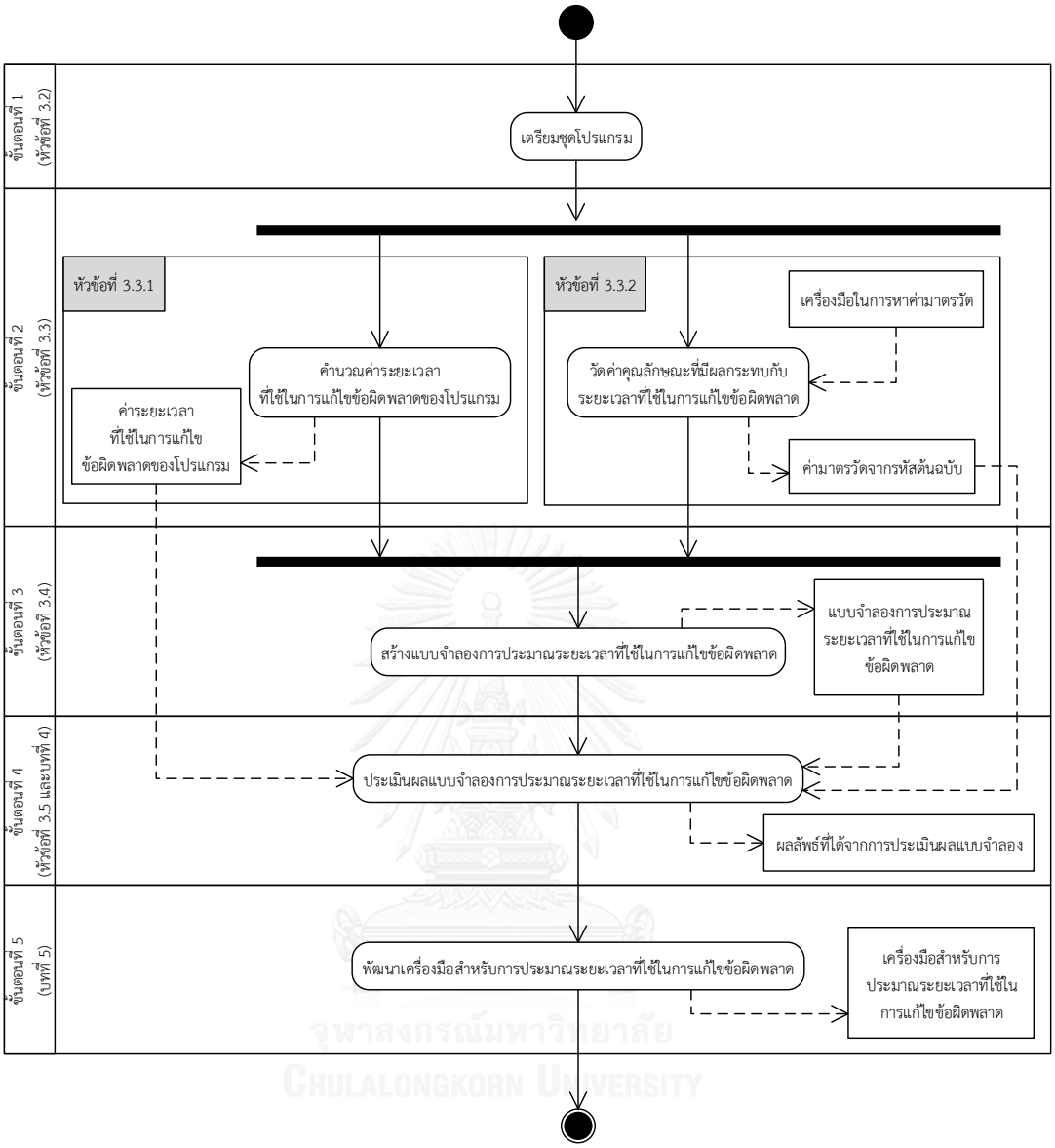
3.1 ภาพรวมแนวคิดของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

การบำรุงรักษาซอฟต์แวร์ คือกิจกรรมทั้งหมดที่จำเป็นในการสนับสนุน ต้นทุนและประสิทธิภาพ ให้กับซอฟต์แวร์ ซึ่งกิจกรรมต่างๆ เป็นกิจกรรมที่ดำเนินการหลังการส่งมอบซอฟต์แวร์ โดยกิจกรรมหลังการส่งมอบซอฟต์แวร์รวมทั้งการปรับเปลี่ยนแก้ไขซอฟต์แวร์,การฝึกอบรม และการดำเนินงานให้ความช่วยเหลือ เป็นต้น การบำรุงรักษาเพื่อความถูกต้อง คือการแก้ไขข้อผิดพลาดต่างๆ เพื่อให้ซอฟต์แวร์สามารถใช้งานได้ต่อไป ส่วนระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด คือจำนวนวันที่ใช้แก้ไขข้อผิดพลาด หลังจากที่ได้รับการแจ้งเตือนมาจากผู้ใช้ว่าเกิดข้อผิดพลาดตรงส่วนใด โดยสามารถวัดได้เมื่อการแก้ไขซอฟต์แวร์เสร็จ

งานวิจัยนี้ศึกษาระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ผู้วิจัยได้ศึกษาแบบจำลองความน่าเชื่อถือเพื่อมาประยุกต์ใช้กับงานวิจัย พบว่าแบบจำลองความน่าเชื่อถือนั้นได้มีการนำทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลมาประยุกต์ใช้ ผู้วิจัยจึงได้ศึกษาการแจกแจงข้อมูลของระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด เมื่อทำการศึกษาพบว่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดมีการแจกแจงแบบเอกซ์โปแนนเชียล [21] ผู้วิจัยจึงได้นำมาประยุกต์ใช้ในงานวิจัยนี้ ซึ่งระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนี้มีการแจกแจงแบบเอกซ์โปแนนเชียล นอกจากนี้ผู้วิจัยต้องการศึกษาผลกระทบที่มีผลกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยใช้คุณลักษณะของความไม่เสถียรของซอฟต์แวร์ งานวิจัยนี้จึงได้นำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล ซึ่งใช้ข้อมูลของระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และค่าคุณลักษณะของความไม่เสถียรของซอฟต์แวร์ เพื่อช่วยให้นักพัฒนาสามารถทราบถึงระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณของแต่ละซอฟต์แวร์ โดยภาพรวมแนวคิดของงานวิจัยดังรูปที่ 4 โดยอธิบายตั้งแต่ขั้นตอนการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด การเตรียมชุดข้อมูล ตลอดจนถึงขั้นตอนการประเมินผลแบบจำลอง



รูปที่ 4 ภาพรวมแนวคิดของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์



รูปที่ 5 ภาพรวมของกิจกรรมและขั้นตอนของงานวิจัย

จากรูปที่ 5 เป็นแผนภาพกิจกรรมที่แสดงให้เห็นถึงภาพรวมของกิจกรรมและขั้นตอนของงานวิจัย ซึ่งประกอบด้วย 5 ขั้นตอน โดยอธิบายรายละเอียดของแต่ละขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 คือขั้นตอนการเตรียมชุดโปรแกรม มีจุดประสงค์เพื่อเตรียมชุดโปรแกรมที่จะใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยรายละเอียดของขั้นตอนนี้ ได้อธิบายไว้ในหัวข้อที่ 3.2

ขั้นตอนที่ 2 คือขั้นตอนการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด มีจุดประสงค์เพื่อคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของโปรแกรมที่ใช้ในการทดสอบแบบจำลอง โดยอธิบาย

รายละเอียดของขั้นตอนนี้ในหัวข้อที่ 3.3.1 และวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยอธิบายรายละเอียดของขั้นตอนนี้ในหัวข้อที่ 3.3.2

ขั้นตอนที่ 3 คือขั้นตอนการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด มีจุดประสงค์เพื่อสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในงานวิจัยนี้จะทำการสร้างแบบจำลองด้วยวิธีที่แตกต่างกัน คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ แล้วนำผลที่ได้จากการทดลองมาเปรียบเทียบ วิเคราะห์ผล และสรุปผล โดยอธิบายรายละเอียดของขั้นตอนนี้ในหัวข้อที่ 3.4

ขั้นตอนที่ 4 คือขั้นตอนการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยทำการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้ในขั้นตอนที่ 3 มาประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของชุดโปรแกรมที่ใช้ในการทดสอบแบบจำลอง แล้วนำค่าประมาณที่ได้และค่าจริงที่ได้จากขั้นตอนที่ 2 มาทำการทดสอบนัยสำคัญด้วยสถิติทดสอบแบบที (Paired t-test) จากนั้นทำการวิเคราะห์ความถูกต้องของแบบจำลองว่าสามารถยอมรับแบบจำลองนั้นๆ ในการนำมาประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้หรือไม่ ซึ่งจะอธิบายรายละเอียดในหัวข้อที่ 3.5 และในบทที่ 4

ขั้นตอนที่ 5 คือขั้นตอนการพัฒนาเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด มีจุดประสงค์เพื่อออกแบบและพัฒนาเครื่องมือเพื่อสนับสนุนการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งอธิบายรายละเอียดเกี่ยวกับการออกแบบและพัฒนาเครื่องมือในบทที่ 5

จากที่ได้กล่าวมานี้ส่วนสำคัญของงานวิจัยคือขั้นตอนที่ 3 คือการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งมีวิธีที่แตกต่างกัน โดยจะอธิบายถึงลำดับขั้นตอนในการสร้างแบบจำลองอย่างละเอียดดังนี้

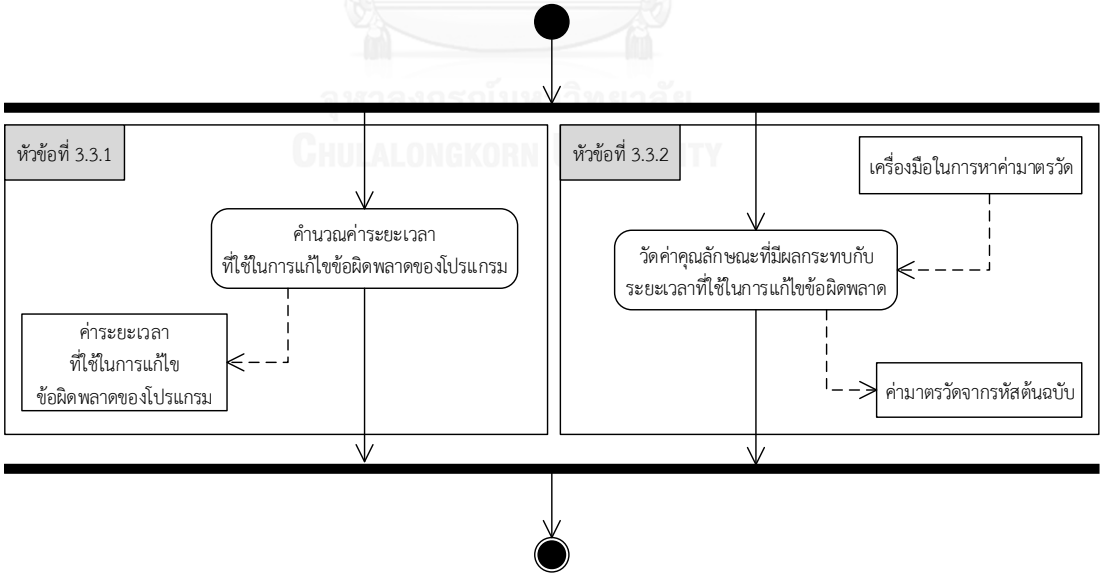
3.2 ขั้นตอนการเตรียมชุดโปรแกรม

ขั้นตอนนี้มีจุดประสงค์เพื่อเตรียมชุดโปรแกรมสำหรับใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยรวบรวมชุดโปรแกรมที่พัฒนาด้วยภาษาจาวาจากเว็บไซต์ของโปรแกรมโอเพนซอร์ซ [22] ซึ่งพิจารณาเลือกโปรแกรมที่มีข้อผิดพลาดมากกว่า 1 ข้อผิดพลาด โดยเป็นโปรแกรมที่มีการแจ้งข้อมูลการเกิดข้อผิดพลาดและรายละเอียดของข้อผิดพลาดเข้ามายังระบบ และผู้บำรุงรักษาทำการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว จากนั้นทำการแยก

ประเภทของข้อผิดพลาดเป็น 5 ประเภท ตามที่ได้กล่าวไว้ในหัวข้อที่ 2.1.3 ซึ่งเป็นประเภทของข้อผิดพลาดที่เกิดขึ้นในรหัสต้นฉบับ โดยแบ่งเป็น 5 ประเภทคือ ข้อผิดพลาดด้านตรรกะ (Logic problem), ข้อผิดพลาดที่เกิดจากอัลกอริทึมบางส่วนที่อยู่ในโค้ดไม่ถูกต้อง (Computation problem), ข้อผิดพลาดที่มีลักษณะของซอฟต์แวร์อินเตอร์เฟซหรือฮาร์ดแวร์อินเตอร์เฟซบางส่วนไม่สามารถทำงานได้อย่างถูกต้อง (Interface/timing problem), ข้อผิดพลาดด้านการจัดการข้อมูล (Data handling problem) และข้อผิดพลาดด้านข้อมูล (Data problem) ซึ่งผลลัพธ์จากขั้นตอนนี้คือชุดโปรแกรมสำหรับทดสอบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

3.3 ขั้นตอนการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

ขั้นตอนนี้มีจุดประสงค์เพื่อคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของโปรแกรม และทำการวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยใช้ชุดโปรแกรมสำหรับทดสอบแบบจำลองที่ได้ในขั้นตอนที่ 3.2 มาคำนวณ ในการคำนวณหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของโปรแกรม และการคำนวณหาค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด มีวิธีการที่แตกต่างกัน โดยแต่ละวิธีจะมีขั้นตอนดังแผนภาพกิจกรรมที่แสดงให้เห็นถึงภาพรวมของการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดและวัดค่าคุณลักษณะของโปรแกรมดังรูปที่ 6



รูปที่ 6 ภาพรวมการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดและการวัดค่าคุณลักษณะของโปรแกรม

จากรูปที่ 6 แสดงภาพรวมการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดและการวัดค่าคุณลักษณะของโปรแกรม โดยแบ่งเป็นขั้นตอนย่อย 2 ขั้นตอน ซึ่งอธิบายรายละเอียดดังต่อไปนี้

3.3.1 ขั้นตอนการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของโปรแกรม

ขั้นตอนนี้มีจุดประสงค์เพื่อคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของโปรแกรม ซึ่งโปรแกรม 1 โปรแกรม อาจจะมีข้อผิดพลาดได้มากกว่า 1 ข้อผิดพลาด ดังนั้นจึงต้องคำนวณหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละข้อผิดพลาดในแต่ละโปรแกรม ซึ่งในการคำนวณหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้น จะใช้ข้อมูลจากเว็บไซต์ของโปรแกรมแต่ละโปรแกรม ซึ่งในแต่ละเว็บไซต์ของโปรแกรมได้รวบรวมข้อมูลวันที่และรายละเอียดของข้อผิดพลาดที่ผู้ใช้แจ้งเข้ามายังเว็บไซต์ของโปรแกรม ข้อมูลที่อยู่ในเว็บไซต์ของโปรแกรมที่นำมาใช้ในงานวิจัยประกอบด้วย วันที่มีการแจ้งการแก้ไขข้อผิดพลาด, วันที่ทำการแก้ไขข้อผิดพลาดเสร็จเรียบร้อย และรายละเอียดของแต่ละข้อผิดพลาดนั้นๆ ซึ่งในการคำนวณหาค่านั้นสามารถหาได้จากการนำวันที่ทำการแก้ไขข้อผิดพลาดเสร็จเรียบร้อย มาลบกับวันที่มีการแจ้งการแก้ไขข้อผิดพลาด ดังสมการที่ (11)

$$r_i = \left(D_{Updated}\right)_i - (D_{Created})_i \qquad (11)$$

เมื่อ r_i คือระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i

$\left(D_{Updated}\right)_i$ คือวันที่ทำการแก้ไขข้อผิดพลาดที่ i เสร็จเรียบร้อย

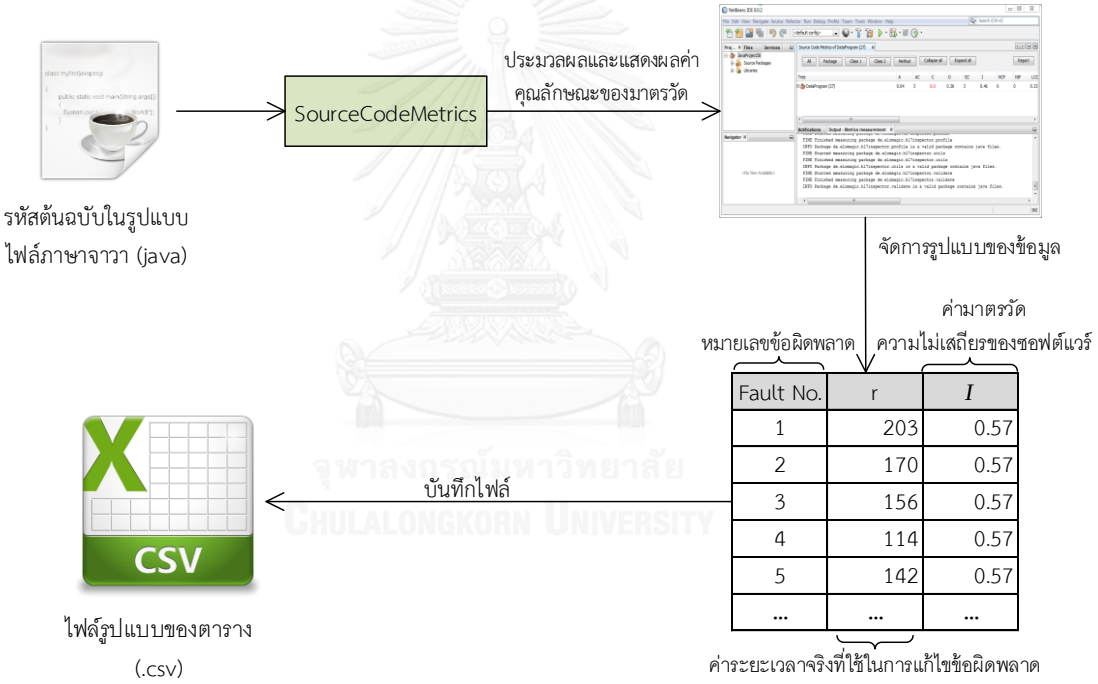
$(D_{Created})_i$ คือวันที่มีการแจ้งการแก้ไขข้อผิดพลาดที่ i

เมื่อทำการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดแล้ว จะพบว่าข้อมูลของระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นมีการแจกแจงแบบเอกซ์โปแนนเชียล จึงได้นำทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลมาประยุกต์ใช้ในงานวิจัยนี้

3.3.2 ขั้นตอนการวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

ขั้นตอนนี้มีจุดประสงค์เพื่อคำนวณค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งในงานวิจัยนี้ต้องการศึกษามาตรวัดความไม่เสถียรของซอฟต์แวร์ว่ามีผลต่อระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยมาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric: I) นั้นเป็นตัวบ่งชี้ถึงความต้านทานการได้รับผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงของคลาสต่างๆ ในซอฟต์แวร์ ในการคำนวณค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์นั้น สามารถหาได้จากการใช้รหัสต้นฉบับจากชุดโปรแกรมสำหรับทดสอบแบบจำลองที่ได้ในขั้นตอนที่ 3.2 มาคำนวณโดยใช้ปลั๊กอินของโปรแกรมเน็ตบีนส์ (NetBeans) ชื่อว่าซอร์ซโค้ดเมตริกซ์ (Source Code Metrics) [23] โดยนำเข้าไฟล์รหัสต้นฉบับที่อยู่ในรูปแบบไฟล์จาวา (Java) โดยโปรแกรมจะทำการ

ประมวลผลและแสดงค่าคุณลักษณะของรหัสต้นฉบับของแต่ละโปรแกรม โดยประกอบด้วยค่ามาตรวัดต่างๆ ซึ่งในงานวิจัยนี้จะเลือกค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ โดยสามารถอธิบายรายละเอียดการคำนวณค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในภาคผนวก จ จากนั้นนำค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละโปรแกรมที่ได้จากขั้นตอนที่ 3.3.1 และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์จากขั้นตอนนี้ มาจัดการรูปแบบของข้อมูลซึ่งประกอบด้วย หมายเลขข้อผิดพลาด (Fault No.), ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด (r) และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ (I) จากนั้นบันทึกไฟล์ให้อยู่ในรูปแบบของตาราง (.csv) เพื่อนำไปใช้ในขั้นตอนของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในขั้นตอนที่ 3.5 โดยภาพรวมในขั้นตอนของการวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด สามารถแสดงได้ดังรูปที่ 7

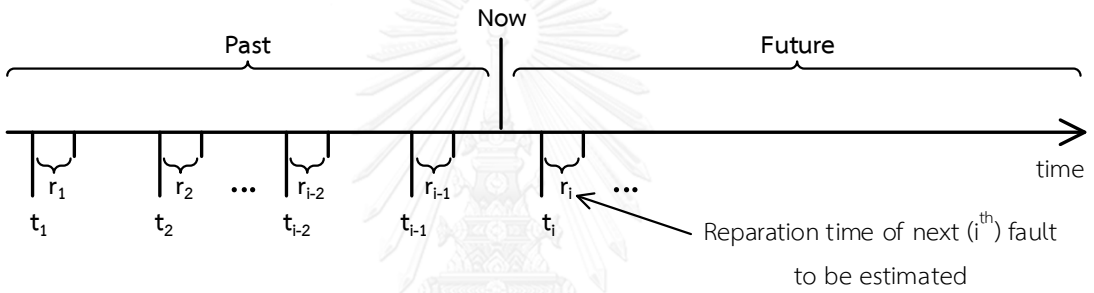


รูปที่ 7 ขั้นตอนการวัดค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

3.4 ขั้นตอนการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

ขั้นตอนนี้มีจุดประสงค์เพื่อสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นมีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล รูปที่ 8 แสดงภาพระหว่างการเกิดความล้มเหลวในแต่ละครั้งและแต่ละครั้งมีการแก้ไขข้อผิดพลาดแล้ว สามารถอธิบายได้ว่าเมื่อซอฟต์แวร์ได้มีการส่งมอบให้กับลูกค้าแล้ว ข้อผิดพลาดอาจจะเกิดขึ้นได้ในระหว่างที่ใช้งานอยู่ ทำให้ต้องมีการหยุดการใช้งานซอฟต์แวร์และทำการแจ้งให้กับผู้บำรุงรักษาเพื่อให้ทีมผู้บำรุงรักษาได้ทำการแก้ไขให้ซอฟต์แวร์สามารถใช้งานได้ตามปกติ เมื่อซอฟต์แวร์ทำงาน

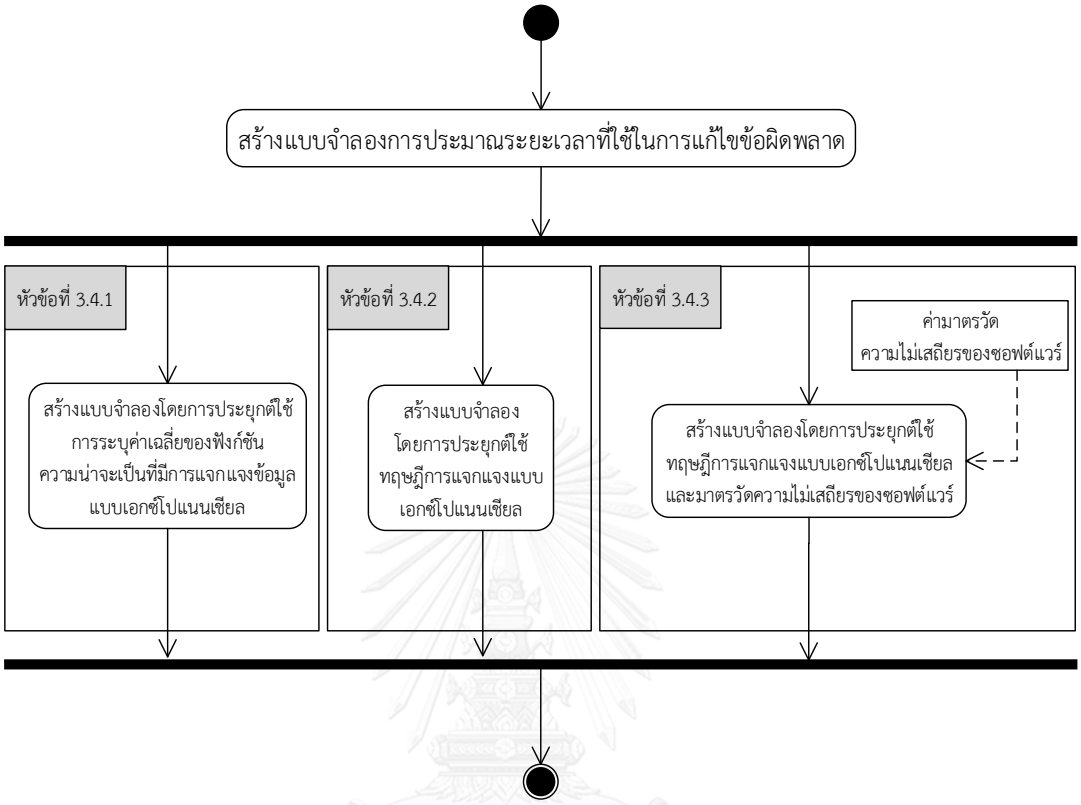
และเกิดข้อผิดพลาดเกิดขึ้นจะมีระยะเวลาที่สำคัญสองค่าคือ ระยะเวลาระหว่างการเกิดข้อผิดพลาด (t) และระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (r) ซึ่งในงานวิจัยนี้สนใจระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด (r) ซึ่งคือระยะเวลาที่สูญเสียไปเพื่อแก้ไขข้อผิดพลาด เมื่อข้อผิดพลาดที่หนึ่งเกิดขึ้นด้วยเวลาระหว่างการเกิดข้อผิดพลาด t_1 ผู้ใช้งานจะทำการแจ้งการแก้ไขข้อผิดพลาดมายังผู้บำรุงรักษา จากนั้นผู้บำรุงรักษาจะทำการแก้ไขข้อผิดพลาดและเมื่อแก้ไขข้อผิดพลาดเสร็จเรียบร้อยจะทำให้ทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่หนึ่ง r_1 จากนั้นซอฟต์แวร์ก็จะทำงานต่อจนกว่าจะเกิดข้อผิดพลาดตัวถัดไป ผู้บำรุงรักษาก็จะทำการแก้ไขข้อผิดพลาดและทำให้ซอฟต์แวร์สามารถทำงานต่อได้ตามปกติ เมื่อข้อผิดพลาดที่ i เกิดขึ้น แบบจำลองที่นำเสนอนี้จะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i โดยในงานวิจัยนี้ได้ใช้ตัวย่อ FRT (Fault Reparation Time) ซึ่งคือระยะเวลาประมาณที่ใช้ในการแก้ไขข้อผิดพลาดนั้นๆ



รูปที่ 8 ภาพระหว่างการเกิดความล้มเหลวในแต่ละครั้งและแต่ละครั้งมีการแก้ไขข้อผิดพลาดแล้ว

งานวิจัยนี้นำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ทั้งหมด 3 แบบจำลอง คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ การสร้างแบบจำลองในแต่ละวิธีนั้นมีแนวคิดมาจากการศึกษางานวิจัยที่ผ่านมาเกี่ยวกับแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งงานวิจัยที่ผ่านมาแบบจำลองส่วนใหญ่ใช้วิธีการของการเรียนรู้ของเครื่องจักร (Machine Learning Algorithms) นอกจากนี้ผู้วิจัยได้ศึกษาแบบจำลองความน่าเชื่อถือเพื่อมาประยุกต์ใช้กับงานวิจัย พบว่าแบบจำลองความน่าเชื่อถือนั้นได้มีการนำทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลมาประยุกต์ใช้ ผู้วิจัยจึงได้ศึกษาการแจกแจงข้อมูลของระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด เมื่อทำการศึกษาพบว่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดมีการแจกแจงแบบเอกซ์โปแนนเชียล ผู้วิจัยจึงได้นำมาประยุกต์ใช้ในงานวิจัยนี้ตามที่ได้กล่าวมาข้างต้น โดยแบบจำลองแต่ละแบบจำลองนั้นมีขั้นตอนในการสร้างที่แตกต่างกัน โดยแบ่งเป็นขั้นตอนย่อย 3 ขั้นตอน ซึ่งได้แสดงดังแผนภาพ

กิจกรรมที่แสดงให้เห็นถึงภาพรวมของกิจกรรมการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ดังรูปที่ 9



รูปที่ 9 ภาพรวมของกิจกรรมการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

3.4.1 สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 1)

ขั้นตอนนี้มีจุดประสงค์เพื่อสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล ซึ่งระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นมีการแจกแจงแบบเอกซ์โปแนนเชียล งานวิจัยนี้จึงประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลและแสดงรายละเอียดการแก้สมการเพื่อใช้ในการหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในภาคผนวก ก ซึ่งผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยจะทำการปรับข้อกำหนดในแบบจำลองพื้นฐานเกี่ยวกับการแจกแจงแบบเอกซ์โปแนนเชียล โดยพิจารณาให้สอดคล้องกับแบบจำลองเดิมจากสมการที่ (2)(3)(4) หน้า 16 ในการปรับข้อกำหนดซึ่งทำการปรับข้อกำหนดเป็นดังนี้

ระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i สามารถประมาณได้ด้วยเวลา r_i ซึ่งคือระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้ทำการรวบรวมไว้ ในการแจกแจงแบบเอกซโปแนนเชียลนั้น ฟังก์ชันความน่าจะเป็นของ r_i (Probability density function (pdf) of r_i) หรือ $f_i(r_i)$ คือ

$$f_i(r_i) = \lambda_i e^{-\lambda_i r_i} \tag{12}$$

เมื่อ r_i คือระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i

λ_i คืออัตราการแก้ไขข้อผิดพลาดที่ i ด้วย r_i

การแจกแจงแบบเอกซโปแนนเชียลนั้นเมื่อ $f_i(r_i) = \lambda_i e^{-\lambda_i r_i}$ จะได้ฟังก์ชันความน่าจะเป็นสะสม $F_i(r_i)$ ดังนี้

$$F_i(r_i) = \int_0^r f_i(r)dr$$
$$F_i(r_i) = \int_0^r \lambda_i e^{-\lambda_i r_i} dr$$
$$F_i(r_i) = \left[-e^{-\lambda_i r_i}\right]_0^r$$
$$F_i(r_i) = 1 - e^{-\lambda_i r_i}$$

ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (r_i) จะถูกใช้ในการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (FRT_i) และระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด คือค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นหรือเรียกอีกอย่างว่า ค่าเฉลี่ยของ R (Expected value of R, $E(R)$) ซึ่งแสดงดังสมการที่ (13)

$$E(R_i) = \int_{i=1}^{n} r_i f_i(r_i)dr \tag{13}$$

ในการแจกแจงแบบเอกซ์โปแนนเชียล $f_i(r_i) = \lambda_i e^{-\lambda_i r_i}$ ซึ่งระยะเวลาเฉลี่ยที่ใช้ในการแก้ไขข้อผิดพลาดที่ i จะได้ดังสมการที่ (14)

$$FRT_i = E(R_i) = \int_{i=1}^{n} r_i \lambda_i e^{-\lambda_i r_i} dr = \frac{1}{\lambda_i} \tag{14}$$

ในการหาค่าของ $\frac{1}{\lambda_i}$ นั้น สามารถหาได้จากค่าเฉลี่ยของระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ $i-1$ (r_{i-1}) และ $i-2$ (r_{i-2}) ดังนั้นสามารถคำนวณระยะเวลาเฉลี่ยที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (FRT_i) ได้จากสมการที่ (15) และสามารถคำนวณอัตราการแก้ไขข้อผิดพลาดที่ i (λ_i) ได้จากสมการที่ (16)

$$FRT_i = \frac{1}{\lambda_i} = \frac{r_{i-2} + r_{i-1}}{2} \tag{15}$$

เมื่อ FRT_i คือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 3,4,5, \ldots$

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3, \ldots$

$$\lambda_i = \frac{2}{r_{i-2} + r_{i-1}} \tag{16}$$

3.4.2 สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 2)

ขั้นตอนนี้มีจุดประสงค์เพื่อสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล ซึ่งผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยแบบจำลองเดิมคือแบบจำลองความน่าเชื่อถือนั้น ได้ใช้ข้อมูลระยะเวลาระหว่างการเกิดข้อผิดพลาดมาใช้ในแบบจำลอง โดยแบบจำลองความน่าเชื่อถือได้มาจากการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล งานวิจัยนี้ได้มีการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล เพื่อให้สามารถหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยจะปรับข้อกำหนดในแบบจำลองความน่าเชื่อถือโดยพิจารณาให้สอดคล้องกับแบบจำลองเดิมจากสมการ (5)(6)(7) หน้า 17 ซึ่งทำการปรับข้อกำหนดเป็นดังนี้

สมการคำนวณอัตราการแก้ไขข้อผิดพลาดที่ i (λ_i) แสดงในสมการที่ (17) ดังนั้นสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (FRT_i) ได้จากสมการที่ (18)

$$\lambda_i = [N - (i-1)]\phi, i = 1,2, \ldots, N \tag{17}$$

$$FRT_i = \frac{1}{\lambda_i} = \frac{1}{[N - (i-1)]\phi} \tag{18}$$

เมื่อ FRT_i คือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3, \ldots$

λ_i คืออัตราการแก้ไขข้อผิดพลาดที่ i ด้วย r_i

N คือจำนวนเริ่มต้นของข้อผิดพลาดทั้งหมดในซอฟต์แวร์

ϕ คือค่าคงที่ของผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมด

จากสมการที่ (17)(18) ϕ ในแบบจำลองนั้นสามารถประมาณโดยใช้วิธีการประมาณค่าความน่าจะเป็นสูงสุด (Maximum Likelihood Estimation) โดยจะแสดงรายละเอียดของการแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดของแบบจำลองที่ 2 ในภาคผนวก ข ซึ่งการหาค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมด (ϕ) แสดงในสมการที่ (19)

$$\phi = \frac{n}{N(\sum_{i=1}^{n} r_i) - \sum_{i=1}^{n}(i-1)r_i} \quad (19)$$

เมื่อ ϕ คือค่าคงที่ของผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไข

N คือจำนวนเริ่มต้นของข้อผิดพลาดทั้งหมดในซอฟต์แวร์

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3,\dots,n$

n คือจำนวนข้อผิดพลาด ณ ตำแหน่งที่ต้องการประมาณค่า เมื่อ $n \leq N$

3.4.3 สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ (แบบจำลองที่ 3)

ขั้นตอนนี้มีจุดประสงค์เพื่อสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ ซึ่งผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ ในการสร้างแบบจำลองวิธีนี้จะคล้ายกับขั้นตอนที่ 3.4.2 และจะใช้ข้อมูลของค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric) ซึ่งมาตรวัดนี้เป็นตัวบ่งชี้ถึงความต้านทานการได้รับผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงของคลาสต่างๆ ในซอฟต์แวร์ ซึ่งคำนวณได้จากรหัสต้นฉบับ เพื่อเป็นการศึกษาว่าความเสถียรของซอฟต์แวร์นั้นมีผลกระทบให้ระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นเพิ่มขึ้นหรือน้อยลง งานวิจัยนี้ได้มีการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล โดยจะปรับข้อกำหนดในแบบจำลองความน่าเชื่อถือโดยพิจารณาให้สอดคล้องกับแบบจำลองเดิม ซึ่งทำการปรับข้อกำหนดเป็นดังนี้

สมการคำนวณอัตราการแก้ไขข้อผิดพลาดที่ i (λ_i) แสดงในสมการที่ (20) ดังนั้นสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (FRT_i) ได้จากสมการที่ (21)

$$\lambda_i = [N - (i-1)(Instability)]\phi, i = 1,2,\dots,N \quad (20)$$

$$FRT_i = \frac{1}{\lambda_i} = \frac{1}{[N - (i-1)(Instability)]\phi} \quad (21)$$

เมื่อ FRT_i คือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3,\dots$

λ_i คืออัตราการแก้ไขข้อผิดพลาดที่ i ด้วย r_i

N คือจำนวนเริ่มต้นของข้อผิดพลาดทั้งหมดในซอฟต์แวร์

$Instability$ คือค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์

ϕ คือค่าคงที่ของผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมด

จากสมการที่ (20)(21) ϕ ในแบบจำลองนั้นสามารถประมาณโดยใช้วิธีการประมาณค่าความน่าจะเป็นสูงสุด (Maximum Likelihood Estimation) โดยจะแสดงรายละเอียดของการแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดของแบบจำลองที่ 2 ในภาคผนวก ค ซึ่งการหาค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมด (ϕ) แสดงในสมการที่ (22)

$$\phi = \frac{n}{N(\sum_{i=1}^{n} r_i) - \sum_{i=1}^{n}(i-1)(Instability)r_i} \quad (22)$$

เมื่อ ϕ คือค่าคงที่ของผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไข

N คือจำนวนเริ่มต้นของข้อผิดพลาดทั้งหมดในซอฟต์แวร์

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3,\dots,n$

n คือจำนวนข้อผิดพลาด ณ ตำแหน่งที่ต้องการประมาณค่า เมื่อ $n \leq N$

$Instability$ คือค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์

ตารางที่ 3 ตารางแสดงแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

วิธีการสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด	**แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด**
การระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 1)	$FRT_i = \frac{r_{i-2} + r_{i-1}}{2}$
ประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 2)	$FRT_i = \frac{1}{[N-(i-1)]\phi}$
ประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ (แบบจำลองที่ 3)	$FRT_i = \frac{1}{[N-(i-1)(Instability)]\phi}$

3.5 ขั้นตอนการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

เมื่อได้แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด จะทำการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลอง โดยใช้ข้อมูลจากชุดโปรแกรมสำหรับทดสอบมาประมาณค่าจากแบบจำลองทั้งหมด เมื่อประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากทั้ง

สามแบบจำลองเสร็จเรียบร้อยแล้ว จากนั้นทำการทดสอบนัยสำคัญด้วยสถิติทดสอบที (Paired t-test) เพื่อเปรียบเทียบผลจากแบบจำลองว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน และเมื่อทำการทดสอบนัยสำคัญด้วยสถิติทดสอบทีเรียบร้อยแล้ว จากนั้นทำการประเมินผลแบบจำลองโดยการหาค่าความผิดพลาดสัมพัทธ์ และการทำนายที่ระดับแอล ซึ่งมีรายละเอียดของแต่ละขั้นตอนดังนี้

3.5.1 การเปรียบเทียบแบบจำลองโดยใช้การทดสอบนัยสำคัญด้วยสถิติทดสอบที (Paired t-test)

เมื่อประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากทั้งสามแบบจำลองเสร็จเรียบร้อยแล้ว จากนั้นทำการทดสอบนัยสำคัญด้วยสถิติทดสอบที (Paired t-test) [24-26] เพื่อเปรียบเทียบผลจากแบบจำลองว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน ในการทดสอบนัยสำคัญด้วยสถิติทดสอบทีนั้น จะทำการเปรียบแบบจำลองเป็นคู่ๆ ดังนี้

1) แบบจำลองที่ 1 เปรียบเทียบกับแบบจำลองที่ 2 เพื่อเปรียบเทียบค่าประมาณที่ได้จากแบบแบบจำลองที่ 1 ที่สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียลกับแบบจำลองที่ 2 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล ว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน ซึ่งสามารถเขียนเป็นสมมุติฐานได้คือ

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

เมื่อ μ_1 คือผลต่างระหว่างค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 กับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งคำนวณได้จาก

$\mu_1 =$ (ค่าประมาณจากแบบจำลองที่ 1) – (ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด)

μ_2 คือผลต่างระหว่างค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 กับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งคำนวณได้จาก

$\mu_2 =$ (ค่าประมาณจากแบบจำลองที่ 2) – (ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด)

2) แบบจำลองที่ 1 เปรียบเทียบกับแบบจำลองที่ 3 เพื่อเปรียบเทียบค่าประมาณที่ได้จากแบบแบบจำลองที่ 1 ที่สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียลกับแบบจำลองที่ 3 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ ว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน ซึ่งสามารถเขียนเป็นสมมุติฐานได้คือ

$$H_0: \mu_1 = \mu_3$$
$$H_1: \mu_1 > \mu_3$$

เมื่อ μ_1 คือผลต่างระหว่างค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 กับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งคำนวณได้จาก

$\mu_1 =$ (ค่าประมาณจากแบบจำลองที่ 1) – (ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด)

μ_3 คือผลต่างระหว่างค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 กับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งคำนวณได้จาก

$\mu_3 =$ (ค่าประมาณจากแบบจำลองที่ 3) – (ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด)

3) แบบจำลองที่ 2 เปรียบเทียบกับแบบจำลองที่ 3 เพื่อเปรียบเทียบค่าประมาณที่ได้จากแบบแบบจำลองที่ 2 ที่สร้างแบบจำลองโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลกับแบบจำลองที่ 3 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ ว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน ซึ่งสามารถเขียนเป็นสมมุติฐานได้คือ

$$H_0: \mu_2 = \mu_3$$
$$H_1: \mu_2 > \mu_3$$

เมื่อ μ_2 คือผลต่างระหว่างค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 กับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งคำนวณได้จาก

$\mu_2 =$ (ค่าประมาณจากแบบจำลองที่ 2) – (ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด)

μ_3 คือผลต่างระหว่างค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 กับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งคำนวณได้จาก

$\mu_3 =$ (ค่าประมาณจากแบบจำลองที่ 3) – (ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด)

3.5.2 การประเมินผลแบบจำลองโดยการหาค่าความผิดพลาดสัมพัทธ์และการทำนายที่ระดับแอล

เมื่อทำการทดสอบนัยสำคัญด้วยสถิติทดสอบทีเรียบร้อยแล้ว จากนั้นทำการประเมินผลแบบจำลอง โดยการนำชุดโปรแกรมสำหรับทดสอบมาประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด จากนั้นใช้ค่าระยะเวลาในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมาเปรียบเทียบกับค่าจริง โดยใช้การหาค่าความผิดพลาดสัมพัทธ์โดยใช้สมการที่ (23) และการทำนายที่ระดับแอล [27] ดังสมการที่ (24) ซึ่งทั้ง 2 วิธีนี้เป็นวิธีการที่ใช้ในการประเมินผลระบบการประมาณค่า โดยอธิบายรายละเอียดดังนี้

$$RE_i = \left|\frac{r_i - FRT_i}{r_i}\right| \quad (23)$$

เมื่อ RE_i คือค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ i

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i

FRT_i คือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i

$$PRED(l) = \frac{k}{n} \quad (24)$$

เมื่อ $PRED(l)$ คือการทำนายที่ระดับแอล

l คือช่วงระดับความคลาดเคลื่อน มีค่าตั้งแต่$0 - 1$

k คือจำนวนหน่วยตัวอย่างที่มีค่า $RE \leq l$

n คือจำนวนหน่วยตัวอย่างทั้งหมด

โดยผลลัพธ์ที่ได้จากการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดสามารถอธิบายได้ เช่น ถ้า $PRED(0.20) = 0.75$ หมายความว่า 75 เปอร์เซ็นต์ของค่าประมาณที่ได้จากแบบจำลองมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 20 เปอร์เซ็นต์ ซึ่งการสร้างแบบจำลองในการประมาณ โดยทั่วไปสามารถยอมรับแบบจำลองการประมาณค่าที่ผลการทำนายที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 $(PRED(0.25) \geq 0.75)$ [27] ซึ่งมีความหมายว่าค่าประมาณที่ได้จากแบบจำลองสามารถมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 75 เปอร์เซ็นต์ขึ้นไปจากการประมาณค่าทั้งหมดของแบบจำลอง

ในการสร้างแบบจำลองสำหรับการประมาณค่านั้นจะต้องมีการประเมินผลแบบจำลอง เพื่อตรวจสอบว่าแบบจำลองนั้นๆ มีความแม่นยำและสามารถยอมรับแล้วสามารถนำไปใช้ในการประมาณค่าแล้วได้ผลลัพธ์ที่ใกล้เคียงกับค่าจริง ซึ่งในงานวิจัยนี้ใช้การประเมินผลแบบจำลองและการยอมรับแบบจำลองตามมาตรฐานทั่วไปที่ได้กล่าวมาข้างต้น นอกจากนี้ยังทำการเปรียบเทียบแบบจำลองทั้ง 3 แบบจำลอง โดยนำผลลัพธ์จากการประเมินผลของแต่ละแบบจำลองมาเปรียบเทียบกัน โดยหากแบบจำลองที่มีผลการประเมินผลแบบจำลองที่ดีที่สุดหรือมีค่าความคลาดเคลื่อนน้อยที่สุด ก็จะสามารถนำแบบจำลองไปใช้ในการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ได้

บทที่ 4
การทดลองและการประเมินผล
แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

ในบทนี้จะกล่าวถึงรายละเอียดของการทดลอง เพื่อประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ตามที่ได้นำเสนอแบบจำลองทั้ง 3 แบบจำลองในบทที่ 3 โดยเริ่มจากการกล่าวถึงรายละเอียดของการเตรียมข้อมูลสำหรับทดสอบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด วัตถุประสงค์ของการทดลองประมาณค่าละประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และการประเมินผลและการเปรียบเทียบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งอธิบายรายละเอียดดังนี้

4.1 การเตรียมข้อมูลสำหรับทดสอบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

การเตรียมข้อมูลสำหรับทดสอบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยพิจารณาเลือกโปรแกรมที่มีข้อผิดพลาดมากกว่า 1 ข้อผิดพลาด โดยเป็นโปรแกรมที่มีการแจ้งข้อมูลการเกิดข้อผิดพลาดและรายละเอียดของข้อผิดพลาดเข้ามายังระบบ และผู้บำรุงรักษาทำการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว ซึ่งงานวิจัยได้รวบรวมชุดโปรแกรมสำหรับทดสอบแบบจำลองทั้งหมด 10 โปรแกรม โดยทำการรวบรวมชุดโปรแกรมที่พัฒนาด้วยภาษาจาวาจากเว็บไซต์ของโปรแกรมโอเพนซอร์ซ [22] โดยจะแสดงรายละเอียดชื่อโปรแกรม จำนวนข้อผิดพลาดของโปรแกรม และรายละเอียดของแต่ละโปรแกรมดังตารางที่ 4 จากนั้นทำการแยกประเภทของข้อผิดพลาด ซึ่งเป็นประเภทของข้อผิดพลาดที่เกิดขึ้นในรหัสต้นฉบับ โดยแบ่งเป็น 5 ประเภทคือ

1) ข้อผิดพลาดด้านตรรกะ (Logic problem)

2) ข้อผิดพลาดที่เกิดจากอัลกอริทึมบางส่วนที่อยู่ในโค้ดไม่ถูกต้อง (Computation problem)

3) ข้อผิดพลาดที่มีลักษณะของซอฟต์แวร์อินเตอร์เฟซหรือฮาร์ดแวร์อินเตอร์เฟซบางส่วนไม่สามารถทำงานได้อย่างถูกต้อง (Interface/timing problem)

4) ข้อผิดพลาดด้านการจัดการข้อมูล (Data handling problem)

5) ข้อผิดพลาดด้านข้อมูล (Data problem)

เมื่อทำการแยกประเภทข้อผิดพลาดของแต่ละโปรแกรมเสร็จเรียบร้อย ซึ่งแสดงชื่อโปรแกรมและจำนวนข้อผิดพลาดของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมดังตารางที่ 5 จากนั้นทำการคำนวณหาค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละโปรแกรมโดยใช้สมการที่ (11) ซึ่งอธิบายในหัวข้อที่ 3.3.1 และทำการคำนวณค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งในงานวิจัยนี้ใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ซึ่งอธิบายรายละเอียดในหัวข้อที่ 3.3.2 แล้วนำข้อมูลของทั้ง 10 โปรแกรมมาทำการจัดรูปแบบข้อมูลในรูปแบบตารางซึ่งสามารถแบ่งข้อมูลออกเป็น 4 ชุดข้อมูล โดยอธิบายรายละเอียดดังนี้

- ชุดข้อมูลที่ 1 คือชุดข้อมูลของแต่ละโปรแกรมที่**ไม่มีการแยกประเภทข้อผิดพลาด** ประกอบด้วยหมายเลขข้อผิดพลาดและระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด โดยชุดข้อมูลนี้จะใช้สำหรับทดสอบแบบจำลองที่ 1 และแบบจำลองที่ 2

- ชุดข้อมูลที่ 2 คือชุดข้อมูลของแต่ละโปรแกรมที่**มีการแยกประเภทข้อผิดพลาด** ประกอบด้วยหมายเลขข้อผิดพลาดและระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด โดยชุดข้อมูลนี้จะใช้สำหรับทดสอบแบบจำลองที่ 1 และแบบจำลองที่ 2

- ชุดข้อมูลที่ 3 คือชุดข้อมูลของแต่ละโปรแกรมที่**มีค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์และไม่มีการแยกประเภทข้อผิดพลาด** ประกอบด้วยหมายเลขข้อผิดพลาด, ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ โดยชุดข้อมูลนี้จะใช้สำหรับทดสอบแบบจำลองที่ 3

- ชุดข้อมูลที่ 4 คือชุดข้อมูลของแต่ละโปรแกรมที่**มีค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์และมีการแยกประเภทข้อผิดพลาด** ประกอบด้วยหมายเลขข้อผิดพลาด, ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ โดยชุดข้อมูลนี้จะใช้สำหรับทดสอบแบบจำลองที่ 3

ตารางที่ 4 ตารางแสดงรายละเอียดของแต่ละโปรแกรมที่ใช้สำหรับทดสอบแบบจำลองแบบไม่มีการแยกประเภทข้อผิดพลาด

ลำดับที่	ชื่อโปรแกรม	จำนวนข้อผิดพลาด	รายละเอียด
1	Proguard	33	โปรแกรมสำหรับพัฒนาโปรแกรมภาษาจาวา
2	Buddi	48	โปรแกรมการจัดทำงบประมาณ

ตารางที่ 4 ตารางแสดงรายละเอียดของแต่ละโปรแกรมที่ใช้สำหรับทดสอบแบบจำลองแบบไม่มีการแยกประเภทข้อผิดพลาด (ต่อ)

ลำดับที่	ชื่อโปรแกรม	จำนวนข้อผิดพลาด	รายละเอียด
3	OpenI	22	ปลั๊กอินที่สนับสนุนการเชื่อมต่อที่ใช้บริการต่างๆ ในระบบ
4	HtmlCleaner	14	โปรแกรมสำหรับแปลง HTML เป็น XML
5	DavMail	37	โปรแกรมสำหรับจัดการอีเมล์
6	ZK Spreadsheet	63	โปรแกรม spreadsheet ที่ใช้ผ่านออนไลน์
7	Escape From The Maze	30	เกม (ภาษาจาวา)
8	Java Service Wrapper	17	โปรแกรมการจัดการ บริการต่างๆ ของจาวา
9	PyDev	75	ปลั๊กอินสำหรับพัฒนาโปรแกรมภาษา Python บน Eclipse
10	Symphonic	9	โปรแกรมจัดการเพลง

ตารางที่ 5 ตารางแสดงจำนวนข้อผิดพลาดของแต่ละประเภทของแต่ละโปรแกรม

โปรแกรม / ประเภท	จำนวนข้อผิดพลาด				
	Logic	Computation	Interface/ timing	Data handling	Data
โปรแกรมที่ 1	6	-	4	15	8
โปรแกรมที่ 2	13	-	-	21	14
โปรแกรมที่ 3	11	-	-	-	11
โปรแกรมที่ 4	11	3	-	-	-
โปรแกรมที่ 5	-	16	-	12	9
โปรแกรมที่ 6	20	8	9	21	5
โปรแกรมที่ 7	2	-	-	13	15
โปรแกรมที่ 8	-	-	10	3	4
โปรแกรมที่ 9	41	-	25	9	-
โปรแกรมที่ 10	4	-	5	-	-
รวม	108	27	53	94	66

* - หมายถึงไม่พบข้อผิดพลาดในประเภทนั้น

4.2 วัตถุประสงค์ของการทดลองประมาณค่าและประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

วัตถุประสงค์ของการทดลองประมาณค่าและประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในงานวิจัยนี้คือ เพื่อตรวจสอบว่าแบบจำลองนั้นๆ มีประสิทธิภาพ ความแม่นยำและสามารถยอมรับได้ แล้วสามารถนำไปใช้ในการประมาณค่าแล้วได้ผลลัพธ์ที่ใกล้เคียงกับค่าจริง โดยจะทำการเปรียบเทียบค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด เพื่อนำไปทดสอบความถูกต้องและความแม่นยำของค่าที่ได้จากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

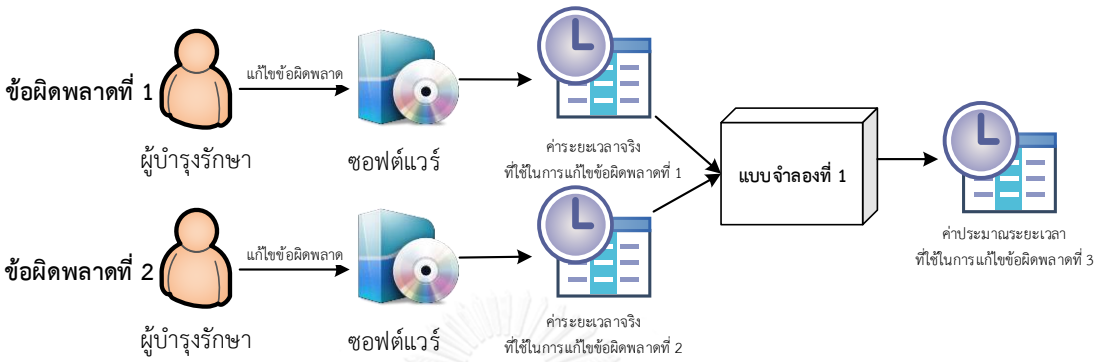
4.3 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจะใช้ชุดโปรแกรมสำหรับทดสอบแบบจำลองทั้งหมด 10 โปรแกรม ซึ่งแบ่งชุดข้อมูลเป็น 4 ชุดข้อมูลตามที่ได้กล่าวมาในหัวข้อที่ 4.1 โดยจะทำการแทนค่าในแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด จากนั้นนำค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดมาเปรียบเทียบกับค่าที่ได้จากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด เพื่อทดสอบนัยสำคัญ หาความผิดพลาดสัมพัทธ์และค่าของการทำนายที่ระดับแอลของแบบจำลองและประเมินผลแบบจำลอง และเนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งโปรแกรมนี้เป็นเกมที่พัฒนาในภาษาจาวาและมีข้อผิดพลาดทั้งหมด 30 ข้อผิดพลาด โดยจะแสดงรายละเอียดผลการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดแต่ละแบบจำลองดังนี้

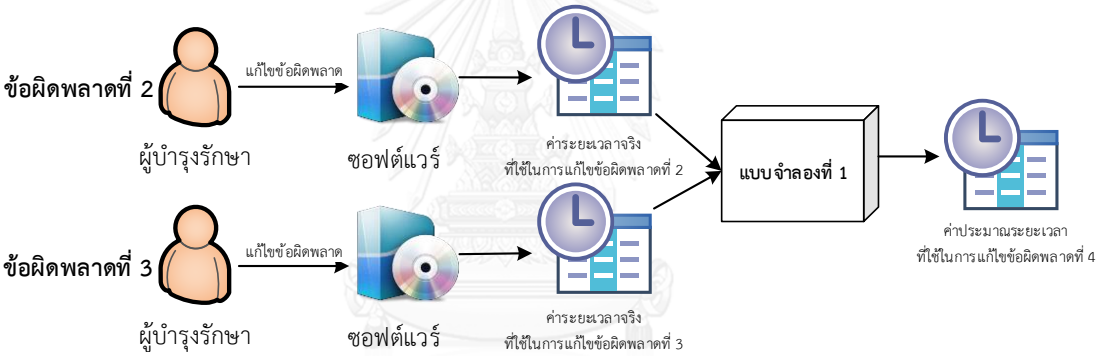
4.3.1 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล

แบบจำลองนี้สร้างโดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล ซึ่งระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นมีการแจกแจงแบบเอกซ์โปแนนเชียล ผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยในการประเมินผลแบบจำลองนั้นจะใช้ชุดข้อมูลที่ 1 คือชุดข้อมูลของแต่ละโปรแกรมที่ไม่มีการแยกประเภทข้อผิดพลาด และชุดข้อมูลที่ 2 คือชุดข้อมูลของแต่ละโปรแกรมที่มีการแยกประเภทข้อผิดพลาด และใช้ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากชุดข้อมูลทั้งสองและค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองนี้ ซึ่งอธิบายการหาค่าประมาณได้ดังนี้

ในการหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองนี้ สามารถหาได้โดยการแทนค่าในสมการที่ (15) โดยแบบจำลองนี้จะสามารถใช้ได้ก็ต่อเมื่อมีข้อผิดพลาดเกิดขึ้นแล้วอย่างน้อย 2 ข้อผิดพลาด และมีการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว ซึ่งสามารถอธิบายได้ดังรูปที่ 10 และรูปที่ 11



รูปที่ 10 การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3



รูปที่ 11 การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 4

จากรูปที่ 10 แสดงการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 ซึ่งสามารถอธิบายได้ว่าเมื่อข้อผิดพลาดที่ 1 และข้อผิดพลาดที่ 2 เกิดขึ้น และทำการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว ซึ่งจะได้ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 1 (r_1)และข้อผิดพลาดที่ 2 (r_2) เมื่อได้ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 1 และข้อผิดพลาดที่ 2 จะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) โดยแทนค่าในสมการที่ (15) และเมื่อข้อผิดพลาดที่ 3 เกิดขึ้นและถูกแก้ไขเสร็จเรียบร้อย ก็จะได้ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (r_3) ซึ่งในทำนองเดียวกันนั้นจะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 4 (FRT_4) โดยใช้ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 2 (r_2) และข้อผิดพลาดที่ 3 (r_3) แทนค่าในสมการที่ (15) ซึ่งแสดงได้ดังรูปที่ 11 เมื่อข้อผิดพลาดที่ 3 ถูกแก้ไขเสร็จเรียบร้อย ดังนั้นก็จะทำให้ทราบระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (r_3) และค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3)

ตัวอย่างการหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) โดยระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 1 (r_1) มีค่าเป็น 118 วัน และระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 2 (r_2) มีค่าเป็น 166 วัน เมื่อเกิดข้อผิดพลาดที่ 3 เกิดขึ้น จะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้ดังนี้

จากสมการที่ (15) ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดสามารถคำนวณได้จาก

$$FRT_i = \frac{r_{i-2} + r_{i-1}}{2}$$

ดังนั้นสามารถคำนวณค่าประมาณที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) ได้ดังสมการที่ (25)

$$FRT_3 = \frac{r_1 + r_2}{2} = \frac{118 + 166}{2} = 142 \qquad (25)$$

ตารางที่ 6 ตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1

หมายเลขข้อผิดพลาด	r	FRT	**หมายเลขข้อผิดพลาด**	r	FRT
1	118	-	16	100	104.50
2	166	-	17	107	103.50
3	166	142.00	18	100	103.50
4	166	166.00	19	115	103.50
5	168	166.00	20	115	107.50
6	169	167.00	21	115	115.00
7	170	168.50	22	121	115.00
8	114	169.50	23	103	118.00
9	116	142.00	24	132	112.00
10	137	115.00	25	128	117.50
11	142	126.50	26	112	130.00
12	34	139.50	27	127	120.00
13	36	88.00	28	121	119.50
14	102	35.00	29	120	124.00
15	107	69.00	30	117	120.50

จากตารางที่ 6 แสดงตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งโปรแกรมนี้เป็นเกมที่พัฒนาในภาษาจาวาและมีข้อผิดพลาดทั้งหมด 30 ข้อผิดพลาด ซึ่งเมื่อทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองทั้งหมด 30 ข้อผิดพลาดแล้ว

4.3.2 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล

แบบจำลองนี้สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล โดยผลลัพธ์ที่ได้จากแบบจำลองนี้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยในการประเมินผลแบบจำลองนั้นจะใช้ชุดข้อมูลที่ 1 คือชุดข้อมูลของแต่ละโปรแกรมที่ไม่มีการแยกประเภทข้อผิดพลาด และชุดข้อมูลที่ 2 คือชุดข้อมูลของแต่ละโปรแกรมที่มีการแยกประเภทข้อผิดพลาด ซึ่งอธิบายการหาค่าประมาณได้ดังนี้

ในการหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองนี้ สามารถหาได้โดยการแทนค่าในสมการที่ (18) ซึ่งแสดงตัวอย่างในการหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 ดังนี้

จากสมการที่ (18) ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดสามารถคำนวณได้จาก

$$FRT_i = \frac{1}{\lambda_i} = \frac{1}{[N-(i-1)]\phi}$$

ดังนั้นสามารถคำนวณค่าประมาณที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) เมื่อจำนวนข้อผิดพลาดมีค่าเป็น 30 (N = 30) และค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดมีค่าเป็น 0.00023 (ϕ = 0.00023) ซึ่งแสดงได้ดังสมการที่ (26)

$$FRT_3 = \frac{1}{[30-(3-1)](0.00023)} = 154.79 \qquad (26)$$

เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งโปรแกรมนี้เป็นเกมที่พัฒนาในภาษาจาวาและมีข้อผิดพลาดทั้งหมด 30 ข้อผิดพลาด ซึ่งเมื่อทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองทั้งหมด 30 ข้อผิดพลาดแล้ว จะแสดงได้ดังตัวอย่างในตารางที่ 7

ตารางที่ 7 ตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2

หมายเลข ข้อผิดพลาด	r	FRT	**หมายเลข ข้อผิดพลาด**	r	FRT
1	118	118.00	16	100	197.03
2	166	144.03	17	107	204.98
3	166	154.79	18	100	214.04
4	166	161.89	19	115	225.73
5	168	168.09	20	115	239.69
6	169	173.85	21	115	256.58
7	170	179.51	22	121	277.63
8	114	178.15	23	103	303.23
9	116	178.44	24	132	337.61
10	137	181.94	25	128	383.24
11	142	186.58	26	112	446.51
12	34	182.87	27	127	542.17
13	36	180.95	28	121	701.39
14	102	185.19	29	120	1019.95
15	107	190.78	30	117	1975.80

4.3.3 การทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์

แบบจำลองนี้สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ โดยผลลัพธ์ที่ได้จากแบบจำลองนี้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยในการประเมินผลแบบจำลองนั้นจะใช้ชุดข้อมูลที่ 3 คือชุดข้อมูลของแต่ละโปรแกรมที่มีค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์และไม่มีการแยกประเภทข้อผิดพลาด และชุดข้อมูลที่ 4 คือชุดข้อมูลของแต่ละโปรแกรมที่มีค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์และมีการแยกประเภทข้อผิดพลาด ซึ่งอธิบายการหาค่าประมาณได้ดังนี้

ในการหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองนี้ สามารถหาได้โดยการแทนค่าในสมการที่ (21) ซึ่งแสดงตัวอย่างในการหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 ดังนี้

จากสมการที่ (21) ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดสามารถคำนวณได้จาก

$$FRT_i = \frac{1}{\lambda_i} = \frac{1}{[N-(i-1)(Instability)]\phi}$$

ดังนั้นสามารถคำนวณค่าประมาณที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) เมื่อจำนวนข้อผิดพลาดมีค่าเป็น 30 (N = 30), ค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดมีค่าเป็น 0.00022 (ϕ = 0.00022) และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์มีค่าเป็น 0 ($Instability$ = 0) ซึ่งแสดงได้ดังสมการที่ (27)

$$FRT_3 = \frac{1}{[30-(3-1)(0)](0.00022)} = 150 \qquad (27)$$

ตารางที่ 8 ตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3

หมายเลขข้อผิดพลาด	r	FRT	หมายเลขข้อผิดพลาด	r	FRT
1	118	118.00	16	100	125.69
2	166	142.00	17	107	124.59
3	166	150.00	18	100	123.22
4	166	154.00	19	115	122.79
5	168	156.80	20	115	122.40
6	169	158.83	21	115	122.05
7	170	160.43	22	121	122.00
8	114	154.63	23	103	121.17
9	116	150.33	24	132	121.63
10	137	149.00	25	128	121.88
11	142	148.36	26	112	121.50
12	34	138.83	27	127	121.70
13	36	130.92	28	121	121.68
14	102	128.86	29	120	121.62
15	107	127.40	30	117	121.47

จากตารางที่ 8 แสดงตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งโปรแกรมนี้เป็นเกมที่พัฒนาในภาษาจาวาและมีข้อผิดพลาดทั้งหมด 30 ข้อผิดพลาด ซึ่งเมื่อทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองทั้งหมด 30 ข้อผิดพลาดแล้ว

4.4 การประเมินผลและการเปรียบเทียบแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

4.4.1 การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบที

จากการทดลองประมาณค่าจากแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดทั้ง 3 แบบจำลอง ในงานวิจัยนี้ได้ทำการประเมินผลแบบจำลองโดยใช้ชุดโปรแกรมสำหรับทดสอบแบบจำลองทั้งหมด 10 โปรแกรม รายละเอียดแต่ละโปรแกรมได้ระบุไว้ตารางที่ 4 ในการเปรียบเทียบแบบจำลองนั้นจะใช้การทดสอบนัยสำคัญด้วยสถิติทดสอบที โดยใช้ Paired t-test [26] ที่ระดับความเชื่อมั่น 95% เพื่อเปรียบเทียบผลจากแบบจำลองว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน ในการทดสอบนัยสำคัญด้วยสถิติทดสอบทีนั้น จะทำการเปรียบแบบจำลองเป็นคู่ๆ เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการเปรียบเทียบแบบจำลอง โดยสามารถดูรายละเอียดการเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมทั้งหมดได้ในภาคผนวก ง โปรแกรมที่ได้ยกตัวอย่างนี้คือโปรแกรมที่ 7 ซึ่งเป็นเกมที่พัฒนาในภาษาจาวาและมีข้อผิดพลาดทั้งหมด 30 ข้อผิดพลาดโดยผลการทดสอบนัยสำคัญด้วยสถิติทดสอบทีสามารถแสดงรายละเอียดได้ดังนี้

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-222.164	375.32396662	70.92956	-367.699	-76.6285	-3.132	27	.004
Pair 2	Model 1 - Model 3	-11.9922	23.22894806	4.389859	-20.9994	-2.98493	-2.732	27	.011
Pair 3	Model 2 - Model 3	196.2282	369.51826710	67.46450	58.24776	334.2085	2.909	29	.007

รูปที่ 12 ผลการเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7

จากรูปที่ 12 แสดงผลการเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

1) ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า

- ค่า t มีค่าเป็น -3.132
- ค่า Mean มีค่าเป็น -222.164
- ค่า Sig.(2-tailed) มีค่าเป็น 0.004
- ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.004}{2} = 0.002$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

2) ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า

- ค่า t มีค่าเป็น -2.732
- ค่า Mean มีค่าเป็น -11.9922
- ค่า Sig.(2-tailed) มีค่าเป็น 0.011
- ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.011}{2} = 0.0055$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

3) ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า

- ค่า t มีค่าเป็น 2.909
- ค่า Mean มีค่าเป็น 196.2282
- ค่า Sig.(2-tailed) มีค่าเป็น 0.007
- ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.007}{2} = 0.0035$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

Chulalongkorn University

เมื่อทำการเปรียบเทียบแบบจำลองทั้งสามแบบจำลองด้วยสถิติทดสอบทีกับ 10 โปรแกรมซึ่งจะแสดงในภาคผนวก ง โดยแสดงการเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของทั้ง 10 โปรแกรม ซึ่งเมื่อเปรียบเทียบทั้งหมดแล้วพบว่า แบบจำลองที่ 1 ที่สร้างด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียลมีประสิทธิภาพมากที่สุด ในขณะที่แบบจำลองที่ 3 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์นั้นมีประสิทธิภาพเป็นอันดับที่สอง และแบบจำลองที่ 2 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล มีประสิทธิภาพน้อยที่สุด

4.4.2 การประเมินผลแบบจำลองโดยการคำนวณค่าความผิดพลาดสัมพัทธ์และการทำนายที่ระดับแอล

เมื่อทำการทดสอบนัยสำคัญด้วยสถิติทดสอบทีเสร็จเรียบร้อย จากนั้นทำการประเมินผลแต่ละแบบจำลอง โดยจะใช้ค่าความผิดพลาดสัมพัทธ์ที่ได้จากแต่ละแบบจำลองของแต่ละโปรแกรม มาคำนวณหาค่าการทำนายที่ระดับแอล ซึ่งในงานวิจัยนี้ใช้การทำนายที่ระดับ 0.25 ซึ่งเป็นมาตรฐานโดยทั่วไปว่าสามารถยอมรับแบบจำลองการประมาณค่าที่ผลการทำนายที่ระดับ 0.25 จะต้องมากกว่าหรือเท่ากับ 0.75 หรือ 75 เปอร์เซ็นต์ของค่าประมาณที่ได้จากแบบจำลองสามารถมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์

4.4.2.1 ผลลัพธ์ของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล

แบบจำลองนี้สร้างโดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล ซึ่งผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยในการประเมินผลแบบจำลองนั้นจะใช้ชุดข้อมูลที่ 1 คือชุดข้อมูลของแต่ละโปรแกรมที่ไม่มีการแยกประเภทข้อผิดพลาด และชุดข้อมูลที่ 2 คือชุดข้อมูลของแต่ละโปรแกรมที่มีการแยกประเภทข้อผิดพลาด และใช้ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากชุดข้อมูลทั้งสองและค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองนี้ ซึ่งการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ได้กล่าวไว้ในหัวข้อที่ 4.3.1 เมื่อได้ค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ทั้งหมดแล้ว จากนั้นนำมาคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลโดยมีแสดงตัวอย่างการคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลดังนี้

เมื่อข้อผิดพลาดที่ 3 เกิดขึ้น และผู้บำรุงรักษาทำการแก้ไขข้อข้อผิดพลาดที่ 3 เสร็จเรียบร้อยแล้ว จะทำให้ทราบระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (r_3) ซึ่งมีค่าเป็น 166 วัน และค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) ของแบบจำลองที่ 1 ที่ได้กล่าวไว้ในหัวข้อที่ 4.3.1 มีค่าเป็น 142 ดังนั้นสามารถหาค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ 3 ได้ดังนี้

จากสมการที่ (23) ค่าความผิดพลาดสัมพัทธ์สามารถคำนวณได้จาก

$$RE_i = \left|\frac{r_i - FRT_i}{r_i}\right|$$

ดังนั้นสามารถหาค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ 3 (RE_3) ได้ดังสมการที่ (28)

$$RE_3 = \left|\frac{r_3 - FRT_3}{r_3}\right| = \left|\frac{166 - 142}{166}\right| = 0.1446 \quad (28)$$

ตารางที่ 9 ตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 1

หมายเลขข้อผิดพลาด	r	FRT	RE	**หมายเลขข้อผิดพลาด**	r	FRT	RE
1	118	-	-	16	100	104.50	**0.0450**
2	166	-	-	17	107	103.50	**0.0327**
3	166	142.00	**0.1446**	18	100	103.50	**0.0350**
4	166	166.00	**0.0000**	19	115	103.50	**0.1000**
5	168	166.00	**0.0119**	20	115	107.50	**0.0652**
6	169	167.00	**0.0118**	21	115	115.00	**0.0000**
7	170	168.50	**0.0088**	22	121	115.00	**0.0496**
8	114	169.50	0.4868	23	103	118.00	**0.1456**
9	116	142.00	**0.2241**	24	132	112.00	**0.1515**
10	137	115.00	**0.1606**	25	128	117.50	**0.0820**
11	142	126.50	**0.1092**	26	112	130.00	**0.1607**
12	34	139.50	3.1029	27	127	120.00	**0.0551**
13	36	88.00	1.4444	28	121	119.50	**0.0124**
14	102	35.00	0.6569	29	120	124.00	**0.0333**
15	107	69.00	0.3551	30	117	120.50	**0.0299**

จากตารางที่ 9 แสดงตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 1 เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งเมื่อทำการคำนวณค่าความผิดพลาดสัมพัทธ์ของโปรแกรมนี้ทั้งหมด 30 ข้อผิดพลาดแล้ว โดยเมื่อคำนวณค่าความผิดพลาดสัมพัทธ์ของทั้งโปรแกรมเสร็จเรียบร้อยแล้ว จะทำการคำนวณค่าการทำนายที่ระดับแอล ($PRED(l)$) จากที่กล่าวมาในหัวข้อที่ 3.5 ว่าโดยทั่วไปสามารถยอมรับแบบจำลองการประมาณค่าที่ผลการทำนายที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 ($PRED(0.25) \geq 0.75$) ซึ่งมีความหมายว่าค่าประมาณที่ได้จากแบบจำลองสามารถมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 75 เปอร์เซ็นต์ขึ้นไปจากการประมาณค่าทั้งหมดของแบบจำลอง ซึ่งการคำนวณค่า $PRED(0.25)$ สามารถอธิบายได้ดังนี้

จากสมการที่ (24) การทำนายที่ระดับแอลสามารถคำนวณได้จาก

$$PRED(l) = \frac{k}{n}$$

เมื่อ $PRED(l)$ คือการทำนายที่ระดับแอล

l คือช่วงระดับความคลาดเคลื่อน มีค่าตั้งแต่$0 - 1$

k คือจำนวนหน่วยตัวอย่างที่มีค่า $RE \leq l$

n คือจำนวนหน่วยตัวอย่างทั้งหมด

ดังนั้นสามารถคำนวณค่า $PRED(0.25)$ ของโปรแกรมตัวอย่างนี้ได้ ซึ่งแสดงตัวอย่างดังสมการที่ (29) โดยที่ k มีค่าเป็น 23 โดยสามารถคำนวณจากค่าความผิดพลาดสัมพัทธ์ของแต่ละข้อผิดพลาดที่มีค่าน้อยกว่าหรือเท่ากับ 0.25 ($RE \leq 0.25$) และ n คือจำนวนข้อผิดพลาดทั้งหมดซึ่งมีค่าเป็น 30

$$PRED(0.25) = \frac{23}{30} = 0.7667 = 76.67\% \quad (29)$$

นั่นหมายความว่าค่าประมาณที่ได้จากแบบจำลองมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์ เป็นจำนวน 76.67 เปอร์เซ็นต์จากการประมาณค่าทั้งหมดของแบบจำลอง

เมื่อทำการประเมินผลแบบจำลองที่สร้างโดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียลกับชุดข้อมูลที่ 1 ผลการประเมินผลแบบจำลองสามารถแสดงได้ดังตารางที่ 10 จากนั้นทำการประเมินผลแบบจำลองที่สร้างโดยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียลกับชุดข้อมูลที่ 2 ซึ่งผลการประเมินผลแบบจำลองสามารถแสดงได้ดังตารางที่ 11

ตารางที่ 10 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 ของแต่ละโปรแกรมในชุดข้อมูลที่ 1

โปรแกรม	$PRED(0.25)$
โปรแกรมที่ 1	57.58%
โปรแกรมที่ 2	22.92%
โปรแกรมที่ 3	27.27%
โปรแกรมที่ 4	35.71%
โปรแกรมที่ 5	16.22%
โปรแกรมที่ 6	50.79%
โปรแกรมที่ 7	76.67%
โปรแกรมที่ 8	5.88%
โปรแกรมที่ 9	20.00%
โปรแกรมที่ 10	11.11%

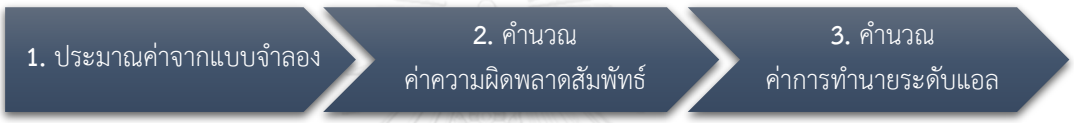
ตารางที่ 11 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 ของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมของชุดข้อมูลที่ 2

โปรแกรม / ประเภท	$PRED(0.25)$				
	Logic	**Computation**	**Interface/ timing**	**Data handling**	**Data**
โปรแกรมที่ 1	16.67%	-	50.00%	60.00%	75.00%
โปรแกรมที่ 2	30.77%	-	-	28.57%	28.57%
โปรแกรมที่ 3	9.09%	-	-	-	18.18%
โปรแกรมที่ 4	72.73%	0.00%	-	-	-
โปรแกรมที่ 5	-	50.00%	-	66.67%	55.56%
โปรแกรมที่ 6	60.00%	37.50%	22.22%	57.14%	60.00%
โปรแกรมที่ 7	0.00%	-	-	76.92%	80.00%
โปรแกรมที่ 8	-	-	0.00%	33.33%	50.00%
โปรแกรมที่ 9	19.51%	-	56.00%	77.78%	-
โปรแกรมที่ 10	50.00%	-	0.00%	-	-

* - หมายถึงไม่พบข้อผิดพลาดในประเภทนั้น

4.4.2.2 ผลลัพธ์ของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล

แบบจำลองนี้สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล ซึ่งผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยในการประเมินผลแบบจำลองนั้นจะใช้ชุดข้อมูลที่ 1 คือชุดข้อมูลของแต่ละโปรแกรมที่ไม่มีการแยกประเภทข้อผิดพลาด และชุดข้อมูลที่ 2 คือชุดข้อมูลของแต่ละโปรแกรมที่มีการแยกประเภทข้อผิดพลาด และใช้ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากชุดข้อมูลทั้งสองและค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองนี้ ซึ่งการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ได้กล่าวไว้ในหัวข้อที่ 4.3.2 ซึ่งแสดงภาพเป็นลำดับขั้นตอนตั้งแต่การหาค่าประมาณ การคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลได้ดังรูปที่ 13



รูปที่ 13 ลำดับขั้นตอนของการประเมินผลแบบจำลองที่ 2

เมื่อได้ค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ทั้งหมดแล้ว จากนั้นนำมาคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลโดยมีแสดงตัวอย่างการคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลดังนี้

เมื่อข้อผิดพลาดที่ 3 เกิดขึ้น และผู้บำรุงรักษาทำการแก้ไขข้อข้อผิดพลาดที่ 3 เสร็จเรียบร้อยแล้ว จะทำให้ทราบระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (r_3) ซึ่งมีค่าเป็น 166 วัน และค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) ของแบบจำลองที่ 2 ที่ได้กล่าวไว้ในหัวข้อที่ 4.3.2 มีค่าเป็น 154.79 ดังนั้นสามารถหาค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ 3 ได้ดังนี้

จากสมการที่ (23) ค่าความผิดพลาดสัมพัทธ์สามารถคำนวณได้จาก

$$RE_i = \left|\frac{r_i - FRT_i}{r_i}\right|$$

ดังนั้นสามารถหาค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ 3 (RE_3) ได้ดังสมการที่ (30)

$$RE_3 = \left|\frac{r_3 - FRT_3}{r_3}\right| = \left|\frac{166 - 154.79}{166}\right| = 0.0676 \qquad (30)$$

เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งเมื่อทำการคำนวณค่าความผิดพลาดสัมพัทธ์ของโปรแกรมนี้ทั้งหมด 30 ข้อผิดพลาดแล้ว จะแสดงได้ดังตัวอย่างในตารางที่ 12

ตารางที่ 12 ตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 2

หมายเลขข้อผิดพลาด	r	FRT	RE	**หมายเลขข้อผิดพลาด**	r	FRT	RE
1	118	118.00	**0.0000**	16	100	197.03	0.9703
2	166	144.03	**0.1323**	17	107	204.98	0.9157
3	166	154.79	**0.0676**	18	100	214.04	1.1404
4	166	161.89	**0.0248**	19	115	225.73	0.9629
5	168	168.09	**0.0005**	20	115	239.69	1.0842
6	169	173.85	**0.0287**	21	115	256.58	1.2311
7	170	179.51	**0.0559**	22	121	277.63	1.2944
8	114	178.15	0.5627	23	103	303.23	1.9440
9	116	178.44	0.5383	24	132	337.61	1.5576
10	137	181.94	0.3281	25	128	383.24	1.9941
11	142	186.58	0.3140	26	112	446.51	2.9867
12	34	182.87	4.3785	27	127	542.17	3.2690
13	36	180.95	4.0264	28	121	701.39	4.7966
14	102	185.19	0.8156	29	120	1019.95	7.4996
15	107	190.78	0.7830	30	117	1975.80	15.8872

จากตารางที่ 12 แสดงตัวอย่างประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 2 โดยเมื่อคำนวณค่าความผิดพลาดสัมพัทธ์ของทั้งโปรแกรมเสร็จเรียบร้อยแล้ว จะทำการคำนวณค่าการทำนายที่ระดับแอล ($PRED(l)$) จากที่กล่าวมาในหัวข้อที่ 3.5 ว่าโดยทั่วไปสามารถยอมรับแบบจำลองการประมาณค่าที่ผลการทำนายที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 ($PRED(0.25) \geq 0.75$) ซึ่งมีความหมายว่าค่าประมาณที่ได้จากแบบจำลองสามารถมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 75 เปอร์เซ็นต์ขึ้นไปจากการประมาณค่าทั้งหมดของแบบจำลอง ซึ่งการคำนวณค่า $PRED(0.25)$ สามารถอธิบายได้ดังนี้

จากสมการที่ (24) การทำนายที่ระดับแอลสามารถคำนวณได้จาก

$$PRED(l) = \frac{k}{n}$$

เมื่อ $PRED(l)$ คือการทำนายที่ระดับแอล

l คือช่วงระดับความคลาดเคลื่อน มีค่าตั้งแต่$0 - 1$

k คือจำนวนหน่วยตัวอย่างที่มีค่า $RE \leq l$

n คือจำนวนหน่วยตัวอย่างทั้งหมด

ดังนั้นสามารถคำนวณค่า $PRED(0.25)$ ของโปรแกรมตัวอย่างนี้ได้ ซึ่งแสดงตัวอย่างดังสมการที่ (31) โดยที่ k มีค่าเป็น 7 โดยสามารถคำนวณจากค่าความผิดพลาดสัมพัทธ์ของแต่ละข้อผิดพลาดที่มีค่าน้อยกว่าหรือเท่ากับ 0.25 ($RE \leq 0.25$) และ n คือจำนวนข้อผิดพลาดทั้งหมดซึ่งมีค่าเป็น 30

$$PRED(0.25) = \frac{7}{30} = 0.2333 = 23.33\% \quad (31)$$

นั่นหมายความว่าค่าประมาณที่ได้จากแบบจำลองมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 23.33 เปอร์เซ็นต์จากการประมาณค่าทั้งหมดของแบบจำลอง

เมื่อทำการประเมินผลแบบจำลองที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลกับชุดข้อมูลที่ 1 ผลการประเมินผลแบบจำลองสามารถแสดงได้ดังตารางที่ 13 จากนั้นทำการประเมินผลการแบบจำลองที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียลกับชุดข้อมูลที่ 2 ซึ่งผลการประเมินผลแบบจำลองสามารถแสดงได้ดังตารางที่ 14

ตารางที่ 13 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2 ของแต่ละโปรแกรมในชุดข้อมูลที่ 1

โปรแกรม	$PRED(0.25)$
โปรแกรมที่ 1	18.18%
โปรแกรมที่ 2	29.17%
โปรแกรมที่ 3	13.64%
โปรแกรมที่ 4	35.71%
โปรแกรมที่ 5	21.62%
โปรแกรมที่ 6	17.46%
โปรแกรมที่ 7	23.33%
โปรแกรมที่ 8	11.76%
โปรแกรมที่ 9	10.67%
โปรแกรมที่ 10	22.22%

ตารางที่ 14 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2 ของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมของชุดข้อมูลที่ 2

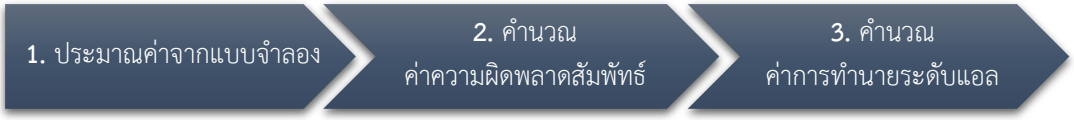
โปรแกรม / ประเภท	$PRED(0.25)$				
	Logic	Computation	Interface/ timing	Data handling	Data
โปรแกรมที่ 1	50.00%	-	50.00%	40.00%	37.50%
โปรแกรมที่ 2	38.46%	-	-	14.29%	28.57%
โปรแกรมที่ 3	18.18%	-	-	-	27.27%
โปรแกรมที่ 4	18.18%	33.33%	-	-	-
โปรแกรมที่ 5	-	37.50%	-	58.33%	22.22%
โปรแกรมที่ 6	25.00%	50.00%	33.33%	28.57%	40.00%
โปรแกรมที่ 7	50.00%	-	-	30.77%	20.00%
โปรแกรมที่ 8	-	-	30.00%	66.67%	50.00%
โปรแกรมที่ 9	19.51%	-	24.00%	44.44%	-
โปรแกรมที่ 10	50.00%	-	20.00%	-	-

* - หมายถึงไม่พบข้อผิดพลาดในประเภทนั้น

4.4.2.3 ผลลัพธ์ของการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

แบบจำลองนี้สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ ซึ่งผลลัพธ์ที่ได้จะเป็นค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยประมาณ โดยในการประเมินผลแบบจำลองนั้นจะใช้ชุดข้อมูลที่ 3 คือชุดข้อมูลของแต่ละโปรแกรมที่มีค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์และไม่มีการแยกประเภทข้อผิดพลาด และชุดข้อมูลที่ 4 คือชุดข้อมูลของแต่ละโปรแกรมที่มีค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์และมีการแยกประเภทข้อผิดพลาด และใช้ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากชุดข้อมูลทั้งสองและค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองนี้ ซึ่งการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ได้กล่าวไว้ในหัวข้อที่ 4.3.3 ซึ่งแสดงภาพเป็นลำดับขั้นตอนตั้งแต่การหาค่าประมาณ การคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลได้ดังรูปที่ 14



รูปที่ 14 ลำดับขั้นตอนของการประเมินผลแบบจำลองที่ 3

เมื่อได้ค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ทั้งหมดแล้ว จากนั้นนำมาคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลโดยมีแสดงตัวอย่างการคำนวณค่าความผิดพลาดสัมพัทธ์และการคำนวณค่าการทำนายที่ระดับแอลดังนี้

เมื่อข้อผิดพลาดที่ 3 เกิดขึ้น และผู้บำรุงรักษาทำการแก้ไขข้อข้อผิดพลาดที่ 3 เสร็จเรียบร้อยแล้ว จะทำให้ทราบระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (r_3) ซึ่งมีค่าเป็น 166 วัน และค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ 3 (FRT_3) ของแบบจำลองที่ 3 ที่ได้กล่าวไว้ในหัวข้อที่ 4.3.3 มีค่าเป็น 150 ดังนั้นสามารถหาค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ 3 ได้ดังนี้

จากสมการที่ (23) ค่าความผิดพลาดสัมพัทธ์สามารถคำนวณได้จาก

$$RE_i = \left|\frac{r_i - FRT_i}{r_i}\right|$$

ดังนั้นสามารถหาค่าความผิดพลาดสัมพัทธ์ของข้อผิดพลาดที่ 3 (RE_3) ได้ดังสมการที่ (32)

$$RE_3 = \left|\frac{r_3 - FRT_3}{r_3}\right| = \left|\frac{166 - 150}{166}\right| = 0.0964 \quad (32)$$

เนื่องจากข้อมูลที่ใช้ในการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีจำนวนมาก ผู้วิจัยจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบถึงรายละเอียดของการประเมินผลแบบจำลอง ซึ่งเมื่อทำการคำนวณค่าความผิดพลาดสัมพัทธ์ของโปรแกรมนี้ทั้งหมด 30 ข้อผิดพลาดแล้ว จะแสดงได้ดังตัวอย่างในตารางที่ 15 ซึ่งแสดงตัวอย่างการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดและการประเมินผลแบบจำลองที่ 3 โดยการคำนวณค่าความผิดพลาดสัมพัทธ์ โดยเมื่อคำนวณค่าความผิดพลาดสัมพัทธ์ของทั้งโปรแกรมเสร็จเรียบร้อยแล้ว จะทำการคำนวณค่าการทำนายที่ระดับแอล ($PRED(l)$) จากที่กล่าวมาในหัวข้อที่ 3.5 ว่าโดยทั่วไปสามารถยอมรับแบบจำลองการประมาณค่าที่ผลการทำนายที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 ($PRED(0.25) \geq 0.75$) ซึ่งมีความหมายว่าค่าประมาณที่ได้จากแบบจำลองสามารถมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 75 เปอร์เซ็นต์ขึ้นไปจากการประมาณค่าทั้งหมดของแบบจำลอง ซึ่งการคำนวณค่า $PRED(0.25)$ สามารถอธิบายได้ดังนี้

ตารางที่ 15 ตัวอย่างการประเมินผลโดยการคำนวณค่าความผิดพลาดสัมพัทธ์ของแบบจำลองที่ 3

หมายเลขข้อผิดพลาด	r	FRT	RE	หมายเลขข้อผิดพลาด	r	FRT	RE
1	118	118.00	**0.0000**	16	100	125.69	0.2569
2	166	142.00	**0.1446**	17	107	124.59	**0.1644**
3	166	150.00	**0.0964**	18	100	123.22	**0.2322**
4	166	154.00	**0.0723**	19	115	122.79	**0.0677**
5	168	156.80	**0.0667**	20	115	122.40	**0.0643**
6	169	158.83	**0.0602**	21	115	122.05	**0.0613**
7	170	160.43	**0.0563**	22	121	122.00	**0.0083**
8	114	154.63	0.3564	23	103	121.17	**0.1764**
9	116	150.33	0.2960	24	132	121.63	**0.0786**
10	137	149.00	**0.0876**	25	128	121.88	**0.0478**
11	142	148.36	**0.0448**	26	112	121.50	**0.0848**
12	34	138.83	3.0833	27	127	121.70	**0.0417**
13	36	130.92	2.6368	28	121	121.68	**0.0056**
14	102	128.86	0.2633	29	120	121.62	**0.0135**
15	107	127.40	**0.1907**	30	117	121.47	**0.0382**

จากสมการที่ (24) การทำนายที่ระดับแอลสามารถคำนวณได้จาก

$$PRED(l) = \frac{k}{n}$$

เมื่อ $PRED(l)$ คือการทำนายที่ระดับแอล

l คือช่วงระดับความคลาดเคลื่อน มีค่าตั้งแต่$0 - 1$

k คือจำนวนหน่วยตัวอย่างที่มีค่า $RE \leq l$

n คือจำนวนหน่วยตัวอย่างทั้งหมด

ดังนั้นสามารถคำนวณค่า $PRED(0.25)$ ของโปรแกรมตัวอย่างนี้ได้ ซึ่งแสดงตัวอย่างดังสมการที่ (33) โดยที่ k มีค่าเป็น 24 โดยสามารถคำนวณจากค่าความผิดพลาดสัมพัทธ์ของแต่ละข้อผิดพลาดที่มีค่าน้อยกว่าหรือเท่ากับ 0.25 ($RE \leq 0.25$) และ n คือจำนวนข้อผิดพลาดทั้งหมดซึ่งมีค่าเป็น 30

$$PRED(0.25) = \frac{24}{30} = 0.8000 = 80.00\% \quad (33)$$

นั่นหมายความว่าค่าประมาณที่ได้จากแบบจำลองมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 80.00 เปอร์เซ็นต์จากการประมาณค่าทั้งหมดของแบบจำลอง

ตารางที่ 16 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 3 ของแต่ละโปรแกรมในชุดข้อมูลที่ 3

โปรแกรม	$PRED(0.25)$
โปรแกรมที่ 1	42.42%
โปรแกรมที่ 2	29.17%
โปรแกรมที่ 3	31.82%
โปรแกรมที่ 4	35.71%
โปรแกรมที่ 5	24.32%
โปรแกรมที่ 6	15.87%
โปรแกรมที่ 7	80.00%
โปรแกรมที่ 8	23.53%
โปรแกรมที่ 9	16.00%
โปรแกรมที่ 10	44.44%

ตารางที่ 17 ค่าการทำนายที่ระดับแอลของแบบจำลองที่ 3 ของแต่ละประเภทข้อผิดพลาดในแต่ละโปรแกรมของชุดข้อมูลที่ 4

โปรแกรม / **ประเภท**	$PRED(0.25)$				
	Logic	**Computation**	**Interface/ timing**	**Data handling**	**Data**
โปรแกรมที่ 1	66.67%	-	75.00%	46.67%	62.50%
โปรแกรมที่ 2	46.15%	-	-	23.81%	42.86%
โปรแกรมที่ 3	27.27%	-	-	-	27.27%
โปรแกรมที่ 4	18.18%	33.33%	-	-	-
โปรแกรมที่ 5	-	37.50%	-	58.33%	22.22%
โปรแกรมที่ 6	30.00%	50.00%	33.33%	33.33%	60.00%
โปรแกรมที่ 7	100.00%	-	-	92.31%	60.00%
โปรแกรมที่ 8	-	-	40.00%	66.67%	50.00%
โปรแกรมที่ 9	31.71%	-	32.00%	44.44%	-
โปรแกรมที่ 10	75.00%	-	40.00%	-	-

* - หมายถึงไม่พบข้อผิดพลาดในประเภทนั้น

เมื่อทำการประเมินผลแบบจำลองที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ กับชุดข้อมูลที่ 3 ผลการประเมินผลแบบจำลองสามารถแสดงได้ดังตารางที่ 16 จากนั้นทำการประเมินผลการแบบจำลองสร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ กับชุดข้อมูลที่ 4 ซึ่งผลการประเมินผลแบบจำลองสามารถแสดงได้ดังตารางที่ 17

4.4.3 การเปรียบเทียบการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

เมื่อทำการประเมินผลแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดทั้ง 3 แบบจำลองเสร็จเรียบร้อยแล้ว ในกรณีที่ไม่ได้พิจารณาประเภทของข้อผิดพลาดจะได้ผลลัพธ์การประเมินผลของแบบจำลองทั้ง 3 แบบจำลอง ดังตารางที่ 18 ในกรณีที่พิจารณาประเภทของข้อผิดพลาดจะได้ผลลัพธ์การประเมินผลของแบบจำลองทั้ง 3 แบบจำลองดังตารางที่ 19

ตารางที่ 18 การเปรียบเทียบการประเมินผลแบบจำลองทั้งสามแบบจำลอง ในกรณีไม่ได้พิจารณาประเภทของข้อผิดพลาด

โปรแกรม	$PRED(0.25)$		
	แบบจำลองที่ 1	**แบบจำลองที่ 2**	**แบบจำลองที่ 3**
โปรแกรมที่ 1	57.58%	18.18%	42.42%
โปรแกรมที่ 2	22.92%	29.17%	29.17%
โปรแกรมที่ 3	27.27%	13.64%	31.82%
โปรแกรมที่ 4	35.71%	35.71%	35.71%
โปรแกรมที่ 5	16.22%	21.62%	24.32%
โปรแกรมที่ 6	50.79%	17.46%	15.87%
โปรแกรมที่ 7	76.67%	23.33%	80.00%
โปรแกรมที่ 8	5.88%	11.76%	23.53%
โปรแกรมที่ 9	20.00%	10.67%	16.00%
โปรแกรมที่ 10	11.11%	22.22%	44.44%

ตารางที่ 19 การเปรียบเทียบการประเมินผลแบบจำลองทั้งสามแบบจำลอง ในกรณีพิจารณาประเภทของข้อผิดพลาด

โปรแกรม	PRED (0.25)														
	Logic fault			Computation fault			Interface/timing fault			Data handling fault			Data fault		
	Model 1	*Model 2*	*Model 3*	*Model 1*	*Model 2*	*Model 3*	*Model 1*	*Model 2*	*Model 3*	*Model 1*	*Model 2*	*Model 3*	*Model 1*	*Model 2*	*Model 3*
โปรแกรม 1	16.67%	50.00%	66.67%	-	-	-	50.00%	50.00%	75.00%	60.00%	40.00%	46.67%	75.00%	37.50%	62.50%
โปรแกรม 2	30.77%	38.46%	46.15%	-	-	-	-	-	-	28.57%	14.29%	23.81%	28.57%	28.57%	42.86%
โปรแกรม 3	9.09%	18.18%	27.27%	-	-	-	-	-	-	-	-	-	18.18%	27.27%	27.27%
โปรแกรม 4	72.73%	18.18%	18.18%	0.00%	33.33%	33.33%	-	-	-	-	-	-	-	-	-
โปรแกรม 5	-	-	-	50.00%	37.50%	37.50%	-	-	-	66.67%	58.33%	58.33%	55.56%	22.22%	22.22%
โปรแกรม 6	60.00%	25.00%	30.00%	37.50%	50.00%	50.00%	22.22%	33.33%	33.33%	57.14%	28.57%	33.33%	60.00%	40.00%	60.00%
โปรแกรม 7	0.00%	50.00%	100.00%	-	-	-	-	-	-	76.92%	30.77%	92.31%	80.00%	20.00%	60.00%
โปรแกรม 8	-	-	-	-	-	-	0.00%	30.00%	40.00%	33.33%	66.67%	66.67%	50.00%	50.00%	50.00%
โปรแกรม 9	19.51%	19.51%	31.71%	-	-	-	56.00%	24.00%	32.00%	77.78%	44.44%	44.44%	-	-	-
โปรแกรม 10	50.00%	50.00%	75.00%	-	-	-	0.00%	20.00%	40.00%	-	-	-	-	-	-

4.4.3.1 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 2 กรณีไม่พิจารณาประเภทข้อผิดพลาด

จากตารางที่ 18 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 2 กรณีไม่พิจารณาประเภทข้อผิดพลาดพบว่า แบบจำลองที่ 1 ดีกว่าแบบจำลองที่ 2 ในกรณีไม่พิจารณาประเภทข้อผิดพลาด เนื่องจากแบบจำลองที่ 1 มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 2 กรณีไม่พิจารณาประเภทข้อผิดพลาด แต่อย่างไรก็ตามนั้นค่าการทำนายระดับแอลของทั้งสองแบบจำลองนั้นยังมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองได้

4.4.3.2 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 2 กรณีพิจารณาประเภทข้อผิดพลาด

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 2 กรณีพิจารณาประเภทข้อผิดพลาดพบว่า แบบจำลองที่ 1 ดีกว่าแบบจำลองที่ 2 ในกรณีพิจารณาประเภทข้อผิดพลาด เนื่องจากแบบจำลองที่ 1 มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 2 กรณีพิจารณาประเภทข้อผิดพลาด แต่อย่างไรก็ตามนั้นค่าการทำนายระดับแอลของทั้งสองแบบจำลองในบางประเภทข้อผิดพลาดนั้นยังมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองได้

4.4.3.3 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 3 กรณีไม่พิจารณาประเภทข้อผิดพลาด

จากตารางที่ 18 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 3 กรณีไม่พิจารณาประเภทข้อผิดพลาดพบว่า แบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 1 ในกรณีไม่พิจารณาประเภทข้อผิดพลาด เนื่องจากแบบจำลองที่ 3 มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 1 ของกรณีไม่พิจารณาประเภทข้อผิดพลาด แต่อย่างไรก็ตามนั้นค่าการทำนายระดับแอลของทั้งสองแบบจำลองนั้นบางโปรแกรมยังมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองได้

4.4.3.4 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 3 กรณีพิจารณาประเภทข้อผิดพลาด

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1 กับแบบจำลองที่ 3 กรณีพิจารณาประเภทข้อผิดพลาดพบว่า แบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 1 ในกรณีพิจารณาประเภทข้อผิดพลาด เนื่องจากแบบจำลองที่ 3 มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 1 กรณีไม่พิจารณาประเภทข้อผิดพลาด แต่อย่างไรก็ตามนั้นค่าการทำนายระดับ

แอลของทั้งสองแบบจำลองในบางประเภทข้อผิดพลาดนั้นยังมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองได้

4.4.3.5 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 2 กับแบบจำลองที่ 3 กรณีไม่พิจารณาประเภทข้อผิดพลาด

จากตารางที่ 18 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2 กับแบบจำลองที่ 3 กรณีไม่พิจารณาประเภทข้อผิดพลาดพบว่า แบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 2 ในกรณีไม่พิจารณาประเภทข้อผิดพลาด เนื่องจากแบบจำลองที่ 3 มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 2 ของกรณีไม่พิจารณาประเภทข้อผิดพลาด แต่อย่างไรก็ตามนั้นค่าการทำนายระดับแอลของทั้งสองแบบจำลองนั้นบางโปรแกรมยังมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองได้

4.4.3.6 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองที่ 2 กับแบบจำลองที่ 3 กรณีพิจารณาประเภทข้อผิดพลาด

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2 กับแบบจำลองที่ 3 กรณีพิจารณาประเภทข้อผิดพลาดพบว่า แบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 2 ในกรณีพิจารณาประเภทข้อผิดพลาด เนื่องจากแบบจำลองที่ 3 มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 2 กรณีไม่พิจารณาประเภทข้อผิดพลาด แต่อย่างไรก็ตามนั้นค่าการทำนายระดับแอลของทั้งสองแบบจำลองในบางประเภทข้อผิดพลาดนั้นยังมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.4.3.7 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองทั้งหมด กรณีไม่พิจารณาประเภทข้อผิดพลาด

จากตารางที่ 18 เมื่อเปรียบเทียบค่าการทำนายระดับแอลของทั้งสามแบบจำลอง ในกรณีที่ไม่พิจารณาประเภทข้อผิดพลาด พบว่าแบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 1 และแบบจำลองที่ 2 เพราะว่าผลลัพธ์ของโปรแกรมส่วนใหญ่มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 1 และแบบจำลองที่ 2 แต่อย่างไรก็ตามแบบจำลองที่ 3 นั้นมีค่าการทำนายที่ระดับแอลน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองนี้ได้

4.4.3.8 เปรียบเทียบค่าการทำนายระดับแอลของแบบจำลองทั้งหมด แยกตามประเภทของข้อผิดพลาด

1) กรณีที่ประเภทของข้อผิดพลาดเป็น Logic fault

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองทั้งหมด ในกรณีที่ประเภทของข้อผิดพลาดเป็น Logic fault พบว่าแบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 1 และแบบจำลองที่ 2 เพราะว่าผลลัพธ์ของโปรแกรมส่วนใหญ่มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 1 และแบบจำลองที่ 2

2) กรณีที่ประเภทของข้อผิดพลาดเป็น Computation fault

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองทั้งหมด ในกรณีที่ประเภทของข้อผิดพลาดเป็น Computation fault พบว่าแบบจำลองที่ 2 และแบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 1 เพราะว่าแบบจำลองที่ 2 และแบบจำลองที่ 3 นั้นมีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 1

3) กรณีที่ประเภทของข้อผิดพลาดเป็น Interface/timing fault

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองทั้งหมด ในกรณีที่ประเภทของข้อผิดพลาดเป็น Interface/timing fault พบว่าแบบจำลองที่ 3 ดีกว่าแบบจำลองที่ 1 และแบบจำลองที่ 2 เพราะว่าผลลัพธ์ของโปรแกรมส่วนใหญ่มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 1 และแบบจำลองที่ 2

4) กรณีที่ประเภทของข้อผิดพลาดเป็น Data handling fault

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองทั้งหมด ในกรณีที่ประเภทของข้อผิดพลาดเป็น Data handling fault พบว่าแบบจำลองที่ 1 ดีกว่าแบบจำลองที่ 2 และแบบจำลองที่ 3 เพราะว่าผลลัพธ์ของโปรแกรมส่วนใหญ่มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 2 และแบบจำลองที่ 3

5) กรณีที่ประเภทของข้อผิดพลาดเป็น Data fault

จากตารางที่ 19 เมื่อเปรียบเทียบค่าการทำนายที่ระดับแอลของแบบจำลองทั้งหมด ในกรณีที่ประเภทของข้อผิดพลาดเป็น Data fault พบว่าแบบจำลองที่ 1 ดีกว่าแบบจำลองที่ 2 และแบบจำลองที่ 3 เพราะว่าผลลัพธ์ของโปรแกรมส่วนใหญ่มีค่าการทำนายที่ระดับแอลมากกว่าแบบจำลองที่ 2 และแบบจำลองที่ 3

บทที่ 5
การออกแบบและพัฒนาเครื่องมือ สนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์

ในบทนี้จะกล่าวถึงรายละเอียดของการออกแบบและพัฒนาเครื่องมือเพื่อสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ ซึ่งมีอธิบายรายละเอียดของความต้องการเชิงหน้าที่ การออกแบบเครื่องมือ การทำงานและส่วนต่อประสานกับผู้ใช้ และการทดสอบเครื่องมือ โดยมีเนื้อหาดังนี้

5.1 ความต้องการเชิงหน้าที่

ในการพัฒนาเครื่องมือที่สนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ระบบจะต้องมีความสามารถในการทำงานที่ถูกต้องตามความต้องการเชิงหน้าที่ ซึ่งความต้องการเชิงหน้าที่ของระบบนั้นประกอบด้วยส่วนการทำงานหลักๆ ดังนี้

5.1.1 การนำเข้าไฟล์ข้อมูล

เครื่องมือที่พัฒนาเพื่อสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนี้ มีการนำเข้าไฟล์ข้อมูลซึ่งอยู่ในรูปแบบของตาราง (.csv) ซึ่งก่อนจะนำเข้าไฟล์นั้นผู้ใช้งานจะต้องเลือกการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดว่าต้องการประมาณแบบพิจารณามาตรวัดหรือแบบไม่พิจารณามาตรวัด ซึ่งหากเลือกแบบพิจารณามาตรวัด ข้อมูลที่อยู่ในไฟล์นำเข้านั้นจะต้องประกอบด้วย หมายเลขข้อผิดพลาด ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ แต่หากเลือกแบบไม่ต้องการพิจารณามาตรวัด ข้อมูลที่อยู่ในไฟล์นำเข้านั้นจะประกอบด้วยหมายเลขข้อผิดพลาด และค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด เมื่อนำเข้าไฟล์ข้อมูลมาในระบบเสร็จเรียบร้อยแล้ว เครื่องมือจะทำการอ่านไฟล์และแสดงผลข้อมูลนำเข้าในรูปแบบตารางในส่วนต่อประสานผู้ใช้

5.1.2 การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

เมื่อนำเข้าไฟล์ข้อมูลมาในระบบของเครื่องมือ เครื่องมือจะทำการอ่านไฟล์และเก็บค่าต่างๆ จากนั้นจะทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดมีทั้งหมด 3 แบบจำลองที่งานวิจัยได้นำเสนอไว้ คือ คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจง

ข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ ซึ่งในการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจะขึ้นอยู่กับผู้ใช้งาน ซึ่งผู้ใช้งานจะต้องเลือกรูปแบบการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดว่าต้องการประมาณแบบพิจารณามาตรวัดหรือแบบไม่พิจารณามาตรวัด หากเลือกแบบพิจารณามาตรวัด เครื่องมือจะคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองทั้ง 3 แบบจำลอง แต่หากเลือกแบบไม่ต้องการพิจารณามาตรวัด เครื่องมือจะคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองที่ 1 และแบบจำลองที่ 2 เท่านั้น เมื่อทำการคำนวณค่าประมาณที่ใช้ในการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว เครื่องมือจะเก็บข้อมูลไว้เพื่อแสดงผลในขั้นตอนถัดไป

5.1.3 การแสดงผลและการส่งออกข้อมูล

เมื่อเครื่องมือทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว ระบบจะทำการแสดงผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบของตาราง และแสดงผลในรูปแบบของกราฟเส้นในส่วนต่อประสานผู้ใช้ ซึ่งหากผู้ใช้งานเลือกการประมาณค่าแบบพิจารณามาตรวัด เครื่องมือจะแสดงผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองทั้ง 3 แบบจำลอง และแสดงการเปรียบเทียบของผลการประมาณค่าของทั้ง 3 แบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด นอกจากนี้เครื่องมือจะแสดงกราฟเส้นของค่าประมาณแต่ละแบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และแสดงกราฟเส้นเปรียบเทียบค่าประมาณที่ได้จากแบบจำลองทั้ง 3 แบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด แต่หากผู้ใช้งานเลือกการประมาณค่าแบบไม่ต้องการพิจารณามาตรวัดเครื่องมือจะแสดงผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 และแบบจำลองที่ 2 และแสดงการเปรียบเทียบของผลการประมาณค่าของทั้ง 2 แบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด นอกจากนี้เครื่องมือจะแสดงกราฟเส้นของค่าประมาณแต่ละแบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และแสดงกราฟเส้นเปรียบเทียบค่าประมาณที่ได้จากแบบจำลองทั้ง 2 แบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และแสดงผลการทำนายระดับแอลของแต่ละแบบจำลอง เมื่อแสดงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดแล้วนั้น เครื่องมือสามารถบันทึกไฟล์ของผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเพื่อนำข้อมูลที่ได้จากการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองไปใช้งานในภายหลัง โดยเครื่องมือจะบันทึกผลการประมาณเป็นไฟล์ในรูปแบบตาราง (.csv) และบันทึกกราฟเส้นเป็นไฟล์ในรูปแบบของรูปภาพ (.PNG)

5.2 การออกแบบเครื่องมือ

เครื่องมือนี้พัฒนาขึ้นเพื่อสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ และอำนายความสะดวกในการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งสามารถอธิบายรายละเอียดต่างๆ ได้ดังนี้

5.2.1 แผนภาพยูสเคส

ในงานวิจัยนี้ ผู้วิจัยได้ทำการพัฒนาเครื่องมือสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดตามวิธีที่งานวิจัยได้นำเสนอ พัฒนาโดยใช้ภาษาจาวาและทดสอบบนระบบปฏิบัติการวินโดวส์ ซึ่งหลักการทำงานและความสามารถของเครื่องมือสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด สามารถอธิบายรายละเอียดดังนี้

1) ผู้ใช้งานสามารถนำเข้าข้อมูลโดยเป็นไฟล์ในรูปแบบของตาราง (.csv) เพื่อนำไปคำนวณหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้

2) ผู้ใช้งานสามารถนำเข้าไฟล์ข้อมูลซึ่งอยู่ในรูปแบบตาราง โดยสามารถเลือกรูปแบบการประมาณค่าโดยการพิจารณามาตรวัดหรือไม่พิจารณามาตรวัดได้

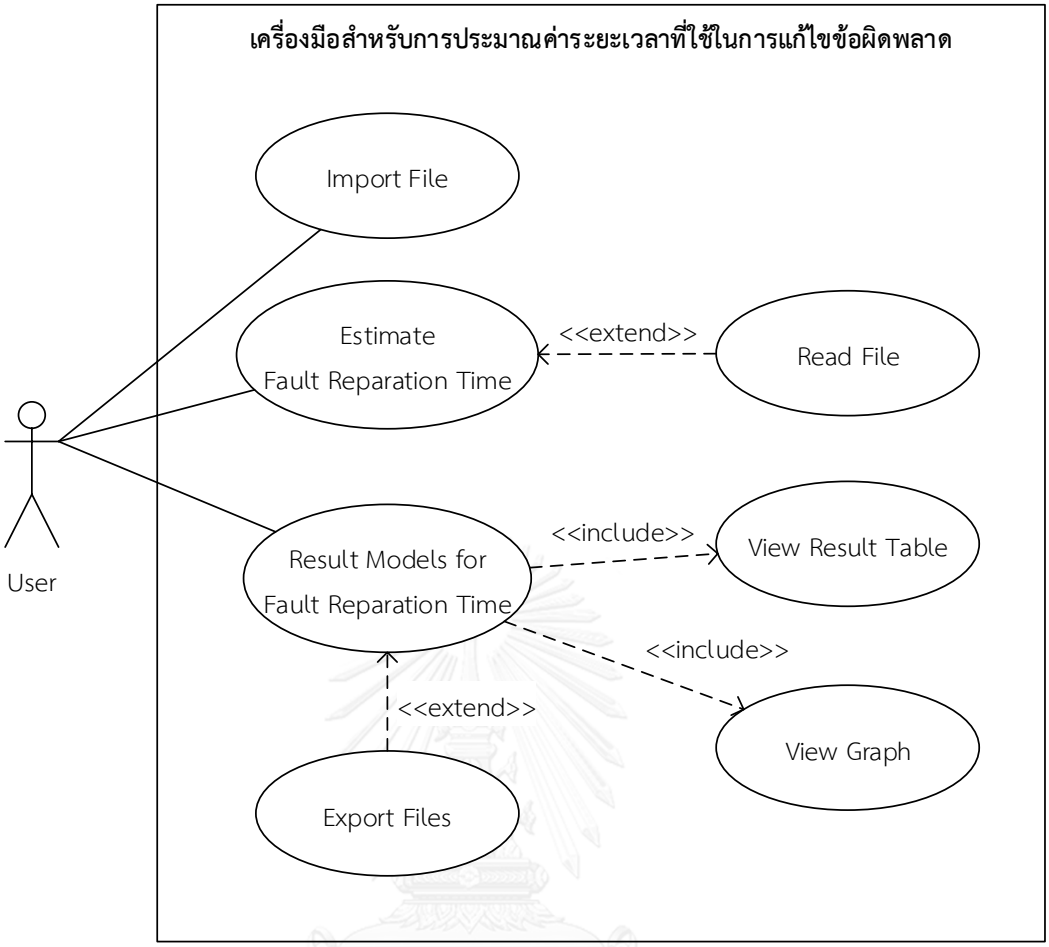
3) ผู้ใช้งานสามารถประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดตามแบบจำลองที่ได้นำเสนอทั้ง 3 แบบจำลองได้

4) ผู้ใช้งานสามารถเลือกการแสดงผลของการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยผู้ใช้สามารถเลือกการแสดงผลได้ทั้ง 3 แบบจำลอง และแสดงผลการเปรียบเทียบค่าประมาณกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดได้ โดยการแสดงผลประกอบด้วย ตารางผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง และกราฟ ซึ่งแสดงการเปรียบเทียบค่าประมาณกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด

5) ผู้ใช้งานสามารถบันทึกไฟล์ผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง และผลการเปรียบเทียบของค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ในรูปแบบของตารางได้

6) ผู้ใช้งานสามารถบันทึกไฟล์รูปภาพของกราฟเส้นแสดงการเปรียบเทียบค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดกับค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง และบันทึกไฟล์รูปภาพของกราฟเส้นแสดงการเปรียบเทียบของทุกแบบจำลองได้

จากการทำงานและความสามารถของเครื่องมือสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดดังที่ได้กล่าวข้างต้นนั้น สามารถอธิบายได้ด้วยแผนภาพยูสเคสได้ดังรูปที่ 15



รูปที่ 15 แผนภาพยูสเคสของเครื่องมือสำหรับการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

จากรูปที่ 15 แสดงแผนภาพยูสเคสของเครื่องมือสำหรับการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยสามารถอธิบายความสัมพันธ์ของยูสเคสที่เกิดขึ้นได้ดังต่อไปนี้

1) ยูสเคส Import File

จากตารางที่ 20 เป็นยูสเคสที่แสดงการทำงาน เมื่อผู้ใช้งานนำเข้าไฟล์ข้อมูลที่อยู่ในรูปแบบของตาราง เพื่อนำข้อมูลไปใช้ในการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

ตารางที่ 20 ขั้นตอนการทำงานของยูสเคส Import File

Use case name	Import File
Entry condition	1. ระบุรูปแบบการประมาณค่า โดยเลือกแบบการพิจารณามาตรวัดหรือเลือกแบบไม่พิจารณามาตรวัด 2. มีไฟล์ชุดข้อมูลของค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด 3. มีข้อมูลของมาตรวัดความไม่เสถียรของซอฟต์แวร์

ตารางที่ 20 ขั้นตอนการทำงานของยูสเคส Import File (ต่อ)

Flow of events	4. ผู้ใช้งานเลือกรูปแบบการประมาณค่า 5. ผู้ใช้งานนำเข้าไฟล์ข้อมูลที่อยู่ในรูปแบบตาราง (.csv) 6. ระบบแสดงข้อมูลที่ได้นำเข้ามา
Exit condition	7. ผู้ใช้งานเลือกทำขั้นตอนต่อไป

2) ยูสเคส Estimate Fault Reparation Time

จากตารางที่ 21 เป็นยูสเคสแสดงการทำงานของระบบ เมื่อผู้ใช้งานนำเข้าไฟล์ชุดข้อมูลเสร็จเรียบร้อยแล้ว ระบบจะทำการอ่านค่าจากไฟล์ชุดข้อมูลที่ผู้ใช้งานนำเข้ามายังระบบ จากนั้นทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ตามแบบจำลองที่ได้นำเสนอไว้ในงานวิจัยนี้ เมื่อระบบทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว ระบบจะทำการเก็บข้อมูลของค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดแต่ละแบบจำลองไว้เพื่อนำไปแสดงในขั้นตอนถัดไป

ตารางที่ 21 ขั้นตอนการทำงานของยูสเคส Estimate Fault Reparation Time

Use case name	Estimate Fault Reparation Time
Entry condition	1. มีไฟล์ชุดข้อมูลของค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และข้อมูลมาตรวัดความไม่เสถียรของซอฟต์แวร์ที่ได้จากการนำเข้า และระบบสามารถเปิดอ่านไฟล์ได้
Flow of events	2. ระบบอ่านค่าที่ได้จากการนำเข้า และเก็บข้อมูลหมายเลขข้อผิดพลาด ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และมาตรวัดความไม่เสถียรของซอฟต์แวร์ 3. ระบบทำการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ตามแบบจำลองที่ได้นำเสนอไว้ข้างต้น 4. ระบบจัดเก็บค่าที่ได้จากการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง 5. ระบบแสดงข้อความว่าคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเสร็จเรียบร้อย
Exit condition	6. ผู้ใช้งานเลือกทำขั้นตอนต่อไป

3) ยูสเคส Result Models for Fault Reparation Time

จากตารางที่ 22 เป็นยูสเคสแสดงการทำงานของระบบ ในส่วนของการแสดงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง โดยแสดงผลของค่าจริงและค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองที่ผู้ใช้งานเลือก ซึ่งแสดงผลในรูปแบบตารางและกราฟเส้น นอกจากนี้ระบบจะทำการแสดงผลค่าการทำนายที่ระดับแอลของแต่ละแบบจำลอง

ตารางที่ 22 ขั้นตอนการทำงานของยูสเคส Result Models for Fault Reparation Time

Use case name	Result Models for Fault Reparation Time
Entry condition	1. มีข้อมูลของการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง
Flow of events	2. ผู้ใช้งานเลือกแถบผลการประมาณของแบบจำลองที่ต้องการ 3. ระบบแสดงค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และแสดงผลการเปรียบเทียบระหว่างค่าประมาณจากแบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด ในรูปแบบของตาราง 4. ระบบแสดงกราฟเส้นการเปรียบเทียบค่าประมาณจากแบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง และเปรียบเทียบทุกแบบจำลอง 5. ระบบแสดงผลค่าการทำนายที่ระดับแอลของแต่ละแบบจำลองเป็นข้อความ
Exit condition	6. ผู้ใช้งานเลือกใช้งานเมนูอื่น

4) ยูสเคส Export Files

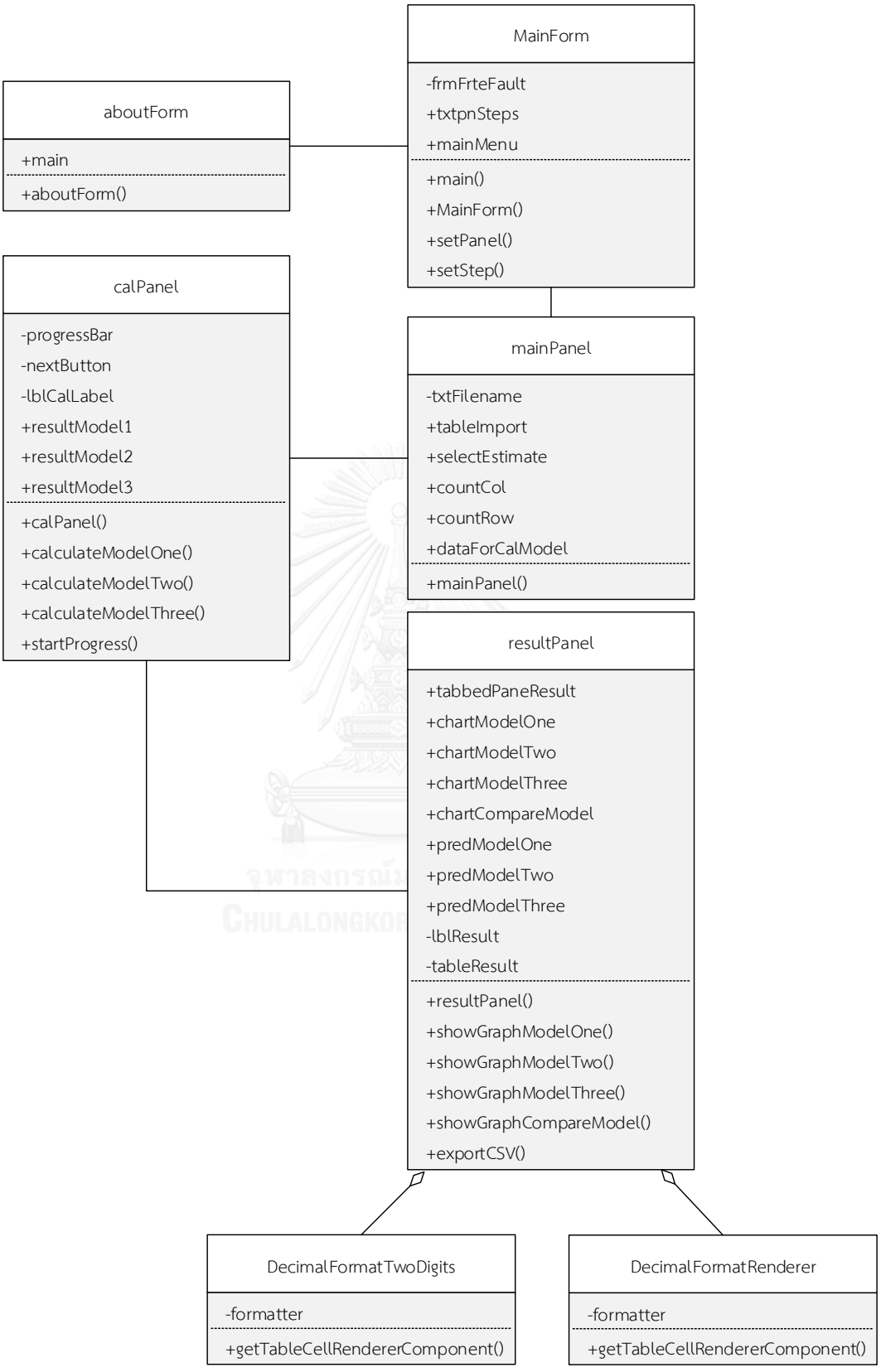
จากตารางที่ 23 เป็นยูสเคสแสดงการทำงานของระบบที่เกิดขึ้นเมื่อผู้ใช้งานต้องการนำออกข้อมูลหรือบันทึกข้อมูลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยบันทึกไฟล์ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบของตาราง (.csv)และกราฟเส้น ของแบบจำลองที่ผู้ใช้งานเลือก

ตารางที่ 23 ขั้นตอนการทำงานของยูสเคส Export Files

Use case name	Export Files
Entry condition	1. มีข้อมูลของการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลอง 2. ผู้ใช้งานเลือกผลการประมาณค่าที่ได้จากแบบจำลอง ที่ต้องการบันทึกลงไฟล์ในรูปแบบตารางและรูปแบบกราฟเส้น 3. ผู้ใช้งานเลือกรายการ Export Files
Flow of events	4. ผู้ใช้งานเลือกที่อยู่ที่ต้องการบันทึกไฟล์ตารางและไฟล์รูปภาพกราฟเส้นแสดงการเปรียบเทียบค่าประมาณจากแบบจำลองกับค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด 5. ผู้ใช้งานตั้งชื่อของไฟล์ที่ต้องการบันทึก และไม่ซ้ำกับชื่อไฟล์เดิมที่มีอยู่แล้วในที่อยู่นั้นๆ
Exit condition	6. ผู้ใช้งานเลือกบันทึกข้อมูล หรือยกเลิกการบันทึกข้อมูล

5.2.2 แผนภาพคลาส

ในงานวิจัยนี้ ผู้วิจัยได้ทำการพัฒนาเครื่องมือสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดตามวิธีที่งานวิจัยได้นำเสนอ ซึ่งสามารถอธิบายองค์ประกอบและความสัมพันธ์ของระบบภายในเครื่องมือ โดยแสดงเป็นแผนภาพคลาสแสดงองค์ประกอบและความสัมพันธ์ของเครื่องมือดังรูปที่ 16



รูปที่ 16 แผนภาพคลาสแสดงองค์ประกอบและความสัมพันธ์ของเครื่องมือ

จากรูปที่ 16 แสดงแผนภาพคลาสขององค์ประกอบและความสัมพันธ์ของคลาสต่างๆ ภายในเครื่องมือ ซึ่งสามารถอธิบายโดยละเอียดดังนี้

1) แผนภาพคลาส MainForm

คลาส MainForm เป็นคลาสแสดงหน้าจอการติดต่อหลักของเครื่องมือ ซึ่งเป็นคลาสหลักของระบบที่ใช้ในการเรียกหน้าจออื่นๆ ขึ้นมาแสดง โดยมีการเรียกใช้คลาส mainPanel เพื่อแสดงส่วนของการนำเข้าข้อมูลและแสดงข้อมูลที่นำเข้ามา และเรียกใช้คลาส aboutForm เพื่อแสดงรายละเอียดของผู้พัฒนาเครื่องมือ

2) แผนภาพคลาส mainPanel

คลาส mainPanel เป็นคลาสที่แสดงส่วนของการเลือกรูปแบบการประมาณ การนำเข้าข้อมูล และการแสดงข้อมูลที่ผู้ใช้งานนำเข้ามายังระบบในรูปแบบตาราง โดยผู้ใช้งานจะเลือกรูปแบบการประมาณว่าต้องการประมาณโดยพิจารณามาตรวัดหรือไม่พิจารณามาตรวัด และนำเข้าข้อมูลในรูปแบบตาราง (.csv) จากนั้นระบบจะอ่านไฟล์ที่ผู้ใช้นำเข้าและนำไปแสดงผลในตารางเพื่อให้ผู้ใช้ตรวจสอบว่าข้อมูลที่นำเข้ามานั้นถูกต้องหรือไม่ ในคลาสนี้จะมีการเรียกใช้คลาส calPanel เพื่อนำข้อมูลที่ได้จากคลาสนี้ไปคำนวณหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

3) แผนภาพคลาส calPanel

คลาส calPanel เป็นคลาสที่ระบบจะทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากข้อมูลที่ผู้ใช้งานนำเข้ามา โดยที่คลาส calculateModelOne ใช้ในการคำนวณค่าประมาณของแบบจำลองที่ 1, คลาส calculateModelTwo ใช้ในการคำนวณค่าประมาณของแบบจำลองที่ 2 และคลาส calculateModelThree ใช้ในการคำนวณค่าประมาณของแบบจำลองที่ 3 ซึ่งในคลาส calPanel นี้จะมีการเรียกใช้คลาส resultPanel เพื่อนำค่าประมาณที่ได้จากคลาสนี้ไปแสดงผล

4) แผนภาพคลาส resultPanel

คลาส resultPanel เป็นคลาสที่แสดงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากการคำนวณในคลาส calPanel ตามแต่ละแบบจำลองที่ได้นำเสนอในงานวิจัยนี้และเปรียบเทียบกับแต่ละแบบจำลองในรูปแบบของตารางและกราฟเส้น นอกจากนี้คลาสนี้ทำหน้าที่ในการบันทึกผลการประมาณเป็นไฟล์ในรูปแบบตาราง (.csv) และบันทึกกราฟเส้นเป็นไฟล์ในรูปแบบของรูปภาพ (.PNG) ซึ่งในคลาสนี้มีการเรียกใช้ซับคลาส DecimalFormatTwoDigits เพื่อให้ผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่แสดงผลในตารางตามคอลัมน์ที่ต้องการนั้นแสดงผลเป็นเลขทศนิยมสองตำแหน่ง และเรียกใช้ซับคลาส DecimalFormatRenderer เพื่อให้ผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่แสดงในตารางตามคอลัมน์ที่ต้องการนั้นหากผลการ

ประมาณใดมีทศนิยมจะให้แสดงผลเป็นเลขทศนิยม แต่หากผลการประมาณใดไม่มีทศนิยมก็ไม่ต้องแสดงผลเป็นเลขทศนิยม

5) แผนภาพคลาส DecimalFormatTwoDigits

คลาส DecimalFormatTwoDigits เป็นส่วนหนึ่งของคลาส resultPanel ทำหน้าที่แปลงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ให้แสดงผลในตารางตามคอลัมน์ที่ต้องการนั้นเป็นเลขทศนิยมสองตำแหน่ง

6) แผนภาพคลาส DecimalFormatRenderer

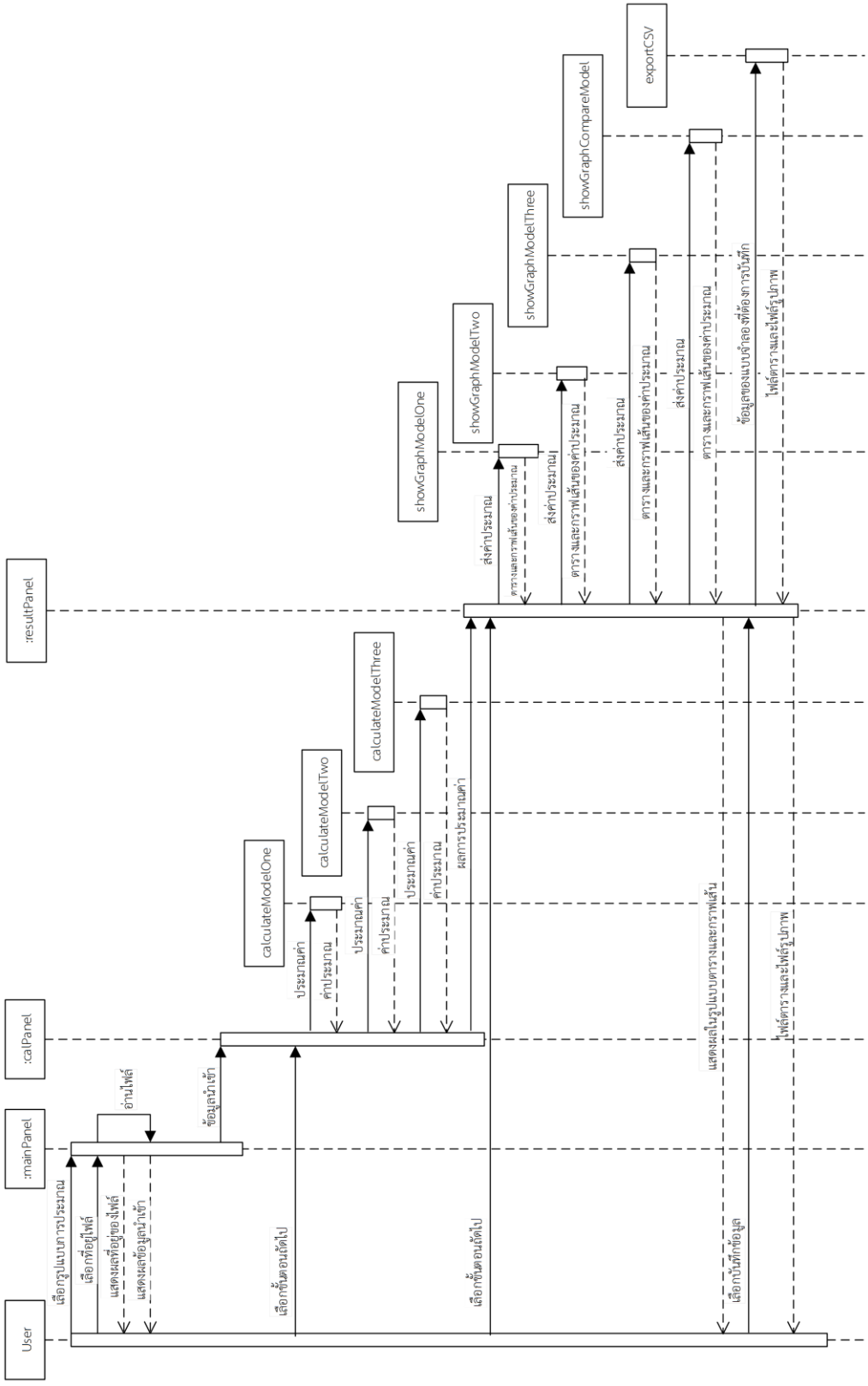
คลาส DecimalFormatRenderer เป็นส่วนหนึ่งของคลาส resultPanel ทำหน้าที่แปลงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ให้แสดงผลในตารางตามคอลัมน์ที่ต้องการ โดยหากผลการประมาณใดมีทศนิยมจะให้แสดงเป็นเลขทศนิยม แต่หากผลการประมาณใดไม่มีทศนิยมก็ไม่ต้องแสดงผลเป็นเลขทศนิยม

7) แผนภาพคลาส aboutForm

คลาส aboutForm เป็นคลาสแสดงชื่อและรุ่นของเครื่องมือ และแสดงรายละเอียดของผู้พัฒนาเครื่องมือ

5.2.3 แผนภาพลำดับ

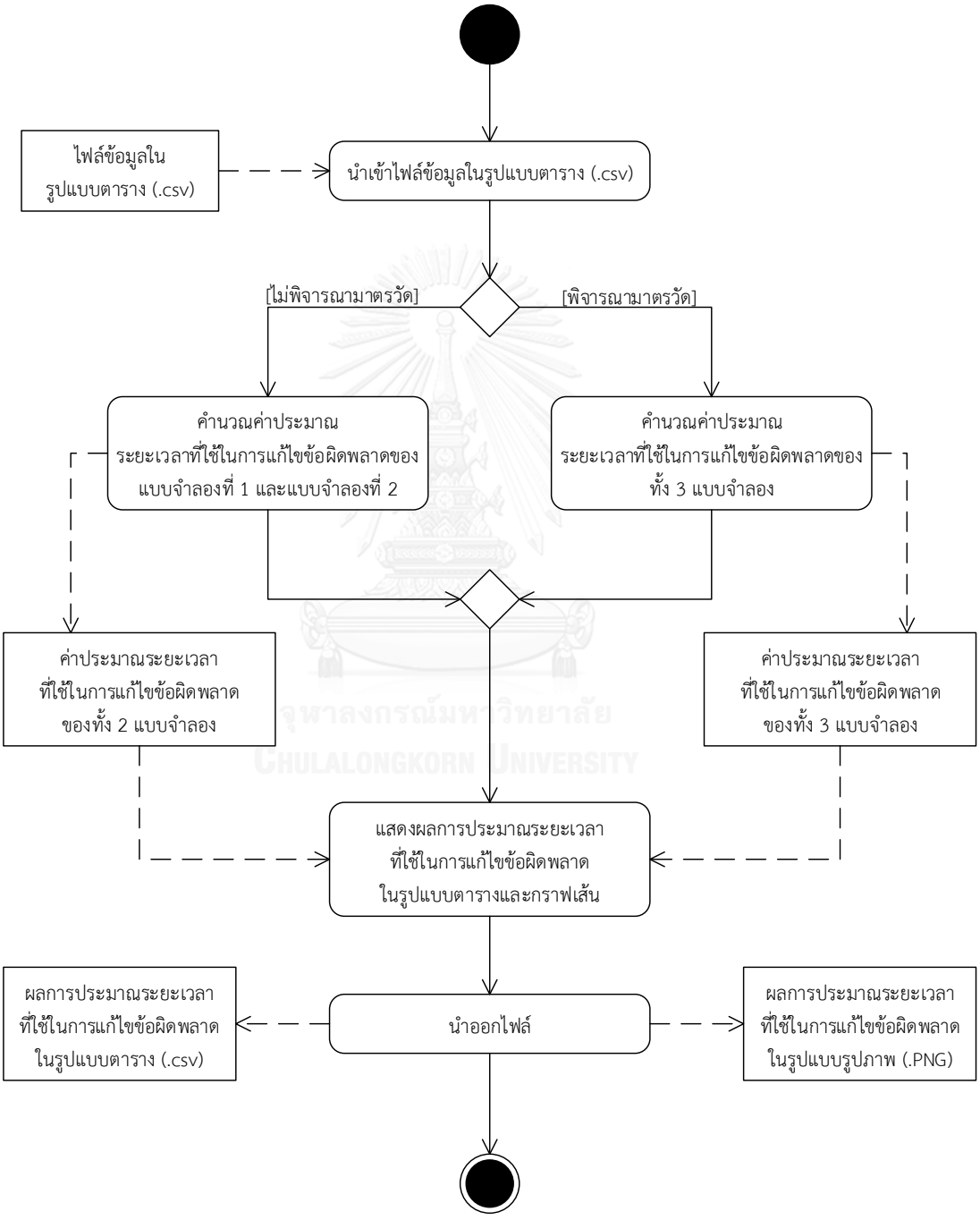
แผนภาพลำดับแสดงการทำงานของระบบ โดยผู้ใช้งานจะเลือกรูปแบบการประมาณว่าต้องการประมาณโดยพิจารณามาตรวัดหรือไม่พิจารณามาตรวัด จากนั้นผู้ใช้งานจะเลือกที่อยู่ของไฟล์ในรูปแบบตาราง (.csv) เมื่อผู้ใช้งานเลือกไฟล์เสร็จเรียบร้อยแล้ว ระบบจะทำการแสดงที่อยู่ของไฟล์และอ่านค่าข้อมูลในไฟล์แล้วนำขึ้นแสดงผลในตารางของหน้าส่วนต่อประสานผู้ใช้ เมื่อข้อมูลถูกต้องแล้ว ผู้ใช้งานจะทำการเลือกขั้นตอนถัดไปเพื่อทำการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยระบบจะส่งค่าที่อ่านได้จากไฟล์นำเข้าไปใช้ในการคำนวณ ในการคำนวณนั้นหากผู้ใช้งานเลือกรูปแบบการประมาณเป็นพิจารณามาตรวัด ระบบจะทำการคำนวณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากทั้งสามแบบจำลอง แต่หากผู้ใช้งานเลือกรูปแบบการประมาณเป็นไม่พิจารณามาตรวัด ระบบจะทำการคำนวณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดจากแบบจำลองที่ 1 และแบบจำลองที่ 2 เมื่อระบบคำนวณเสร็จเรียบร้อยก็จะแสดงข้อความแจ้งเตือน จากนั้นผู้ใช้งานจะเลือกขั้นตอนถัดไปเพื่อเป็นการแสดงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตารางและกราฟเส้น นอกจากนี้ระบบจะแสดงผลของการเปรียบเทียบแบบจำลองตามรูปแบบของการประมาณในรูปแบบตารางและกราฟเส้น หากผู้ใช้งานต้องการจะบันทึกข้อมูลผลการประมาณระบบจะทำการบันทึกไฟล์ในรูปแบบของตาราง (.csv) และไฟล์รูปแบบของรูปภาพ (.PNG) ซึ่งแผนภาพลำดับที่ได้อธิบายข้างต้นนั้นสามารถแสดงได้ดังรูปที่ 17



รูปที่ 17 แผนภาพลำดับแสดงการทำงานของระบบ

5.2.4 แผนภาพกิจกรรม

แผนภาพกิจกรรมแสดงถึงกิจกรรมและการทำงานต่างๆ ของเครื่องมือ ซึ่งเครื่องมือนี้ใช้สำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยสามารถแสดงกิจกรรมต่างๆ ด้วยแผนภาพกิจกรรมของเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดดังรูปที่ 18



รูปที่ 18 แผนภาพกิจกรรมของเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

จากรูปที่ 18 แสดงกิจกรรมของเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยกิจกรรมเริ่มต้นคือนำเข้าไฟล์ข้อมูลในรูปแบบตาราง (.csv) ซึ่งผู้ใช้งานจะเลือกไฟล์นำเข้ามายังระบบของเครื่องมือ ซึ่งผู้ใช้งานจะต้องเลือกรูปแบบของการประมาณโดยมีให้เลือก 2 แบบ คือพิจารณามาตรวัดหรือไม่พิจารณามาตรวัด หากผู้ใช้งานเลือกรูปแบบเป็นพิจารณามาตรวัดระบบจะทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของทั้ง 3 แบบจำลอง ซึ่งเมื่อคำนวณค่าประมาณเสร็จเรียบร้อยแล้ว ก็จะได้ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของทั้ง 3 แบบจำลอง แต่หากผู้ใช้งานเลือกรูปแบบการประมาณเป็นไม่พิจารณามาตรวัด ระบบจะทำการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 และแบบจำลองที่ 2 ซึ่งเมื่อคำนวณค่าประมาณเสร็จเรียบร้อยแล้ว ก็จะได้ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 และแบบจำลองที่ 2 เมื่อได้ค่าประมาณแล้วก็จะทำการแสดงผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตารางและกราฟเส้น โดยผู้ใช้งานสามารถเลือกดูข้อมูลของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลองและสามารถดูการเปรียบเทียบของทุกแบบจำลองได้ นอกจากนี้ระบบจะแสดงผลการทำนายที่ระดับแอลของแต่ละแบบจำลอง เพื่อให้ผู้ใช้งานสามารถพิจารณาเลือกใช้แบบจำลองได้ง่ายขึ้น เมื่อระบบแสดงผลทางหน้าจอทั้งหมดแล้ว หากผู้ใช้งานต้องการนำออกไฟล์หรือบันทึกข้อมูล ระบบจะทำการบันทึกไฟล์ผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตาราง (.csv) และบันทึกไฟล์ผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบรูปภาพ (.PNG) โดยระบบจะบันทึกเฉพาะผลของแบบจำลองที่ผู้ใช้งานเลือกที่จะบันทึกเท่านั้น โดยผู้ใช้งานจะต้องบันทึกข้อมูลโดยที่ชื่อไฟล์ไม่ซ้ำกัน หากผู้ใช้งานเลือกปิดหน้าจอการทำงานของระบบ ระบบจะปิดโปรแกรมและหยุดการทำงานทั้งหมด

Chulalongkorn University

5.2.5 เครื่องมือสนับสนุนในการพัฒนา

ผู้วิจัยได้มีการใช้อุปกรณ์ฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ที่สนับสนุนการพัฒนาเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งมีรายละเอียดดังนี้

5.2.5.1 ด้านฮาร์ดแวร์

1) เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลอินเทล คอร์ ไอเจ็ด รุ่นที่ 3 รหัส 3610QM (Intel Core i7-3610QM) ความเร็ว 2.30 กิกะเฮิรตซ์ (GHz)

2) หน่วยความจำ 8 กิกะไบต์ (GB) แบบ DDR3

3) จานบันทึกแบบแข็ง (Hard disk) ความจุ 1 เทราไบต์ (TB)

5.2.5.2 ด้านซอฟต์แวร์

1) ระบบปฏิบัติการไมโครซอฟท์วินโดวส์รุ่นที่ 7 (Microsoft Windows 7)

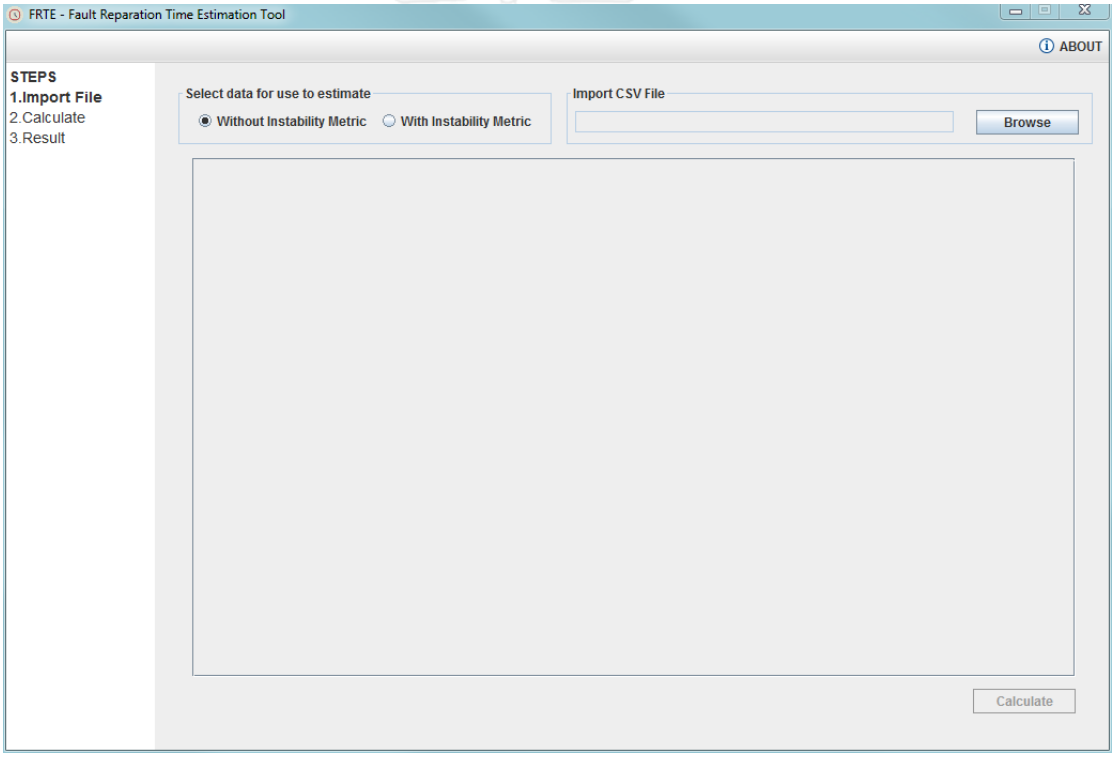
2) เครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวา (Java) อีคลิปส์รุ่นที่ 4.4.2 (Eclipse 4.4.2)

3) เครื่องมือสำหรับคำนวณค่ามาตรวัดจากซอร์ซโค้ด (SourceCodeMetrics) [23]

4) ไมโครซอฟท์ออฟฟิศ 2013 (Microsoft Office 2013)

5.3 การทำงานและส่วนต่อประสานกับผู้ใช้

เครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเป็นเครื่องมือที่พัฒนาจากงานวิจัยที่ได้นำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดทั้ง 3 แบบจำลอง ซึ่งหน้าจอหลักของเครื่องมือที่พัฒนาแสดงได้ดังรูปที่ 19



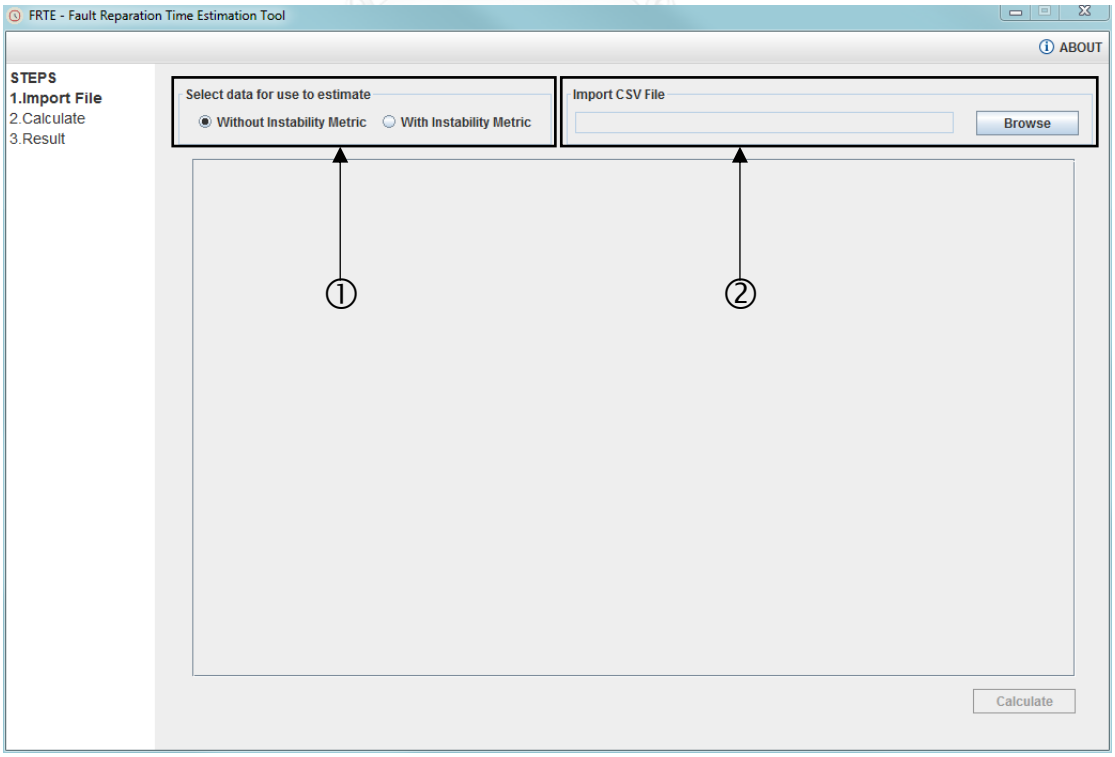
รูปที่ 19 หน้าจอหลักของเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

1) ส่วนของการนำเข้าไฟล์และขั้นตอนการนำเข้าไฟล์

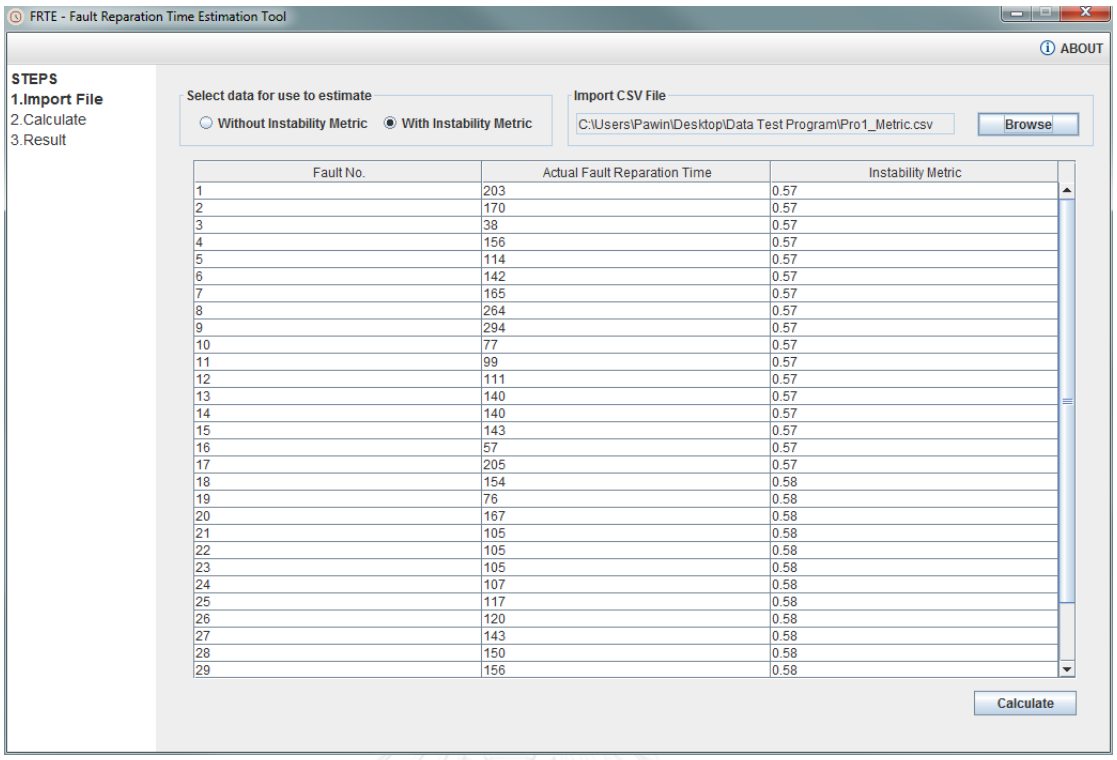
เมื่อโปรแกรมเริ่มทำงาน ระบบจะแสดงหน้าต่างหลักดังรูปที่ 19 จากนั้นผู้ใช้งานจะต้องทำรายการตามหมายเลขดังรูปที่ 20

หมายเลข 1 Select data for use to estimate เป็นการเลือกรูปแบบการประมาณ ซึ่งหากต้องการพิจารณามาตรวัดให้เลือก “With Instability Metric” แต่หากไม่ต้องการพิจารณามาตรวัดให้เลือก “Without Instability Metric”

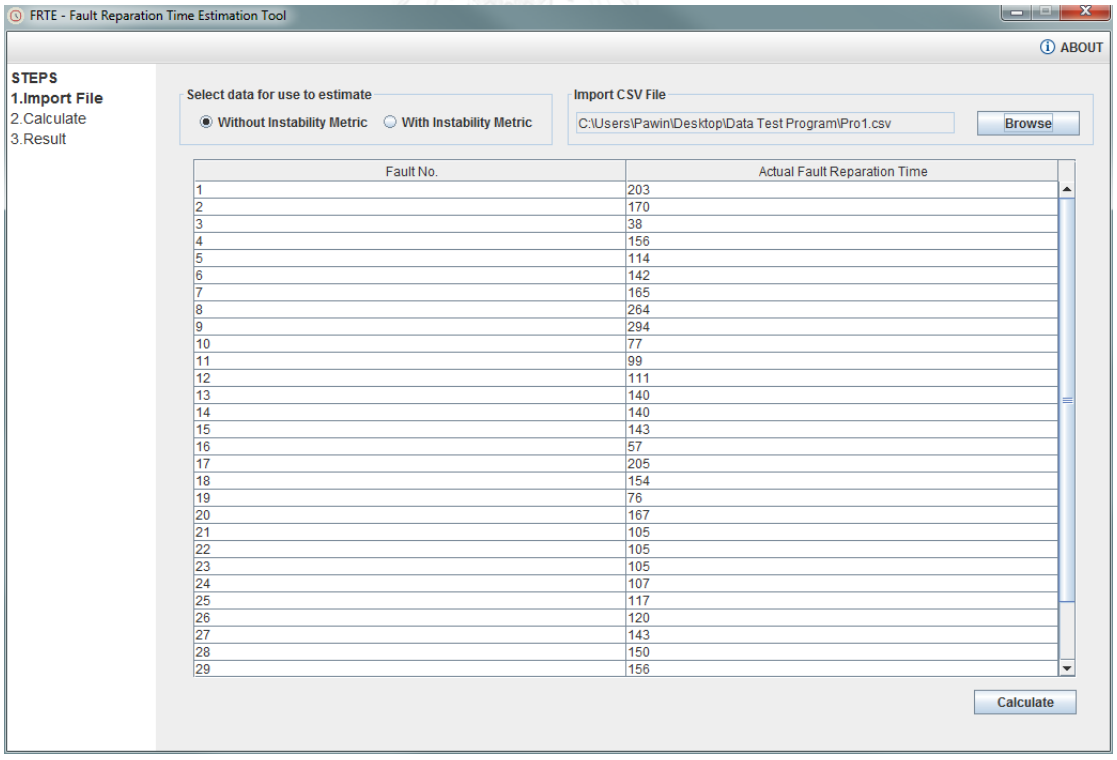
หมายเลข 2 Import CSV File เป็นการเลือกที่อยู่ข้อมูลนำเข้ามายังระบบ โดยผู้ใช้งานกดปุ่ม “Browse” เพื่อเลือกไฟล์ที่ต้องการนำเข้าสู่ระบบ โดยระบบสามารถนำเข้าไฟล์ในรูปแบบตารางที่มีนามสกุลเป็น .csv เท่านั้น โดยที่ไฟล์ที่นำเข้านั้น หากผู้ใช้งานเลือกรูปแบบการประมาณเป็นพิจารณามาตรวัด ข้อมูลในไฟล์นำเข้าจะต้องประกอบไปด้วย หมายเลขข้อผิดพลาด ค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ แต่หากผู้ใช้งานเลือกรูปแบบการประมาณเป็นไม่พิจารณามาตรวัด ข้อมูลในไฟล์นำเข้าจะต้องประกอบไปด้วย หมายเลขข้อผิดพลาด และค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด เมื่อนำเข้าไฟล์แล้วระบบจะแสดงข้อมูลนำเข้าให้กับผู้ใช้งานได้ตรวจสอบความถูกต้องของข้อมูลนำเข้า ซึ่งหากผู้ใช้งานเลือกรูปแบบการประมาณเป็นพิจารณามาตรวัด (With Instability Metric) จะแสดงตัวอย่างได้ดังรูปที่ 21 แต่หากผู้ใช้งานเลือกรูปแบบการประมาณเป็นไม่พิจารณามาตรวัด (Without Instability Metric) จะแสดงตัวอย่างได้ดังรูปที่ 22 แต่หากผู้ใช้งานนำเข้าไฟล์ไม่ถูกต้องระบบจะแจ้งเตือนดังรูปที่ 23 นอกจากนี้เมื่อระบบแสดงผลข้อมูลนำเข้าแล้ว ปุ่ม “Calculate” จะสามารถกดได้ เพื่อส่งข้อมูลนำเข้าไปคำนวณหาค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด



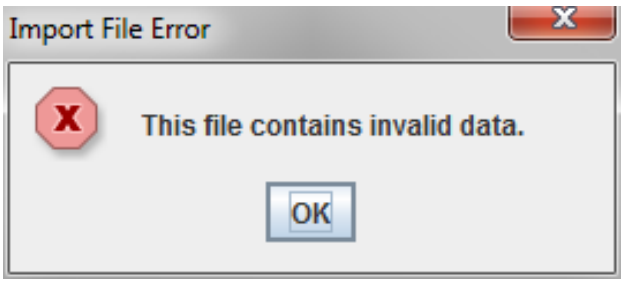
รูปที่ 20 หน้าจอสำหรับการนำเข้าไฟล์



รูปที่ 21 หน้าจอแสดงข้อมูลนำเข้าในกรณีที่พิจารณามาตรวัด



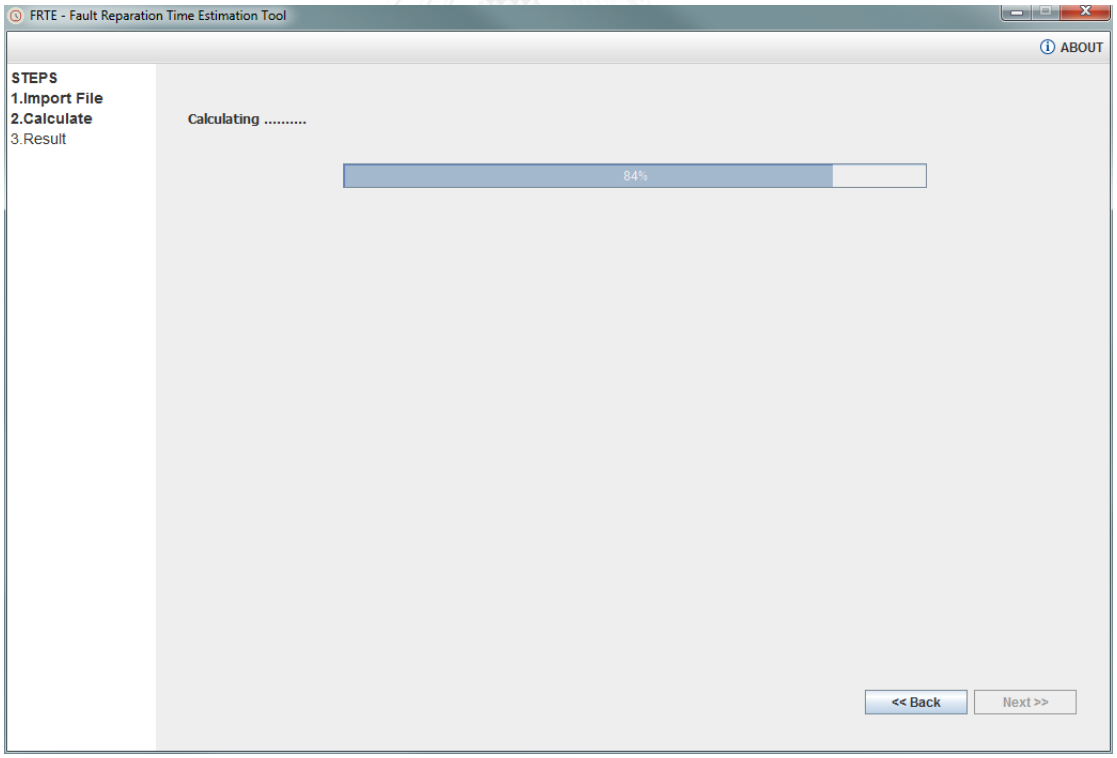
รูปที่ 22 หน้าจอแสดงข้อมูลนำเข้าในกรณีที่ไม่พิจารณามาตรวัด



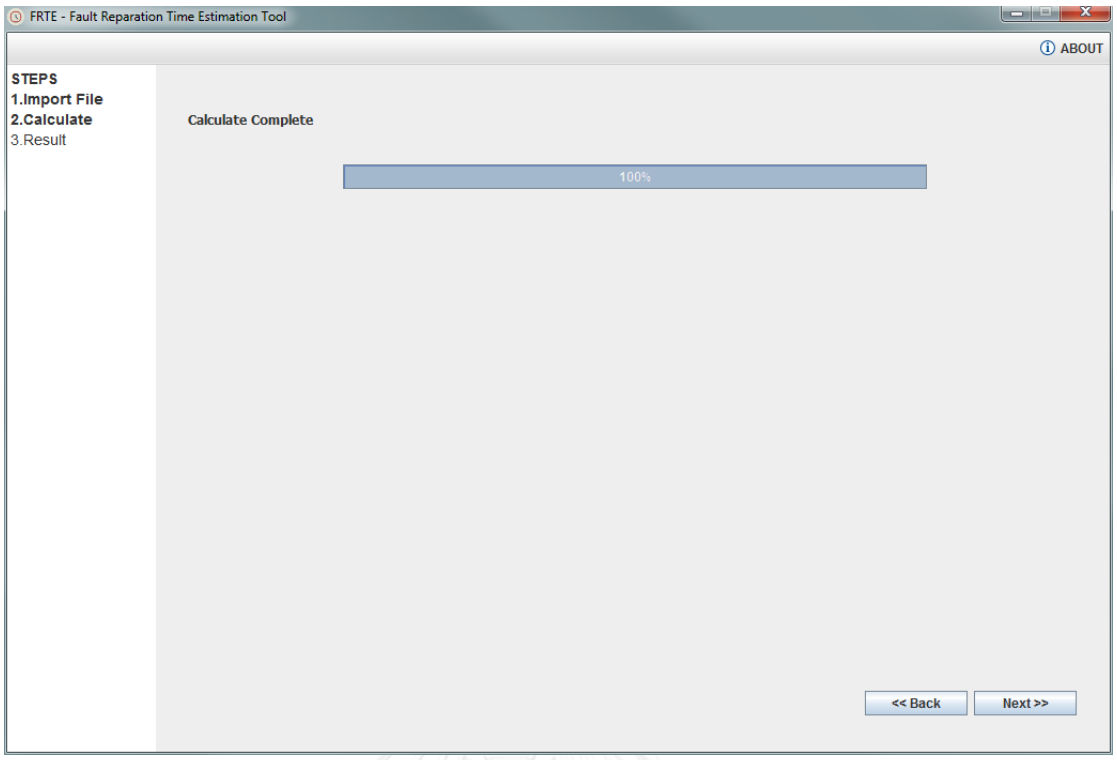
รูปที่ 23 หน้าจอของข้อความแจ้งเตือนเมื่อเลือกไฟล์นำเข้าไม่ถูกต้อง

2) ส่วนของการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

เมื่อผู้ใช้งานกดปุ่ม “Calculate” ระบบจะแสดงแถบสถานะของการคำนวณซึ่งสามารถแสดงได้ดังรูปที่ 24 เมื่อระบบคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแต่ละแบบจำลองเสร็จเรียบร้อยแล้ว ระบบจะแสดงข้อความว่า “Calculate Complete” และสถานะของการคำนวณนั้นจะเป็น 100 เปอร์เซ็นต์เพื่อเป็นการบอกว่าการคำนวณเสร็จเรียบร้อยแล้ว จากนั้นผู้ใช้งานจะสามารถกดปุ่ม “Next” เพื่อไปยังหน้าการแสดงผลข้อมูล หรือผู้ใช้งานสามารถกดปุ่ม “Back” เพื่อกลับไปยังหน้าของการนำเข้าไฟล์ข้อมูลซึ่งแสดงได้ดังรูปที่ 25



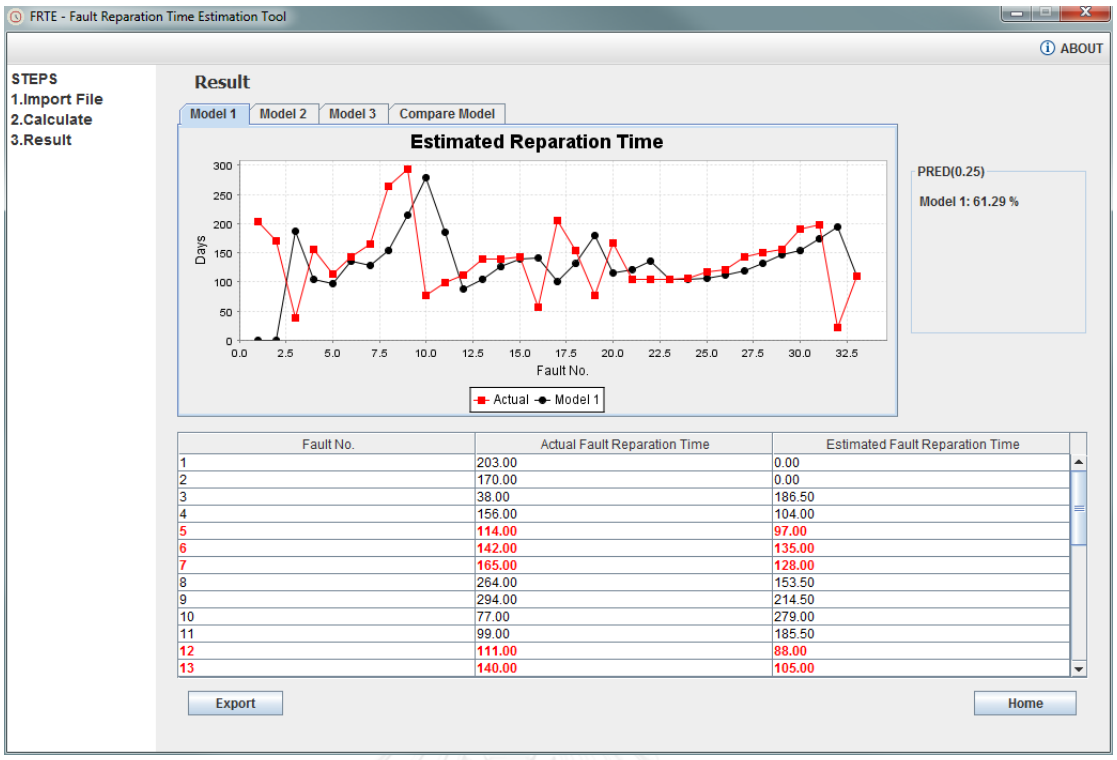
รูปที่ 24 หน้าจอแสดงสถานะของการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด



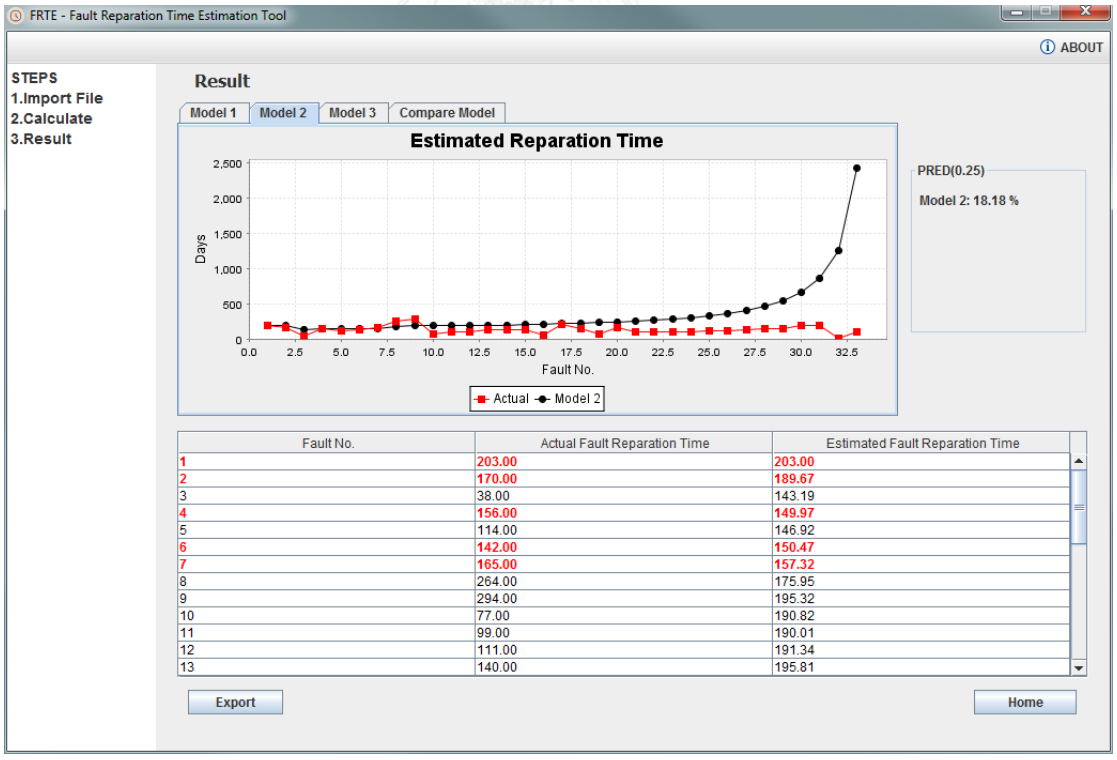
รูปที่ 25 หน้าจอแสดงสถานะของการคำนวณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดเสร็จสิ้น

3) ส่วนของการแสดงผลลัพธ์การประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

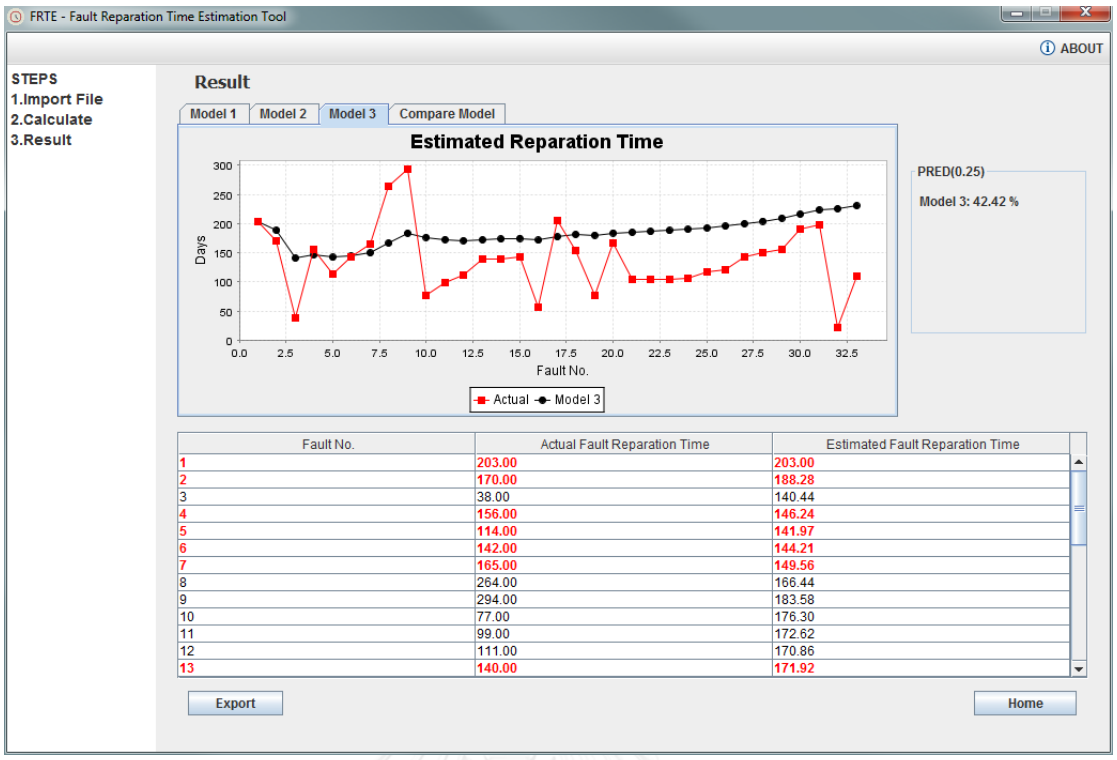
เมื่อผู้ใช้งานกดปุ่ม “Next” ระบบจะแสดงผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 แสดงดัง ซึ่งผู้ใช้งานสามารถคลิกที่แถบเพื่อเลือกดูผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 แบบจำลองที่ 3 และเปรียบเทียบแบบจำลองทั้งหมดได้ โดยระบบจะแสดงผลในรูปแบบตารางผลลัพธ์และแสดงผลในรูปแบบของกราฟเส้น หากผู้ใช้งานคลิกเลือกแถบของผลการประมาณค่าของแบบจำลองที่ 2 จะสามารถแสดงตัวอย่างได้ดัง หากผู้ใช้งานคลิกเลือกแถบของผลการประมาณค่าของแบบจำลองที่ 3 จะสามารถแสดงตัวอย่างได้ดัง และหากผู้ใช้งานคลิกเลือกแถบของการเปรียบเทียบผลการประมาณค่าของแบบจำลองทั้งหมดจะสามารถแสดงตัวอย่างได้ดัง โดยที่ในการแสดงผลนั้นหากผู้ใช้งานเลือกรูปแบบการประมาณเป็นพิจารณามาตรวัด ระบบจะแสดงผลการประมาณทั้ง 3 แบบจำลองและแสดงผลการเปรียบเทียบของทั้งสามแบบจำลอง แต่หากผู้ใช้งานเลือกรูปแบบการประมาณแบบไม่พิจารณามาตรวัด ระบบจะแสดงผลการประมาณของแบบจำลองที่ 1 แบบจำลองที่ 2 และแสดงผลการเปรียบเทียบของทั้งสองแบบจำลอง นอกจากนี้ระบบจะทำการแสดงผลของค่าการทำนายที่ระดับแอลของแต่ละแบบจำลองเพื่อให้ง่ายต่อการเปรียบเทียบผลในแต่ละแบบจำลอง



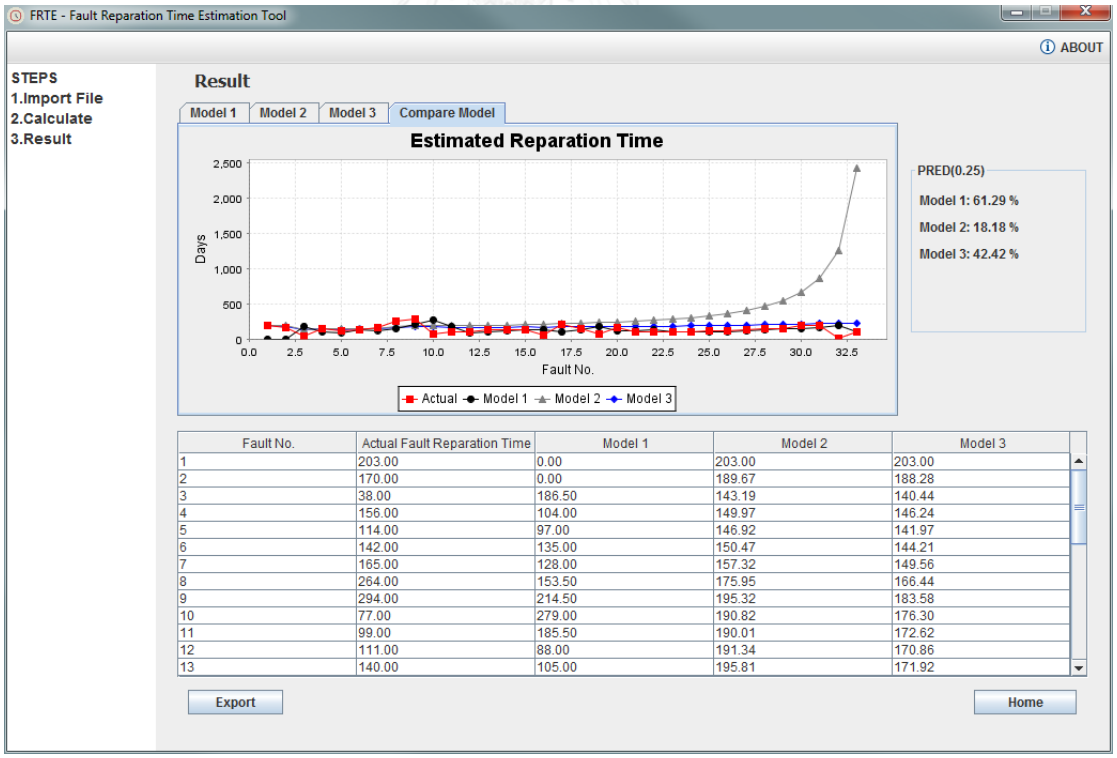
รูปที่ 26 ตัวอย่างผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1



รูปที่ 27 ตัวอย่างผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2



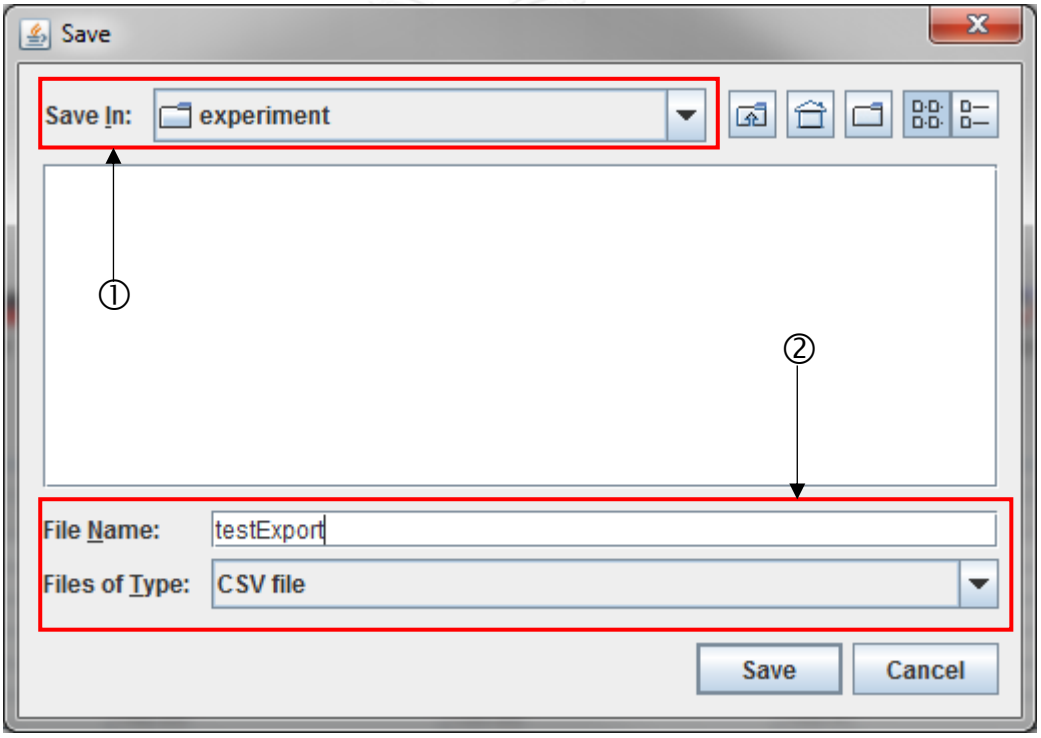
รูปที่ 28 ตัวอย่างผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3



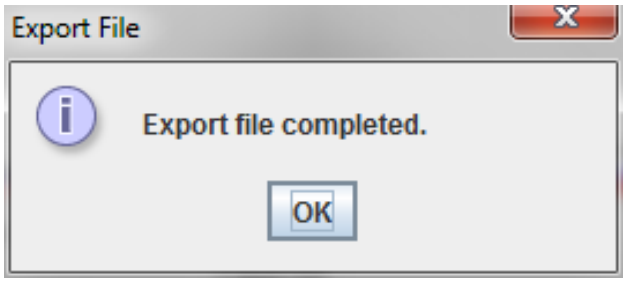
รูปที่ 29 ตัวอย่างการเปรียบเทียบผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของทั้ง 3 แบบจำลอง

4) ส่วนของการนำออกข้อมูลหรือบันทึกผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

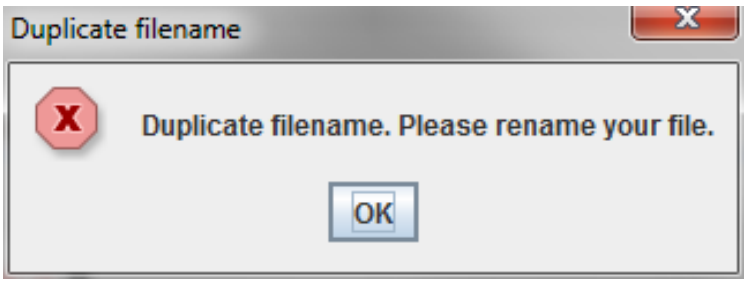
เมื่อผู้ใช้งานกดปุ่ม “Export” ระบบจะแสดงกล่องโต้ตอบเพื่อให้ผู้ใช้งานเลือกที่อยู่ที่จะบันทึกข้อมูล (หมายเลข 1) และตั้งชื่อไฟล์ที่จะทำการบันทึก (หมายเลข 2) ดังรูปที่ 30 เมื่อผู้ใช้งานกดบันทึกไฟล์สำเร็จระบบจะแจ้งว่า “Export file completed” ดังรูปที่ 31 หากผู้ใช้งานทำการบันทึกโดยตั้งชื่อไฟล์ที่มีอยู่แล้ว ระบบจะทำการแจ้งเตือนว่า “Duplicate filename. Please rename your file.” เพื่อให้ทำการตั้งชื่อใหม่ให้ไม่ซ้ำกับชื่อเดิมที่มีอยู่ในที่อยู่เดียวกันดังรูปที่ 32 และเมื่อบันทึกไฟล์สำเร็จแล้วจะได้ไฟล์ผลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบของตารางซึ่งมีนามสกุลเป็น .csv และไฟล์รูปภาพกราฟเส้นซึ่งมีนามสกุลเป็น .PNG ดังรูปที่ 33



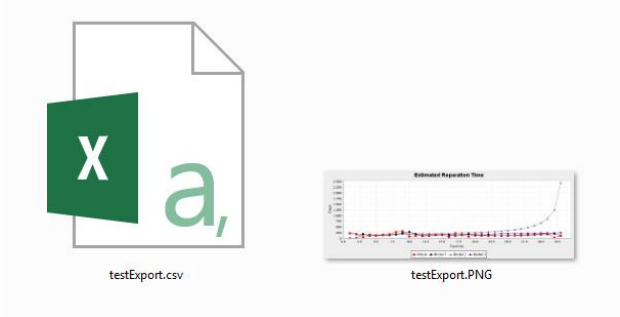
รูปที่ 30 ตัวอย่างการบันทึกข้อมูลผลลัพธ์การประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด



รูปที่ 31 ข้อความแจ้งเตือนเมื่อบันทึกข้อมูลสำเร็จ



รูปที่ 32 ข้อความแจ้งเตือนเมื่อตั้งชื่อไฟล์ซ้ำ



รูปที่ 33 ไฟล์ที่ได้จากการบันทึกผลการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

5) ส่วนของการแสดงชื่อ รุ่นของเครื่องมือ และแสดงรายละเอียดของผู้พัฒนาเครื่องมือ

เมื่อผู้ใช้งานกดปุ่ม “ABOUT” จากหน้าจอหลัก ระบบจะทำการแสดงชื่อของเครื่องมือ รุ่นของเครื่องมือ และรายละเอียดของผู้พัฒนาเครื่องมือซึ่งแสดงดังรูปที่ 34

รูปที่ 34 ส่วนของการแสดงรายละเอียดของเครื่องมือ

5.4 การทดสอบเครื่องมือ

การทดสอบเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด มีวัตถุประสงค์เพื่อค้นหาข้อบกพร่องของเครื่องมือที่ผู้วิจัยได้พัฒนา ซึ่งยังเป็นการบอกว่าเครื่องมือที่ผู้วิจัยได้พัฒนานั้นสามารถทำงานได้อย่างมีประสิทธิภาพและตรงตามความต้องการที่ได้ระบุไว้ ซึ่งในการทดสอบนั้นจะเป็นการทดสอบการทำงานว่าสามารถทำงานได้อย่างถูกต้องหรือไม่ ซึ่งในการ

ทดสอบนี้จะสร้างกรณีทดสอบทั้งหมด 10 กรณี ดังที่ได้ระบุไว้ด้านล่าง และสามารถสรุปผลการทดสอบเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้ตารางที่ 34

1) กรณีทดสอบการนำเข้าไฟล์ข้อมูลในกรณีพิจารณามาตรวัด (ImportFile_TC01) สามารถแสดงได้ดังตารางที่ 24

2) กรณีทดสอบการนำเข้าไฟล์ข้อมูลในกรณีไม่พิจารณามาตรวัด (ImportFile_TC02) สามารถแสดงได้ดังตารางที่ 25

3) กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 (Estimate_TC01) สามารถแสดงได้ดังตารางที่ 26

4) กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 (Estimate_TC02) สามารถแสดงได้ดังตารางที่ 27

5) กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 (Estimate_TC03) สามารถแสดงได้ดังตารางที่ 28

6) กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้น (Result_TC01) สามารถแสดงได้ดังตารางที่ 29

7) กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้น (Result_TC02) สามารถแสดงได้ดังตารางที่ 30

8) กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้น (Result_TC03) สามารถแสดงได้ดังตารางที่ 31

9) กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของการเปรียบเทียบแบบจำลองในรูปแบบตารางและกราฟเส้น (Result_TC04) สามารถแสดงได้ดังตารางที่ 32

10) กรณีทดสอบการนำออกหรือบันทึกไฟล์ในรูปแบบตารางและรูปแบบรูปภาพ (ExportFile_TC01) สามารถแสดงได้ดังตารางที่ 33

ตารางที่ 24 กรณีทดสอบการนำเข้าไฟล์ข้อมูลในกรณีพิจารณามาตรวัด

รหัสทดสอบ	ImportFile_TC01
ชื่อกรณีทดสอบ	การนำเข้าไฟล์ข้อมูลในกรณีพิจารณามาตรวัด
ฟังก์ชันที่ทดสอบ	นำเข้าไฟล์ข้อมูลในกรณีพิจารณามาตรวัด
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถนำเข้าไฟล์รูปแบบตารางในกรณีพิจารณามาตรวัดได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ผู้ใช้งานต้องเลือกรูปแบบการประมาณเป็นพิจารณามาตรวัด (With Instability Metric)
ข้อมูลนำเข้า	ไฟล์ข้อมูลที่ใช้ในการทดสอบแบบจำลองซึ่งประกอบด้วยหมายเลขข้อผิดพลาด ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) ระบบแสดงข้อมูลบนหน้าจอซึ่งเป็นข้อมูลที่ได้จากการนำเข้าไฟล์ข้อมูล
ลำดับการทดสอบ	1) ผู้ใช้งานทำการเลือกรูปแบบการประมาณเป็น With Instability Metric 2) ผู้ใช้งานนำเข้าไฟล์ข้อมูลที่ใช้ในการคำนวณหาค่าประมาณ 3) ระบบทำการตรวจสอบนามสกุลไฟล์ข้อมูลนำเข้า ซึ่งระบบจะให้สามารถเลือกไฟล์นำเข้าในรูปแบบตาราง (.csv) เท่านั้น 4) ระบบแสดงข้อมูลในไฟล์นำเข้าบนหน้าจอ 5) ผู้ใช้งานตรวจสอบความถูกต้องของไฟล์ข้อมูลนำเข้าว่าตรงตามที่ผู้ใช้งานต้องการนำเข้าหรือไม่
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 25 กรณีทดสอบการนำเข้าไฟล์ข้อมูลในกรณีไม่พิจารณามาตรวัด

รหัสทดสอบ	ImportFile_TC02
ชื่อกรณีทดสอบ	การนำเข้าไฟล์ข้อมูลในกรณีไม่พิจารณามาตรวัด
ฟังก์ชันที่ทดสอบ	นำเข้าไฟล์ข้อมูลในกรณีไม่พิจารณามาตรวัด
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถนำเข้าไฟล์รูปแบบตารางในกรณีไม่พิจารณามาตรวัดได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ผู้ใช้งานต้องเลือกรูปแบบการประมาณเป็นไม่พิจารณามาตรวัด (Without Instability Metric)
ข้อมูลนำเข้า	ไฟล์ข้อมูลที่ใช้ในการทดสอบแบบจำลองซึ่งประกอบด้วยหมายเลขข้อผิดพลาด และระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) ระบบแสดงข้อมูลบนหน้าจอซึ่งเป็นข้อมูลที่ได้จากการนำเข้าไฟล์ข้อมูล
ลำดับการทดสอบ	1) ผู้ใช้งานทำการเลือกรูปแบบการประมาณเป็น Without Instability Metric 2) ผู้ใช้งานนำเข้าไฟล์ข้อมูลที่ใช้ในการคำนวณหาค่าประมาณ 3) ระบบทำการตรวจสอบนามสกุลไฟล์ข้อมูลนำเข้า ซึ่งระบบจะให้สามารถเลือกไฟล์นำเข้าในรูปแบบตาราง (.csv) เท่านั้น 4) ระบบแสดงข้อมูลในไฟล์นำเข้าบนหน้าจอ 5) ผู้ใช้งานตรวจสอบความถูกต้องของไฟล์ข้อมูลนำเข้าว่าตรงตามที่ผู้ใช้งานต้องการนำเข้าหรือไม่
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 26 กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1

รหัสทดสอบ	Estimate_TC01
ชื่อกรณีทดสอบ	การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1
ฟังก์ชันที่ทดสอบ	คำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ ImportFile_TC01 และ ImportFile_TC02
ข้อมูลนำเข้า	ข้อมูลจากไฟล์นำเข้าซึ่งประกอบด้วยหมายเลขข้อผิดพลาด และระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ได้
ลำดับการทดสอบ	1) ผู้ใช้งานเลือกขั้นตอนถัดไปจากหน้าจอการนำเข้าข้อมูล 2) ระบบจะทำการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 3) ระบบแสดงแถบสถานการณ์คำนวณ 4) ระบบจะบันทึกข้อมูลเพื่อนำไปแสดงผลในหน้าจอแสดงผล 5) ระบบจะแสดงข้อความว่าคำนวณเสร็จเรียบร้อย (Calculate Complete)
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 27 กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2

รหัสทดสอบ	Estimate_TC02
ชื่อกรณีทดสอบ	การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2
ฟังก์ชันที่ทดสอบ	คำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ ImportFile_TC01 และ ImportFile_TC02
ข้อมูลนำเข้า	ข้อมูลจากไฟล์นำเข้าซึ่งประกอบด้วยหมายเลขข้อผิดพลาด และระยะเวลาจริงที่ใช้ในการแก้ไข
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ได้
ลำดับการทดสอบ	1) ผู้ใช้งานเลือกขั้นตอนถัดไปจากหน้าจอการนำเข้าข้อมูล 2) ระบบจะทำการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 3) ระบบแสดงแถบสถานการณ์คำนวณ 4) ระบบจะบันทึกข้อมูลเพื่อนำไปแสดงผลในหน้าจอแสดงผล 5) ระบบจะแสดงข้อความว่าคำนวณเสร็จเรียบร้อย (Calculate Complete)
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

Chulalongkorn University

ตารางที่ 28 กรณีทดสอบการคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3

รหัสทดสอบ	Estimate_TC03
ชื่อกรณีทดสอบ	การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3
ฟังก์ชันที่ทดสอบ	คำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถคำนวณค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ ImportFile_TC01
ข้อมูลนำเข้า	ข้อมูลจากไฟล์นำเข้าซึ่งประกอบด้วยหมายเลขข้อผิดพลาด ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาด และค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ได้
ลำดับการทดสอบ	1) ผู้ใช้งานเลือกขั้นตอนถัดไปจากหน้าจอการนำเข้าข้อมูล 2) ระบบจะทำการคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 3) ระบบแสดงแถบสถานการณ์คำนวณ 4) ระบบจะบันทึกข้อมูลเพื่อนำไปแสดงผลในหน้าจอแสดงผล 5) ระบบจะแสดงข้อความว่าคำนวณเสร็จเรียบร้อย (Calculate Complete)
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 29 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้น

รหัสทดสอบ	Result_TC01
ชื่อกรณีทดสอบ	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้น
ฟังก์ชันที่ทดสอบ	แสดงผลค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้น
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถแสดงผลค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้นได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ Estimate_TC01
ข้อมูลนำเข้า	ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถแสดงผลลัพธ์ของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 โดยแสดงผลทางหน้าจอในรูปแบบของตารางและกราฟเส้นได้
ลำดับการทดสอบ	1) ระบบทำเรียกข้อมูลของการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 มาแสดงผลทางหน้าจอในรูปแบบตาราง 2) ระบบสร้างกราฟเส้นจากข้อมูลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 3) ระบบแสดงผลกราฟเส้นทางหน้าจอ 4) ระบบแสดงผลค่าการทำนายที่ระดับแอลของแบบจำลองที่ 1
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 30 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้น

รหัสทดสอบ	Result_TC02
ชื่อกรณีทดสอบ	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้น
ฟังก์ชันที่ทดสอบ	แสดงผลค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้น
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถแสดงผลค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้นได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ Estimate_TC02
ข้อมูลนำเข้า	ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถแสดงผลลัพธ์ของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 โดยแสดงผลทางหน้าจอในรูปแบบของตารางและกราฟเส้นได้
ลำดับการทดสอบ	1) ระบบทำเรียกข้อมูลของการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 มาแสดงผลทางหน้าจอในรูปแบบตาราง 2) ระบบสร้างกราฟเส้นจากข้อมูลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 3) ระบบแสดงผลกราฟเส้นทางหน้าจอ 4) ระบบแสดงผลค่าการทำนายที่ระดับแอลของแบบจำลองที่ 2
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 31 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้น

รหัสทดสอบ	Result_TC03
ชื่อกรณีทดสอบ	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้น
ฟังก์ชันที่ทดสอบ	แสดงผลค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้น
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถแสดงผลค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้นได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ Estimate_TC03
ข้อมูลนำเข้า	ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถแสดงผลลัพธ์ของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 โดยแสดงผลทางหน้าจอในรูปแบบของตารางและกราฟเส้นได้
ลำดับการทดสอบ	1) ระบบทำเรียกข้อมูลของการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 มาแสดงผลทางหน้าจอในรูปแบบตาราง 2) ระบบสร้างกราฟเส้นจากข้อมูลการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 3) ระบบแสดงผลกราฟเส้นทางหน้าจอ 4) ระบบแสดงผลค่าการทำนายที่ระดับแอลของแบบจำลองที่ 3
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 32 กรณีทดสอบการแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของการเปรียบเทียบแบบจำลองในรูปแบบตารางและกราฟเส้น

รหัสทดสอบ	Result_TC04
ชื่อกรณีทดสอบ	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของการเปรียบเทียบแบบจำลองในรูปแบบตารางและกราฟเส้น
ฟังก์ชันที่ทดสอบ	แสดงผลการเปรียบเทียบค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองทั้งหมดในรูปแบบตารางและกราฟเส้น
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถแสดงผลการเปรียบเทียบค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองทั้งหมดในรูปแบบตารางและกราฟเส้นได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ Estimate_TC01, Estimate_TC02 และ Estimate_TC03
ข้อมูลนำเข้า	ค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองทั้งหมด
ข้อมูลนำออก	ไม่มี
ผลลัพธ์ที่คาดหวัง	1) เครื่องมือสามารถแสดงผลลัพธ์ของการเปรียบเทียบค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองทั้งหมด โดยแสดงผลทางหน้าจอในรูปแบบของตารางและกราฟเส้นได้
ลำดับการทดสอบ	1) ระบบทำเรียกข้อมูลของการเปรียบเทียบค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองทั้งหมด มาแสดงผลทางหน้าจอในรูปแบบตาราง 2) ระบบสร้างกราฟเส้นจากข้อมูลการเปรียบเทียบค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบแบบจำลองทั้งหมด 3) ระบบแสดงผลกราฟเส้นทางหน้าจอ 4) ระบบแสดงผลค่าการทำนายที่ระดับแอลของแบบจำลองทั้งหมด
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 33 กรณีทดสอบการนำออกหรือบันทึกไฟล์ในรูปแบบตารางและรูปแบบรูปภาพ

รหัสทดสอบ	ExportFile_TC01
ชื่อกรณีทดสอบ	การนำออกหรือบันทึกไฟล์ในรูปแบบตารางและรูปแบบรูปภาพ
ฟังก์ชันที่ทดสอบ	นำออกหรือบันทึกไฟล์ในรูปแบบตารางและรูปแบบรูปภาพ
จุดมุ่งหมายในการทดสอบ	เพื่อทำการทดสอบว่า เครื่องมือสามารถนำออกหรือบันทึกไฟล์ผลลัพธ์จากการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตารางและรูปแบบรูปภาพได้หรือไม่
เงื่อนไขเริ่มการทดสอบ	ต้องผ่านกรณีทดสอบ Result_TC01,Result_TC02, Result_TC03 และ Result_TC04
ข้อมูลนำเข้า	ผลลัพธ์จากการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตารางและรูปแบบรูปภาพ
ข้อมูลนำออก	ไฟล์รูปแบบตาราง (.csv) และไฟล์รูปแบบรูปภาพ (.PNG)
ผลลัพธ์ที่คาดหวัง	1) ระบบแจ้งเตือนว่าการนำออกหรือบันทึกไฟล์เสร็จเรียบร้อย (Export file completed) 2) เครื่องมือสามารถนำออกหรือบันทึกไฟล์ผลลัพธ์การประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตาราง (.csv) และไฟล์รูปแบบรูปภาพ (.PNG)
ลำดับการทดสอบ	1) ผู้ใช้งานเลือกนำออกหรือบันทึกข้อมูล (Export) จากหน้าจอ 2) ผู้ใช้งานเลือกที่อยู่ที่ต้องการบันทึกไฟล์ 3) ระบบจะทำการบันทึกไฟล์ผลลัพธ์การประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดในรูปแบบตาราง (.csv) และไฟล์รูปแบบรูปภาพ (.PNG) โดยบันทึกไว้ในที่อยู่ที่ผู้ใช้งานระบุไว้ 4) ระบบแสดงข้อความแจ้งเตือนว่าการนำออกหรือบันทึกไฟล์เสร็จเรียบร้อย (Export file completed)
สรุปผลการทดสอบ	☑ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 34 สรุปผลการทดสอบเครื่องมือสำหรับการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

รหัสกรณีทดสอบ	**ชื่อกรณีทดสอบ**	**ผลการทดสอบ**
ImportFile_TC01	การนำเข้าไฟล์ข้อมูลในกรณีพิจารณามาตรวัด	ผ่าน
ImportFile_TC02	การนำเข้าไฟล์ข้อมูลในกรณีไม่พิจารณามาตรวัด	ผ่าน
Estimate_TC01	การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1	ผ่าน
Estimate_TC02	การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2	ผ่าน
Estimate_TC03	การคำนวณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3	ผ่าน
Result_TC01	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในรูปแบบตารางและกราฟเส้น	ผ่าน
Result_TC02	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 2 ในรูปแบบตารางและกราฟเส้น	ผ่าน
Result_TC03	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ในรูปแบบตารางและกราฟเส้น	ผ่าน
Result_TC04	การแสดงผลค่าการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของการเปรียบเทียบแบบจำลองในรูปแบบตารางและกราฟเส้น	ผ่าน
ExportFile_TC01	การนำออกหรือบันทึกไฟล์ในรูปแบบตารางและรูปแบบรูปภาพ	ผ่าน

CHULALONGKORN UNIVERSITY

บทที่ 6
สรุปผลการวิจัย

ในบทนี้เป็นบทสุดท้ายของวิทยานิพนธ์ฉบับนี้ โดยจะกล่าวถึงการสรุปผลการวิจัยที่ได้ดำเนินการ ข้อจำกัดของงานวิจัย ทั้งนี้ยังได้นำเสนอแนวทางในการทำงานวิจัยในอนาคต รวมถึงบทความวิชาการที่ได้รับการตีพิมพ์ ซึ่งมีรายละเอียดดังนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการบำรุงรักษาซอฟต์แวร์ เพื่อสามารถประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ผู้ใช้งานได้แจ้งเข้ามาให้ผู้บำรุงรักษาทำการแก้ไข เพราะหากไม่สามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้ อาจจะส่งผลต่อการประมาณงบประมาณและทรัพยากรต่างๆ ขององค์กรได้

การบำรุงรักษาซอฟต์แวร์ คือกิจกรรมทั้งหมดที่จำเป็นในสนับสนุน ต้นทุนและประสิทธิภาพให้กับซอฟต์แวร์ ซึ่งกิจกรรมต่างๆ เป็นกิจกรรมที่ดำเนินหลังการส่งมอบรวมทั้งการปรับเปลี่ยนแก้ไขซอฟต์แวร์,การฝึกอบรม และการดำเนินงานให้ความช่วยเหลือ ซึ่งการบำรุงรักษาซอฟต์แวร์จะเกิดขึ้นเมื่อมีข้อผิดพลาดจากการทำงานของตัวซอฟต์แวร์นั้นๆ หรือเกิดจากความต้องการของผู้ใช้งานที่มีการเปลี่ยนแปลง หรือมีความต้องการเพิ่มเติมจากที่เคยได้ระบุไว้ในกระบวนการระบุความต้องการของซอฟต์แวร์ เมื่อมีข้อผิดพลาดเกิดขึ้นในผลิตภัณฑ์ซอฟต์แวร์ ผู้บำรุงรักษาจึงจำเป็นต้องเร่งแก้ไขให้ซอฟต์แวร์สามารถใช้งานได้ตามปกติ ข้อผิดพลาดที่เกิดขึ้นในซอฟต์แวร์สามารถเกิดจากข้อผิดพลาดของมนุษย์หรือผู้พัฒนาซอฟต์แวร์ ซึ่งส่งผลให้ซอฟต์แวร์ไม่สามารถทำงานได้ตรงตามความต้องการของผู้ใช้งาน ข้อผิดพลาดต่างๆ เหล่านี้สามารถทำให้ซอฟต์แวร์เกิดการล้มเหลว ก็ต่อเมื่อซอฟต์แวร์ส่วนที่มีข้อผิดพลาดถูกเรียกใช้งาน ข้อผิดพลาดที่ถูกค้นพบในรหัสต้นฉบับ จะเรียกว่าข้อผิดพลาดที่เกิดขึ้นในซอฟต์แวร์หรือบั๊ก (Bugs)

เมื่อเกิดข้อผิดพลาดเกิดขึ้นในซอฟต์แวร์ ผู้ใช้งานทำการแจ้งการแก้ไขข้อผิดพลาดให้กับทีมผู้บำรุงรักษา ผู้บำรุงรักษาซอฟต์แวร์จะทำการแก้ไขข้อผิดพลาดในซอฟต์แวร์ หลังจากนั้นจะทำให้ผู้บำรุงรักษาสามารถทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นๆ เสร็จสิ้น แต่เมื่อผู้ใช้งานแจ้งการแก้ไขข้อผิดพลาดมายังทีมผู้บำรุงรักษา ผู้ใช้งานต้องการทราบระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นๆ เพื่อนำไปใช้ในการวางแผนการทำงานในส่วนอื่นๆ หรือใช้ในการทำสัญญาการบำรุงรักษาซอฟต์แวร์ อย่างไรก็ตามผู้บำรุงรักษาจะทราบระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดก็ต่อเมื่อแก้ไขข้อผิดพลาดนั้นเสร็จสิ้น ดังนั้นจึงจำเป็นต้องหาวิธีการประมาณระยะเวลาที่ใช้ในการแก้ไข

ข้อผิดพลาด ซึ่งหากผู้บำรุงรักษาสามารถบอกระยะเวลาประมาณที่ใช้ในการแก้ไขข้อผิดพลาดได้ จะช่วยให้องค์กรหรือหน่วยงานจัดสรรทรัพยากรและงบประมาณไปใช้ในการบำรุงรักษาหรือใช้งานในด้านอื่นๆ ได้อย่างเหมาะสม

งานวิจัยนี้นำเสนอแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดทั้งหมด 3 แบบจำลอง คือ แบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล แบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ ทำการทดสอบแบบจำลองโดยรวบรวมชุดโปรแกรมที่พัฒนาด้วยภาษาจาวาจากเว็บไซต์ของโปรแกรมโอเพนซอร์ซทั้งหมด 10 โปรแกรม ซึ่งพิจารณาเลือกโปรแกรมที่มีข้อผิดพลาดมากกว่า 1 ข้อผิดพลาด โดยเป็นโปรแกรมที่มีการแจ้งข้อมูลการเกิดข้อผิดพลาดและรายละเอียดของข้อผิดพลาดเข้ามายังระบบ และผู้บำรุงรักษาทำการแก้ไขข้อผิดพลาดเสร็จเรียบร้อยแล้ว จากนั้นทำการแยกประเภทของข้อผิดพลาดเป็น 5 ประเภท จากนั้นทำการเปรียบเทียบแบบจำลองโดยใช้การทดสอบนัยสำคัญด้วยสถิติทดสอบที (Paired t-test) ระดับความเชื่อมั่น 95% และประเมินผลแบบจำลองโดยใช้กรคำนวณความผิดพลาดสัมพัทธ์และการทำนายระดับแอล โดยใช้การทำนายที่ระดับ 0.25 ซึ่งมีความหมายว่าค่าประมาณที่ได้จากแบบจำลองสามารถมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เป็นจำนวน 75 เปอร์เซ็นต์ขึ้นไปจากการประมาณค่าทั้งหมดของแบบจำลอง

เมื่อทำการเปรียบเทียบการประเมินผลแบบจำลองทั้ง 3 แบบจำลอง พบว่าแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์นั้น มีประสิทธิภาพและประมาณได้ใกล้เคียงกับค่าจริงมากที่สุด อันดับสองเป็นแบบจำลองที่ 1 สร้างแบบจำลองด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล และอันดับสุดท้ายเป็นแบบจำลองที่ 2 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล

จากงานวิจัยนี้สามารถเลือกแบบจำลองที่ดีที่สุดในการนำไปประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด นั่นก็คือแบบจำลองที่ 3 สร้างแบบจำลองด้วยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ ซึ่งสามารถอธิบายได้ว่าความไม่เสถียรของซอฟต์แวร์มีผลต่อความถูกต้องของการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด และหากทำการพิจารณาประเภทของข้อผิดพลาดจะทำให้แบบจำลองนั้นมีความถูกต้องมากขึ้น แต่อย่างไรก็ตามค่าการทำนายที่ระดับแอลของแบบจำลองที่ 3 บางค่านั้นมีค่าน้อยกว่า 75% ซึ่งยังไม่สามารถยอมรับแบบจำลองนี้ได้ สาเหตุอาจมาจากไม่ได้นำปัจจัยอื่นๆ มาร่วมพิจารณาในการสร้าง

แบบจำลองด้วย เช่น ความซับซ้อนของซอฟต์แวร์ จำนวนบรรทัดของรหัสต้นฉบับ หรือระดับความรุนแรงของข้อผิดพลาด เป็นต้น

6.2 ข้อจำกัดของงานวิจัย

ข้อจำกัดของงานวิจัยนี้ประกอบด้วย

1) แบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ที่สร้างโดยการประยุกต์ใช้แบบจำลองพื้นฐานเกี่ยวกับการแจกแจงแบบเอกซ์โปแนนเชียล (แบบจำลองที่ 1) สามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้ ก็ต่อเมื่อมีข้อผิดพลาดเกิดขึ้นและถูกแก้ไขเรียบร้อยแล้วอย่างน้อย 2 ข้อผิดพลาด

2) เครื่องมือสนับสนุนการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้พัฒนาขึ้นนั้น สามารถนำเข้าไฟล์ในรูปแบบตารางที่มีนามสกุลเป็น .csv เท่านั้น

3) การสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของงานวิจัยนี้ได้ใช้ข้อมูลจากเว็บไซต์ของซอฟต์แวร์โอเพนซอร์ซ ซึ่งในวิทยานิพนธ์นี้ข้อผิดพลาดที่เกิดขึ้นในซอฟต์แวร์ไม่ได้คำนึงถึงจำนวนคนที่ใช้ในการแก้ไขข้อผิดพลาด

6.3 งานวิจัยในอนาคต

สามารถแบ่งการทำงานวิจัยในอนาคตได้ดังนี้

1) ปรับปรุงแบบจำลองโดยใช้ปัจจัยอื่นๆ มาพิจารณาในการประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด เช่น ระดับความรุนแรงของข้อผิดพลาด (severity of fault) ความซับซ้อนของซอฟต์แวร์ (software complexity) จำนวนบรรทัดของซอฟต์แวร์ (lines of code) ความเสถียรของซอฟต์แวร์ (software stability) เป็นต้น

2) ปรับปรุงแบบจำลองเพื่อให้สามารถประมาณค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ในขั้นตอนของการพัฒนาซอฟต์แวร์

3) เปรียบเทียบแบบจำลองที่ได้นำเสนอในงานวิจัยนี้กับแบบจำลองที่สร้างโดยการประยุกต์ใช้วิธี Prequential likelihood ratio

6.4 ผลงานการตีพิมพ์จากวิทยานิพนธ์

1) หัวเรื่องงานวิจัยที่ตีพิมพ์ชื่อ “Estimation of Fault Reparation Time in Software Maintenance Phase” ในการประชุมวิชาการระดับนานาชาติ “The 7th International Conference on Information Science and Applications (ICISA 2016)” ซึ่งจัดขึ้น ณ นครโฮจิมินห์ ประเทศเวียดนาม ระหว่างวันที่ 15 – 18 กุมภาพันธ์ พ.ศ. 2559

2) หัวเรื่องงานวิจัยที่ตีพิมพ์ชื่อ “Fault Reparation Time Estimation Based on Exponential Distribution Function and Software Instability Metric” ในการประชุมวิชาการระดับนานาชาติ “The 6th International Conference on Electronics Information and Emergency Communication (ICEIEC 2016)” ซึ่งจัดขึ้น ณ กรุงปักกิ่ง สาธารณรัฐประชาชนจีน ระหว่างวันที่ 17 – 19 มิถุนายน พ.ศ. 2559

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รายการอ้างอิง

1. Honda, K., et al. *Predicting Time Range of Development Based on Generalized Software Reliability Model*. in *Software Engineering Conference (APSEC),* 2014 21*st Asia-Pacific*. 2014.
2. Bhattacherjee, V. *The Soft Computing Approach to Program Development Time Estimation*. in *Information Technology,* 2006. *ICIT*'06. 9*th International Conference on*. 2006. IEEE.
3. Garcia-Diaz, N., et al. *Software development time estimation based on a new Neuro-fuzzy approach*. in *Information Systems and Technologies (CISTI),* 2015 10*th Iberian Conference on*. 2015. IEEE.
4. Giger, E., M. Pinzger, and H. Gall. *Predicting the fix time of bugs*. in *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. 2010. Cape Town, South Africa: ACM.
5. Hongyu, Z., G. Liang, and S. Versteeg. *Predicting bug-fixing time: An empirical study of commercial software projects*. in *Software Engineering (ICSE),* 2013 35*th International Conference on*. 2013.
6. AbdelMoez, W., M. Kholief, and F.M. Elsalmy. *Bug fix-time prediction model using Naive Bayes classifier*. in *Computer Theory and Applications (ICCTA),* 2012 22*nd International Conference on*. 2012.
7. AbdelMoez, W., M. Kholief, and F.M. Elsalmy. *Improving bug fix-time prediction model by filtering out outliers*. in *Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE),* 2013 *International Conference on*. 2013.
8. Bhattacharya, P. and I. Neamtiu. *Bug-fix time prediction models: can we do better?* in *Proceedings of the 8th Working Conference on Mining Software Repositories*. 2011. Waikiki, Honolulu, HI, USA: ACM.
9. Kim, S. and J. E. James Whitehead. *How long did it take to fix bugs?* in *Proceedings of the 2006 international workshop on Mining software repositories*. 2006. Shanghai, China: ACM.

10. Yi, J., et al. *Markov Reliability Model Based on Error Classification*. in *Parallel Architectures, Algorithms and Programming (PAAP),* 2012 *Fifth International Symposium on*. 2012.
11. Roy, G.S.M.a.P., *Modified Jelinski-Moranda Software Reliability Model with Imperfect Debugging Phenomenon.* International Journal of Computer Applications, 2012. 48(18): p. 38-46.
12. Jelinski, Z. and P. Moranda, *SOFTWARE RELIABILITY RESEARCH*, in *Statistical Computer Performance Evaluation*, W. Freiberger, Editor. 1972, Academic Press. p. 465-484.
13. Martin, R., *OO design quality metrics.* An analysis of dependencies, 1994.
14. *International Standard - ISO/IEC* 14764 *IEEE Std* 14764-2006 *Software Engineering* 2013*; Software Life Cycle Processes* 2013*; Maintenance.* ISO/IEC 14764:2006 (E) IEEE Std 14764-2006 Revision of IEEE Std 1219-1998), 2006: p. 1-46.
15. *Information technology - Software product quality: Quality model*, in *ISO/IEC FDIS* 9126-1. 2000.
16. *IEEE Standard Glossary of Software Engineering Terminology.* IEEE Std 610.12-1990, 1990: p. 1-84.
17. *IEEE Standard Classification for Software Anomalies.* IEEE Std 1044-1993, 1994.
18. Fenton, N. and S.L. Pfleeger, *Software metrics (2nd ed.): a rigorous and practical approach*. 1997: PWS Publishing Co. 638.
19. Michael, R.L., *Handbook of software reliability engineering*, ed. R.L. Michael. 1996: McGraw-Hill, Inc. 851.
20. Xie, M., Y.S. Dai, and K.L. Poh, *Computing System Reliability: Models and Analysis*. 2004: Springer.
21. *Probability, Statistics and Queuing Theory*. 2009: Prentice-Hall Of India Pvt. Limited.
22. *SourceForge Open Source Applications and Software Directory*. Available from: http://sourceforge.net.

Chulalongkorn University

23. *SourceCodeMetrics*. Available from: http://plugins.netbeans.org/plugin/42970/sourcecodemetrics.
24. David, H.A. and J.L. Gunnink, *The Paired t Test Under Artificial Pairing.* The American Statistician, 1997. 51(1): p. 9-12.
25. Rice, J.A., *Mathematical Statistics and Data Analysis*. 2006: Cengage Learning.
26. กัลยา, ว., การใช้ *SPSS for Windows* ในการวิเคราะห์ข้อมูล. พิมพ์ครั้งที่ 20. ed. 2012, กรุงเทพฯ :: ธรรมสาร.
27. Conte, S.D., H.E. Dunsmore, and V.Y. Shen, *Software engineering metrics and models*. 1986: Benjamin-Cummings Publishing Co., Inc. 396.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก

การแก้สมการเพื่อใช้ในการหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1

การแก้สมการเพื่อใช้ในการหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 1 ในงานวิจัยนี้ได้สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยสร้างแบบจำลองด้วยวิธีที่แตกต่างกัน ซึ่งในการสร้างแบบจำลองที่ 1 นั้นเป็นการการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียล ในการสร้างแบบจำลองนี้จะต้องทำการแก้สมการเพื่อให้สอดคล้องกับงานวิจัย ซึ่งการแก้สมการเพื่อหาค่าระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นสามารถอธิบายรายละเอียดได้ดังนี้

ในการแจกแจงแบบเอกซ์โปแนนเชียลนั้น ฟังก์ชันความน่าจะเป็นของ r_i (Probability density function (pdf) of r_i) หรือ $f_i(r_i)$ คือ

$$f_i(r_i) = \lambda_i e^{-\lambda_i r_i}$$

เมื่อ r_i คือระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i

λ_i คืออัตราการแก้ไขข้อผิดพลาดที่ i ด้วย r_i

เมื่อ $f_i(r_i) = \lambda_i e^{-\lambda_i r_i}$ จะได้ฟังก์ชันความน่าจะเป็นสะสม $F_i(r_i)$ ดังนี้

$$F_i(r_i) = \int_0^r f_i(r)dr$$

$$F_i(r_i) = \int_0^r \lambda_i e^{-\lambda_i r_i} dr$$

$$F_i(r_i) = \left[-e^{-\lambda_i r_i}\right]_0^r$$

$$F_i(r_i) = 1 - e^{-\lambda_i r_i}$$

ระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (r_i) จะถูกใช้ในการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (FRT_i) และระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด คือค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นหรือเรียกอีกอย่างว่า ค่าเฉลี่ยของ R (Expected value of R, $E(R)$) ซึ่งแสดงดังนี้

$$E(R_i) = \int_{i=1}^{n} r_i f_i(r_i)dr$$

ในการแจกแจงแบบเอกซ์โปแนนเชียลนั้น เมื่อ $f_i(r_i) = \lambda_i e^{-\lambda_i r_i}$ จะสามารถแก้สมการเพื่อหาค่าประมาณของระยะเวลาเฉลี่ยที่ใช้ในการแก้ไขข้อผิดพลาด โดยแสดงรายละเอียดดังนี้

$$FRT_i = E(R_i) = \int r_i \lambda_i e^{-\lambda_i r_i} dr$$

ให้ $y = \lambda_i r_i$

$$r_i = \frac{y}{\lambda_i}$$

$$dr = dy\frac{1}{\lambda_i}$$

จะได้ $E(R_i) = \frac{1}{\lambda_i}\int_0^\infty y\lambda_i e^{-y} dy$

ใช้การอินทิเกรตทีละส่วน (Integration by Parts) $\int udv = uv - \int vdu$

ให้ $u = y$ และ $du = dy$

$v = e^{-y}$ และ $dv = e^{-y}dv$

จะได้ว่า

$$\begin{aligned}\int ye^{-y}\,dv &= ye^{-y} - \int e^{-y}dy \\ &= ye^{-y} - e^{-y} \\ &= (y-1)e^{-y}\end{aligned}$$

ดังนั้น จะได้

$$\begin{aligned}E(R_i) &= \frac{1}{\lambda_i}[(y-1)e^{-y}]_0^\infty \\ &= \frac{1}{\lambda_i}\left[\left((\infty-1)e^{-\infty}\right) - \left((0-1)e^{0}\right)\right] \\ &= \frac{1}{\lambda_i}\end{aligned}$$

ดังนั้นค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดสามารถหาได้จาก

$$FRT_i = \frac{1}{\lambda_i}$$

จากสมการข้างต้น มีตัวแปรที่ไม่ทราบค่าในแบบจำลองคือ λ_i โดยในการหาค่าของ $\frac{1}{\lambda_i}$ นั้น สามารถหาได้จากค่าเฉลี่ยของระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ $i-1$ (r_{i-1}) และ $i-2$ (r_{i-2}) ดังนั้นสามารถคำนวณระยะเวลาเฉลี่ยที่ใช้ในการแก้ไขข้อผิดพลาดที่ i (FRT_i) ได้จาก

$$FRT_i = \frac{1}{\lambda_i} = \frac{r_{i-2} + r_{i-1}}{2}$$

เมื่อ FRT_i คือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 3,4,5, \ldots$

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3, \ldots$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ดังนั้นสามารถคำนวณอัตราการแก้ไขข้อผิดพลาดที่ i (λ_i) ได้จาก

$$\lambda_i = \frac{2}{r_{i-2} + r_{i-1}}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข
การแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดของแบบจำลองที่ 2

การแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดและทำไปใช้แทนค่าในแบบจำลองที่ 2 ในงานวิจัยนี้ได้สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยสร้างแบบจำลองด้วยวิธีที่แตกต่างกัน ซึ่งในการสร้างแบบจำลองที่ 2 นั้นเป็นการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล ในการสร้างแบบจำลองนี้จะต้องทำการแก้สมการเพื่อให้สอดคล้องกับงานวิจัย ซึ่งผลกระทบของแต่ละข้อผิดพลาดที่มีผลต่ออัตราการแก้ไขข้อผิดพลาดหรือ ϕ ในแบบจำลองที่ 2 นั้นยังไม่ทราบค่า จึงต้องทำการแก้สมการเพื่อใช้ในการหาค่า ϕ แล้วจากนั้นจะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้ ซึ่งสามารถอธิบายรายละเอียดในการแก้สมการเพื่อหาค่า ϕ ได้ดังนี้

ในแบบจำลองที่ 2 เป็นสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล โดยที่อัตราการแก้ไขข้อผิดพลาดที่ i (λ_i) สามารถหาได้จาก

$$\lambda = (N-(i-1))\phi$$

และในการแจกแจงแบบเอกซ์โปแนนเชียลนั้นค่า $f(r_i) = \lambda e^{-\lambda r_i}$ ดังนั้นจะได้

$$f(r_i) = \phi[N-(i-1)]\exp(-\phi[N-(i-1)]r_i)$$

จากนั้นใช้วิธี maximum-likelihood estimation (MLE) ในการประมาณ ϕ และ N โดยในแบบจำลองที่ 2 นั้นทราบค่าของจำนวนข้อผิดพลาดทั้งหมด (N) ดังนั้นจะสามารถหาค่า ϕ ได้โดยแสดงรายละเอียดดังนี้

$$\begin{aligned}
L(N,\phi) &= \prod_{i=1}^{n} \phi\,[N-(i-1)]\exp(-\phi[N-(i-1)]r_i) \\
&= \phi^n \prod_{i=1}^{n}[N-(i-1)]\exp(-\phi\sum_{i=1}^{n}[N-(i-1)]r_i) \\
&= \ln\left[\phi^n \prod_{i=1}^{n}[N-(i-1)]\exp\left(-\phi\sum_{i=1}^{n}[N-(i-1)]r_i\right)\right] \\
&= n\ln\phi + \sum_{i=1}^{n}\ln[N-(i-1)] - \phi\sum_{i=1}^{n}[N-(i-1)]r_i
\end{aligned}$$

$$\frac{\partial \ln L}{\partial \phi}: \quad \frac{n}{\phi} - \sum_{i=1}^{n}[N-(i-1)]r_i = 0$$

$$\phi = \frac{n}{\sum_{i=1}^{n}[N-(i-1)]r_i}$$

$$\phi = \frac{n}{N(\sum_{i=1}^{n} r_i) - \sum_{i=1}^{n}(i-1)r_i}$$

ดังนั้นสามารถหาค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมด (ϕ) ได้จาก

$$\phi = \frac{n}{N(\sum_{i=1}^{n} r_i) - \sum_{i=1}^{n}(i-1)r_i}$$

เมื่อ ϕ คือค่าคงที่ของผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไข

N คือจำนวนเริ่มต้นของข้อผิดพลาดทั้งหมดในซอฟต์แวร์

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3,\ldots,n$

n คือจำนวนข้อผิดพลาด ณ ตำแหน่งที่ต้องการประมาณค่า เมื่อ $n \leq N$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ค
การแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดของแบบจำลองที่ 3

การแก้สมการเพื่อหาผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมดและทำไปใช้แทนค่าในแบบจำลองที่ 3 ในงานวิจัยนี้ได้สร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยสร้างแบบจำลองด้วยวิธีที่แตกต่างกัน ซึ่งในการสร้างแบบจำลองที่ 3 นั้นเป็นการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ ในการสร้างแบบจำลองนี้จะต้องทำการแก้สมการเพื่อให้สอดคล้องกับงานวิจัย ซึ่งผลกระทบของแต่ละข้อผิดพลาดที่มีผลต่ออัตราการแก้ไขข้อผิดพลาดหรือ ϕ ในแบบจำลองที่ 3 นั้นยังไม่ทราบค่า จึงต้องทำการแก้สมการเพื่อใช้ในการหาค่า ϕ แล้วจากนั้นจะสามารถประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดได้ ซึ่งสามารถอธิบายรายละเอียดในการแก้สมการเพื่อหาค่า ϕ ได้ดังนี้

ในแบบจำลองที่ 3 เป็นสร้างแบบจำลองการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด โดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์ โดยที่อัตราการแก้ไขข้อผิดพลาดที่ i (λ_i) สามารถหาได้จาก

$$\lambda_i = [N - (i-1)(Instability)]\phi$$

และในการแจกแจงแบบเอกซ์โปแนนเชียลนั้นค่า $f(r_i) = \lambda e^{-\lambda r_i}$ ดังนั้นจะได้

$$f(r_i) = \phi[N - (i-1)(Instability)]\, exp(-\phi[N - (i-1)(Instability)]r_i)$$

จากนั้นใช้วิธี maximum-likelihood estimation (MLE) ในการประมาณ ϕ และ N โดยในแบบจำลองที่ 3 นั้นทราบค่าของจำนวนข้อผิดพลาดทั้งหมด (N) ดังนั้นจะสามารถหาค่า ϕ ได้โดยแสดงรายละเอียดดังนี้

$$L(N,\phi) = \prod_{i=1}^{n} \phi\,[N - (i-1)(Instability)] \exp(-\phi[N - (i-1)(Instability)]r_i)$$

$$= \phi^n \prod_{i=1}^{n} [N - (i-1)(Instability)] \exp(-\phi \sum_{i=1}^{n} [N - (i-1)(Instability)]r_i)$$

$$= \ln\left[\phi^n \prod_{i=1}^{n} [N - (i-1)(Instability)] \exp\left(-\phi \sum_{i=1}^{n} [N - (i-1)(Instability)]r_i\right)\right]$$

$$= n \ln \phi + \sum_{i=1}^{n} \ln[N - (i-1)(Instability)] - \phi \sum_{i=1}^{n} [N - (i-1)(Instability)]r_i$$

$$\frac{\partial \ln L}{\partial \phi}: \quad \frac{n}{\phi} - \sum_{i=1}^{n}[N-(i-1)(Instability)]r_i = 0$$

$$\phi = \frac{n}{\sum_{i=1}^{n}[N-(i-1)(Instability)]r_i}$$

$$\phi = \frac{n}{N(\sum_{i=1}^{n} r_i) - \sum_{i=1}^{n}(i-1)(Instability)r_i}$$

ดังนั้นสามารถหาค่าผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไขข้อผิดพลาดทั้งหมด (ϕ) ได้จาก

$$\phi = \frac{n}{N(\sum_{i=1}^{n} r_i) - \sum_{i=1}^{n}(i-1)(Instability)r_i}$$

เมื่อ ϕ คือค่าคงที่ของผลกระทบของแต่ละข้อผิดพลาดที่มีต่ออัตราการแก้ไข

N คือจำนวนเริ่มต้นของข้อผิดพลาดทั้งหมดในซอฟต์แวร์

r_i คือค่าระยะเวลาจริงที่ใช้ในการแก้ไขข้อผิดพลาดที่ i เมื่อ $i = 1,2,3,\ldots,n$

n คือจำนวนข้อผิดพลาด ณ ตำแหน่งที่ต้องการประมาณค่า เมื่อ $n \leq N$

$Instability$ คือค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ง
การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบที

งานวิจัยนี้ได้ทำการประเมินผลแบบจำลองโดยใช้ชุดโปรแกรมสำหรับทดสอบแบบจำลองทั้งหมด 10 โปรแกรม ซึ่งในการเปรียบเทียบแบบจำลองนั้นจะใช้การทดสอบนัยสำคัญด้วยสถิติทดสอบที โดยใช้ Paired t-test ที่ระดับความเชื่อมั่น 95% เพื่อเปรียบเทียบผลจากแบบจำลองว่าแบบจำลองใดมีประสิทธิภาพมากกว่ากัน ในการทดสอบนัยสำคัญด้วยสถิติทดสอบทีนั้น จะทำการเปรียบแบบจำลองเป็นคู่ๆ โดยผลการทดสอบนัยสำคัญด้วยสถิติทดสอบทีสามารถแสดงรายละเอียดได้ดังนี้

1) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 1

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-244.818	451.44860466	81.08256	-410.411	-79.2258	-3.019	30	.005
Pair 2	Model 1 - Model 3	-42.9117	45.87787442	8.239909	-59.7398	-26.0836	-5.208	30	.000
Pair 3	Model 2 - Model 3	189.7120	421.67557578	73.40430	40.19239	339.2317	2.584	32	.015

รูปที่ 35 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 1

จากรูปที่ 35 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 1 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.019
  - ค่า Mean มีค่าเป็น -244.818
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.005
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.005}{2} = 0.0025$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -5.208
  - ค่า Mean มีค่าเป็น -42.9117
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.000
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.000}{2} = 0.000$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.584
  - ค่า Mean มีค่าเป็น 189.7120
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.015
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.015}{2} = 0.0075$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 2

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-9.13455	18.84086026	2.777933	-14.7296	-3.53950	-3.288	45	.002
Pair 2	Model 1 - Model 3	-1.65284	2.49815713	.36833317	-2.39470	-.910974	-4.487	45	.000
Pair 3	Model 2 - Model 3	7.170162	17.65605624	2.548432	2.043378	12.29695	2.814	47	.007

รูปที่ 36 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 2

จากรูปที่ 36 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 2 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.288
  - ค่า Mean มีค่าเป็น -9.13455
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.002
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.002}{2} = 0.001$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -4.487
  - ค่า Mean มีค่าเป็น -1.65284
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.000
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.000}{2} = 0.000$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.814
  - ค่า Mean มีค่าเป็น 7.170162
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.007
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.007}{2} = 0.0035$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

3) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 3

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-3.91519	6.35695874	1.421459	-6.89034	-.940039	-2.754	19	.013
Pair 2	Model 1 - Model 3	-.645374	1.28915048	.28826281	-1.24872	-.042033	-2.239	19	.037
Pair 3	Model 2 - Model 3	2.972969	6.11197667	1.303078	.26307073	5.682868	2.281	21	.033

รูปที่ 37 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 3

จากรูปที่ 37 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 3 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.754
  - ค่า Mean มีค่าเป็น -3.91519
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.013
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.013}{2} = 0.0065$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.239
  - ค่า Mean มีค่าเป็น -0.645374
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.037
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.037}{2} = 0.0185$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

Chulalongkorn University

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.281
  - ค่า Mean มีค่าเป็น 2.972969
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.033
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.033}{2} = 0.0165$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

4) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 4

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-91.9301	134.18481360	38.73582	-177.187	-6.67318	-2.373	11	.037
Pair 2	Model 1 - Model 3	-91.9301	134.18481360	38.73582	-177.187	-6.67318	-2.373	11	.037

รูปที่ 38 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 4

จากรูปที่ 38 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 4 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.373
  - ค่า Mean มีค่าเป็น -91.9301
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.037
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.037}{2} = 0.0185$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.373
  - ค่า Mean มีค่าเป็น -91.9301
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.037
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.037}{2} = 0.0185$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่าค่าประมาณที่ได้จากแบบจำลองที่ 2 และแบบจำลองที่ 3 นั้นมีค่าเท่ากัน จึงไม่สามารถหาความแตกต่างของทั้งสองแบบจำลองนี้ได้ จึงไม่ปรากฏผลลัพธ์ของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 ในรูปที่ 38

5) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 5

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-87.9276	156.04985298	26.37724	-141.533	-34.3226	-3.333	34	.002
Pair 2	Model 1 - Model 3	-58.8879	67.96301522	11.48785	-82.2340	-35.5418	-5.126	34	.000
Pair 3	Model 2 - Model 3	27.47232	90.57651852	14.89069	-2.72740	57.67203	1.845	36	.073

รูปที่ 39 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 5

จากรูปที่ 39 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 5 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.333
  - ค่า Mean มีค่าเป็น -87.9276
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.002
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.002}{2} = 0.001$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -5.126
  - ค่า Mean มีค่าเป็น -58.8879
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.000
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.000}{2} = 0.000$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 1.845
  - ค่า Mean มีค่าเป็น 27.47232
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.073
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.073}{2} = 0.0365$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

6) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 6

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-2.64797	6.49108238	.83109793	-4.31041	-.985523	-3.186	60	.002
Pair 2	Model 1 - Model 3	-.337152	.75063123	.09610848	-.529398	-.144906	-3.508	60	.001
Pair 3	Model 2 - Model 3	2.237509	6.10870463	.76962444	.69905185	3.775967	2.907	62	.005

รูปที่ 40 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 6

จากรูปที่ 40 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 6 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.186
  - ค่า Mean มีค่าเป็น -2.64797
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.002
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.002}{2} = 0.001$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.508
  - ค่า Mean มีค่าเป็น -0.337152
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.001
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.001}{2} = 0.0005$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.907
  - ค่า Mean มีค่าเป็น 2.237509
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.005
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.005}{2} = 0.0025$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

7) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-222.164	375.32396662	70.92956	-367.699	-76.6285	-3.132	27	.004
Pair 2	Model 1 - Model 3	-11.9922	23.22894806	4.389859	-20.9994	-2.98493	-2.732	27	.011
Pair 3	Model 2 - Model 3	196.2282	369.51826710	67.46450	58.24776	334.2085	2.909	29	.007

รูปที่ 41 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7

จากรูปที่ 41 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 7 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.132
  - ค่า Mean มีค่าเป็น -222.164
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.004
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.004}{2} = 0.002$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.732
  - ค่า Mean มีค่าเป็น -11.9922
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.011
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.011}{2} = 0.0055$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.909
  - ค่า Mean มีค่าเป็น 196.2282
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.007
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.007}{2} = 0.0035$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

8) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 8

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-79.2290	105.39618861	27.21318	-137.596	-20.8626	-2.911	14	.011
Pair 2	Model 1 - Model 3	-22.9213	26.05807726	6.728167	-37.3518	-8.49082	-3.407	14	.004
Pair 3	Model 2 - Model 3	49.70008	91.54086105	22.20192	2.634109	96.76604	2.239	16	.040

รูปที่ 42 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 8

จากรูปที่ 42 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 8 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.911
  - ค่า Mean มีค่าเป็น -79.2290
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.011
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.011}{2} = 0.0055$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.407
  - ค่า Mean มีค่าเป็น -22.9213
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.004
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.004}{2} = 0.002$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.239
  - ค่า Mean มีค่าเป็น 49.70008
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.040
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.040}{2} = 0.020$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

9) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 9

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-23.8521	62.07258572	7.265047	-38.3347	-9.36953	-3.283	72	.002
Pair 2	Model 1 - Model 3	-3.28004	8.97363448	1.050284	-5.37374	-1.18634	-3.123	72	.003
Pair 3	Model 2 - Model 3	20.02426	59.39878339	6.858781	6.357844	33.69068	2.920	74	.005

รูปที่ 43 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 9

จากรูปที่ 43 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 9 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.283
  - ค่า Mean มีค่าเป็น -23.8521
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.002
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.002}{2} = 0.001$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -3.123
  - ค่า Mean มีค่าเป็น -3.28004
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.003
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.003}{2} = 0.0015$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 2.920
  - ค่า Mean มีค่าเป็น 20.02426
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.005
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.005}{2} = 0.0025$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

10) การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 10

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Model 1 - Model 2	-23.8199	27.25318106	10.30073	-49.0249	1.385062	-2.312	6	.060
Pair 2	Model 1 - Model 3	-5.75967	7.68937420	2.906310	-12.8712	1.351815	-1.982	6	.095
Pair 3	Model 2 - Model 3	14.09784	21.59888348	7.199628	-2.50453	30.70021	1.958	8	.086

รูปที่ 44 ผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 10

จากรูปที่ 44 แสดงผลลัพธ์การเปรียบเทียบแบบจำลองด้วยสถิติทดสอบทีของโปรแกรมที่ 10 ซึ่งใช้วิธีของ Paired t-test เมื่อทำการทดสอบเสร็จเรียบร้อยแล้วสามารถอธิบายรายละเอียดได้ดังนี้

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 2 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -2.312
  - ค่า Mean มีค่าเป็น -23.8199
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.060
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.060}{2} = 0.030$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 1 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น -1.982
  - ค่า Mean มีค่าเป็น -5.75967
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.095
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.095}{2} = 0.0475$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง แต่ว่าค่า t มีค่าเป็นลบและค่า Mean มีค่าเป็นลบ ซึ่งแสดงให้เห็นว่าค่า μ_1 มีค่าน้อยกว่า μ_3 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 1 มีประสิทธิภาพกว่าแบบจำลองที่ 3 ที่ระดับนัยสำคัญ 0.05

- ผลของการเปรียบเทียบแบบจำลองที่ 2 กับแบบจำลองที่ 3 เมื่อทำการทดสอบทีเสร็จพบว่า
  - ค่า t มีค่าเป็น 1.958
  - ค่า Mean มีค่าเป็น 14.09784
  - ค่า Sig.(2-tailed) มีค่าเป็น 0.086
  - ดังนั้น $\frac{Sig.(2-tailed)}{2} = \frac{0.086}{2} = 0.043$ ซึ่งมีค่าน้อยกว่า 0.05

ดังนั้นจึงสรุปได้ว่า H_1 เป็นจริง ซึ่งแสดงให้เห็นว่าค่า μ_3 มีค่าน้อยกว่า μ_2 นั่นคือค่าประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่ได้จากแบบจำลองที่ 3 มีประสิทธิภาพกว่าแบบจำลองที่ 2 ที่ระดับนัยสำคัญ 0.05

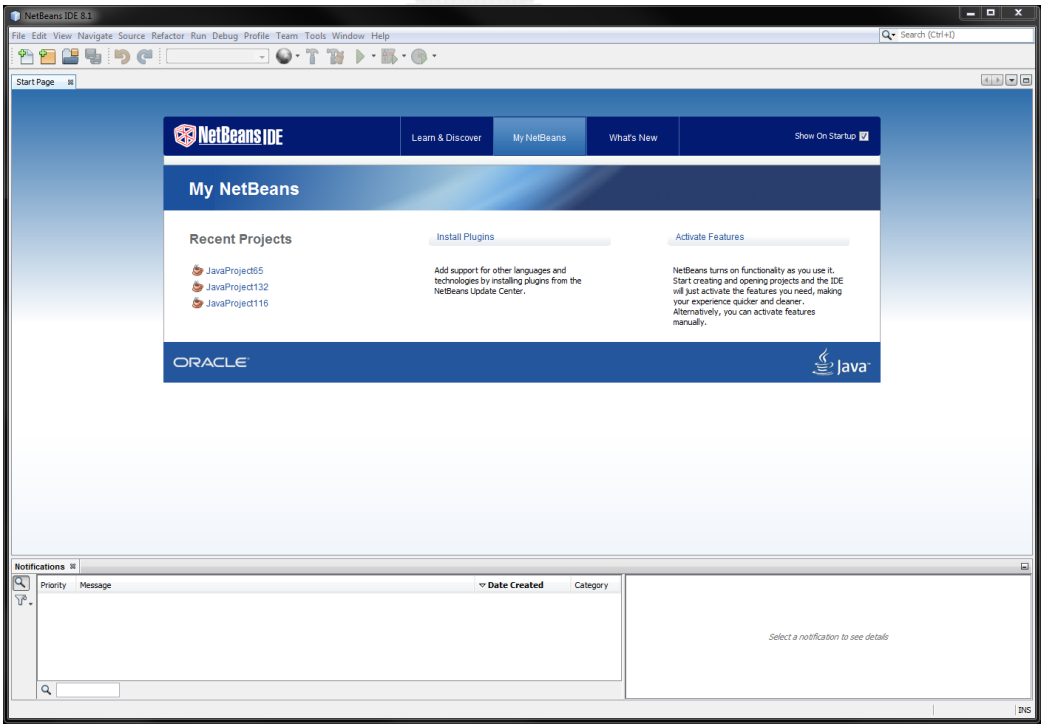
เมื่อทำการเปรียบเทียบแบบจำลองทั้งสามแบบจำลองด้วยสถิติทดสอบทีกับ 10 โปรแกรม พบว่า แบบจำลองที่ 1 ที่สร้างด้วยการระบุค่าเฉลี่ยของฟังก์ชันความน่าจะเป็นที่มีการแจกแจงข้อมูลแบบเอกซ์โปแนนเชียลมีประสิทธิภาพมากที่สุด ในขณะที่แบบจำลองที่ 3 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล และมาตรวัดความไม่เสถียรของซอฟต์แวร์นั้นมีประสิทธิภาพเป็นอันดับที่สอง และแบบจำลองที่ 2 ที่สร้างโดยการประยุกต์ใช้ทฤษฎีการแจกแจงแบบเอกซ์โปแนนเชียล มีประสิทธิภาพน้อยที่สุด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

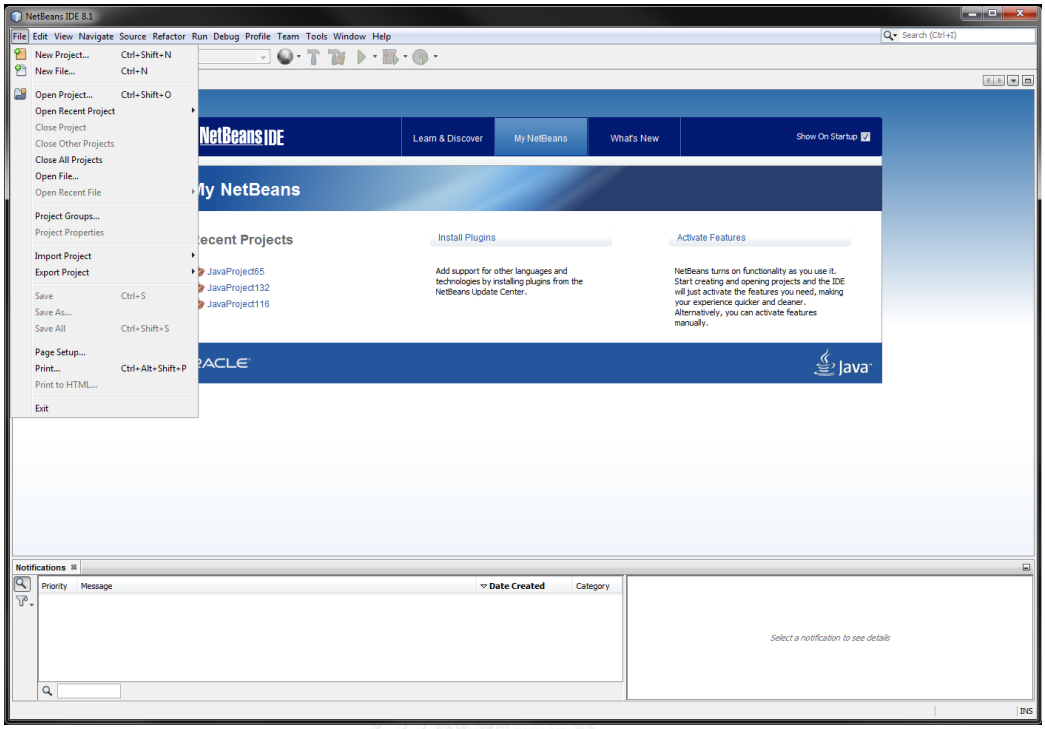
ภาคผนวก จ
การคำนวณค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด

การคำนวณค่าคุณลักษณะที่มีผลกระทบกับระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาด ซึ่งในงานวิจัยนี้เลือกใช้มาตรวัดความไม่เสถียรของซอฟต์แวร์ (Software Instability Metric) ซึ่งมาตรวัดนี้เป็นตัวบ่งชี้ถึงความต้านทานการได้รับผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงของคลาสต่างๆ ในซอฟต์แวร์ ซึ่งคำนวณได้จากรหัสต้นฉบับ เพื่อเป็นศึกษาว่าความเสถียรของซอฟต์แวร์นั้นมีผลกระทบให้ระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดนั้นเพิ่มขึ้นหรือน้อยลง ในการคำนวณค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์นั้นใช้ปลั๊กอินของโปรแกรมเน็ตบีนส์ (NetBeans) ชื่อว่าซอร์ซโค้ดเมตริกซ์ (Source Code Metrics) โดยนำเข้าไฟล์รหัสต้นฉบับที่อยู่ในรูปแบบไฟล์จาวา (Java) ซึ่งงานวิจัยนี้ได้รวบรวมรหัสต้นฉบับไว้ในขั้นตอนของการเตรียมชุดโปรแกรมเรียบร้อยแล้ว เมื่อลงปลั๊กอินในโปรแกรมเน็ตบีนส์เรียบร้อยแล้ว โปรแกรมจะต้องทำการนำเข้าไฟล์รหัสต้นฉบับของโปรแกรมที่ต้องการหาค่ามาตรวัด ซึ่งมีขั้นตอนและรายละเอียดดังนี้

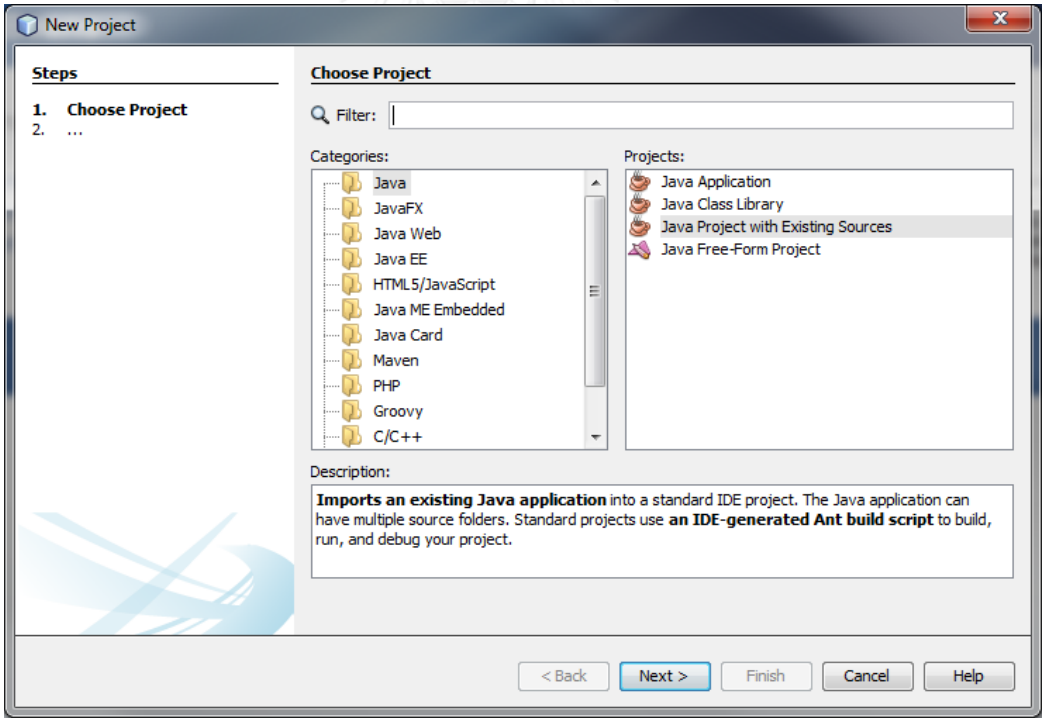
เมื่อเปิดโปรแกรมเน็ตบีนส์ขึ้นมาจะแสดงดังรูปที่ 45 จากนั้นเลือกที่ File และเลือกที่ New Project ดังรูปที่ 46 จากนั้นโปรแกรมจะแสดงดังรูปที่ 47 เพื่อทำการนำเข้าไฟล์รหัสต้นฉบับของโปรแกรมที่ต้องการหาค่ามาตรวัด



รูปที่ 45 หน้าต่างหลักของโปรแกรมเน็ตบีนส์



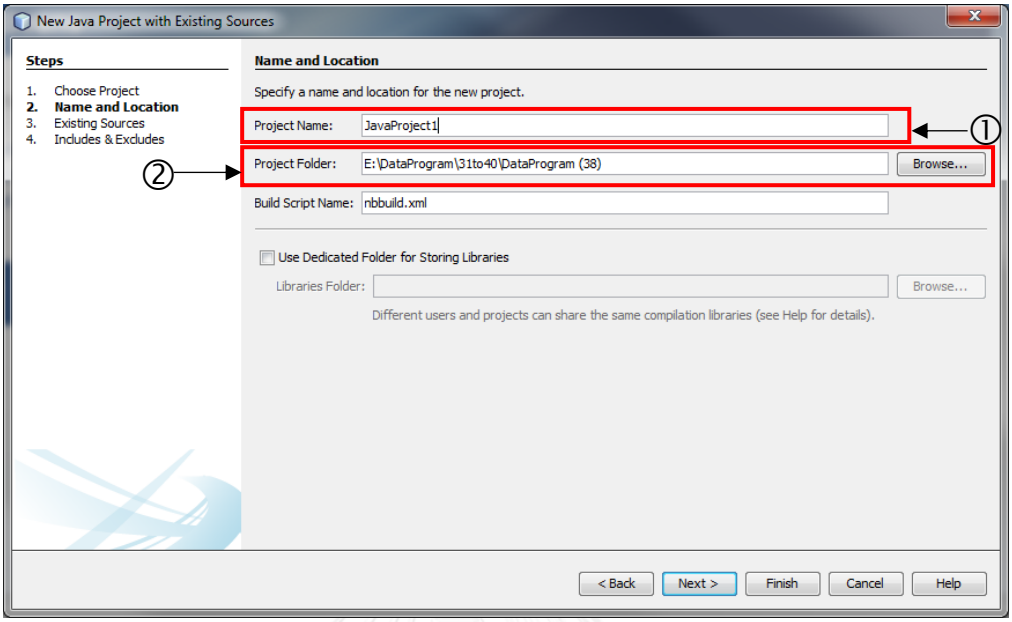
รูปที่ 46 หน้าต่างของการนำเข้าไฟล์รหัสต้นฉบับ



รูปที่ 47 หน้าต่างของการเลือกนำเข้าไฟล์รหัสต้นฉบับจากที่ได้รวบรวมไว้

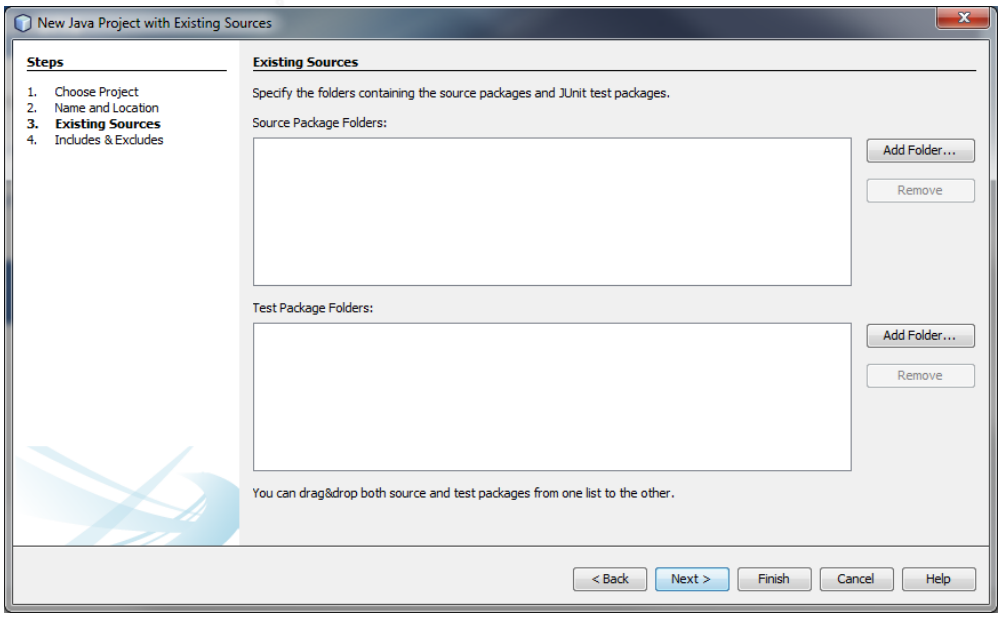
จากรูปที่ 47 ผู้ใช้งานจะต้องเลือก Java Project with Existing Sources และกด Next เพื่อนำเข้าไฟล์รหัสต้นฉบับที่ได้รวบรวมไว้ จากนั้นตั้งชื่อโครงการในช่อง Project Name (หมายเลข 1)

หลังจากนั้นกดปุ่ม Browse แล้วเลือกที่อยู่ของรหัสต้นฉบับของโปรแกรม (หมายเลข 2) และกดปุ่ม Next เพื่อเข้าสู่ขั้นตอนถัดไป ซึ่งแสดงดังรูปที่ 48

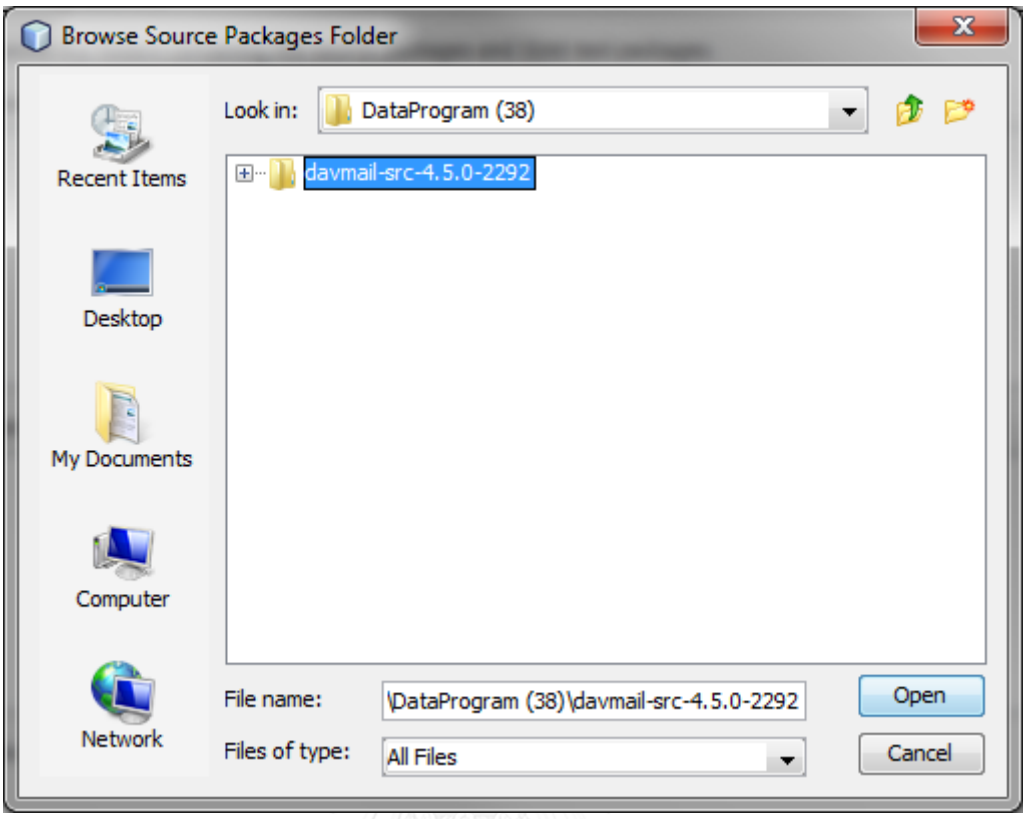


รูปที่ 48 หน้าต่างแสดงการเลือกที่อยู่ไฟล์รหัสต้นฉบับ

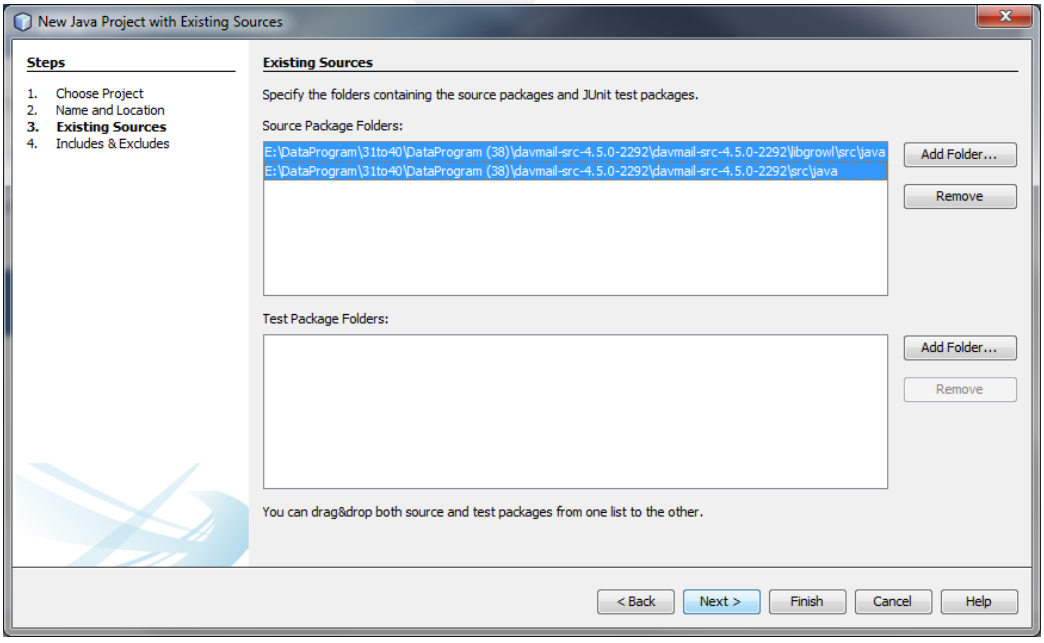
จากนั้นเมื่อกดปุ่ม Next แล้วจะแสดงดังรูปที่ 49 จากนั้นกดปุ่ม Add Folder แล้วเลือกโฟลเดอร์ของรหัสต้นฉบับดังรูปที่ 50 แล้วกดปุ่ม Open จะแสดงดังรูปที่ 51 และกดปุ่ม Next จะแสดงดังรูปที่ 52 จากนั้นกดปุ่ม Finish เพื่อเสร็จขั้นตอนของการนำเข้าไฟล์รหัสต้นฉบับ โดยเมื่อนำเข้าไฟล์สำเร็จจะแสดงดังรูปที่ 53



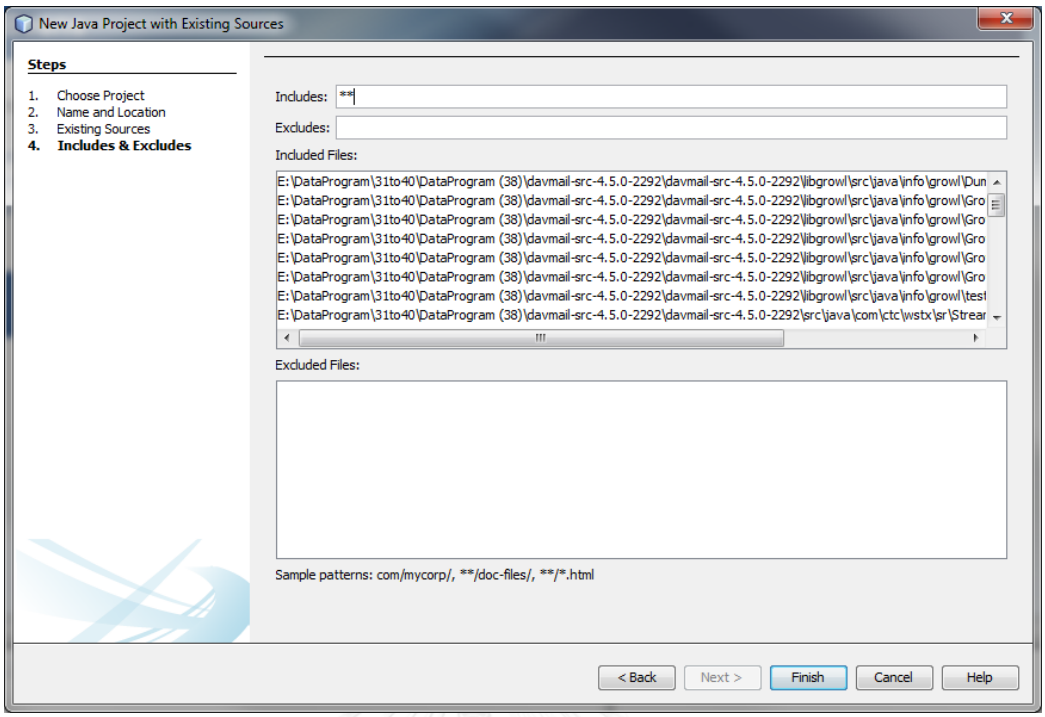
รูปที่ 49 หน้าต่างการเลือกโฟลเดอร์ของรหัสต้นฉบับ



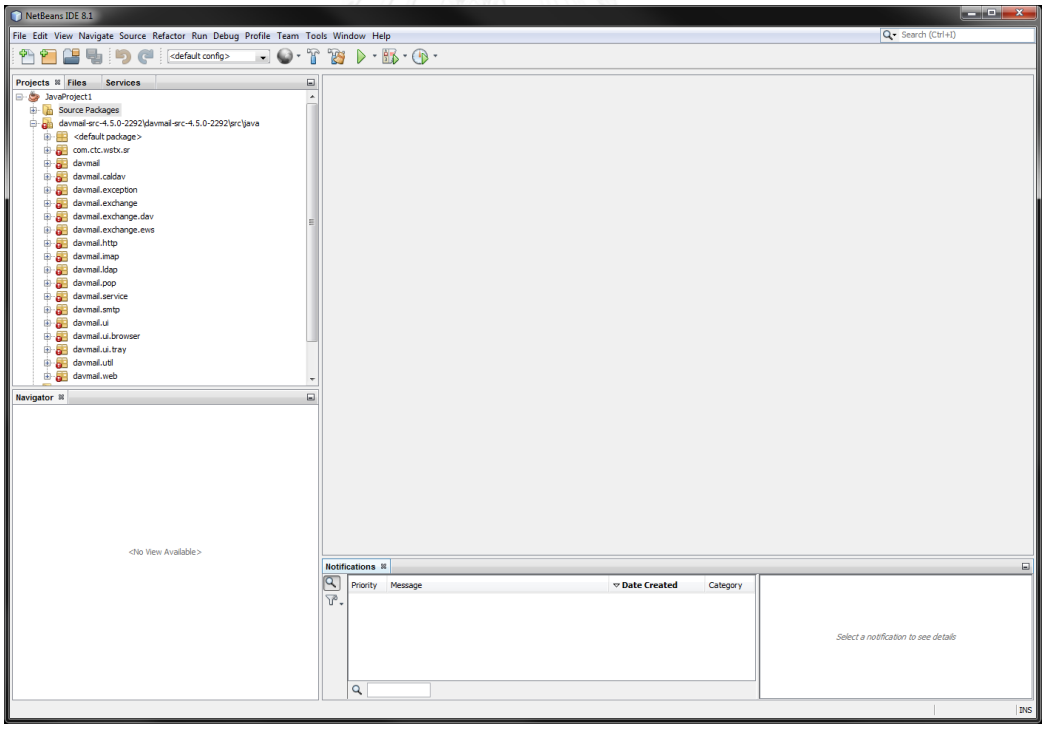
รูปที่ 50 หน้าต่างเลือกที่อยู่ของโฟลเดอร์รหัสต้นฉบับ



รูปที่ 51 หน้าต่างแสดงรายการที่อยู่ของโฟลเดอร์รหัสต้นฉบับ

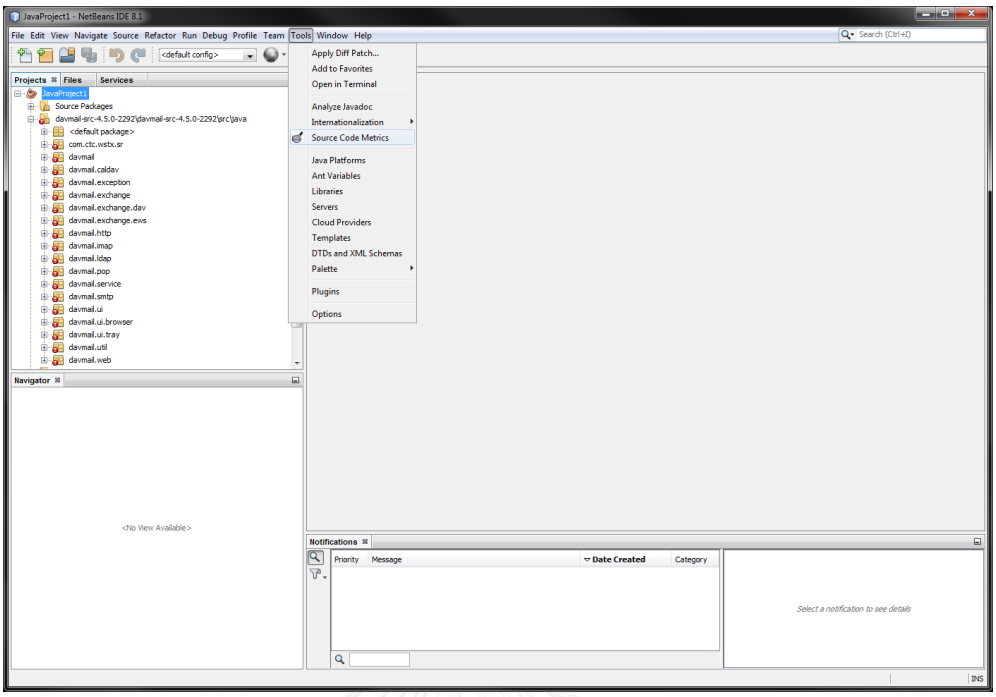


รูปที่ 52 หน้าต่างแสดงรายการของไฟล์รหัสต้นฉบับ



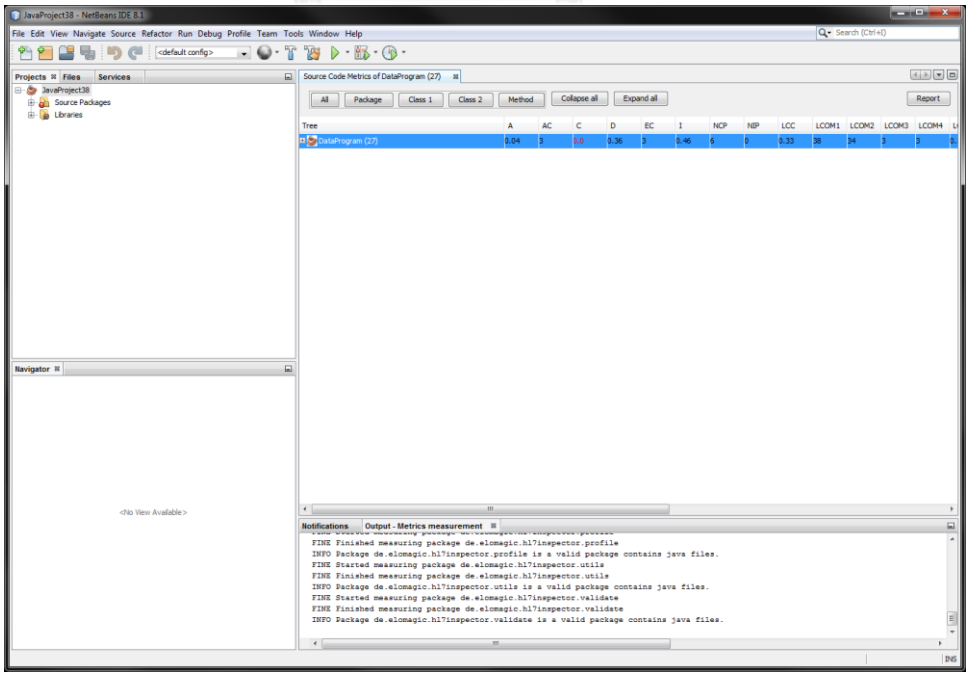
รูปที่ 53 หน้าต่างแสดงการนำเข้าไฟล์รหัสต้นฉบับสำเร็จ

เมื่อนำเข้าไฟล์รหัสต้นฉบับเรียบร้อยแล้ว จากนั้นจะคำนวณค่ามาตรวัดโดยเลือกที่ Tools จากนั้นเลือกที่ Source Code Metrics ดังรูปที่ 54

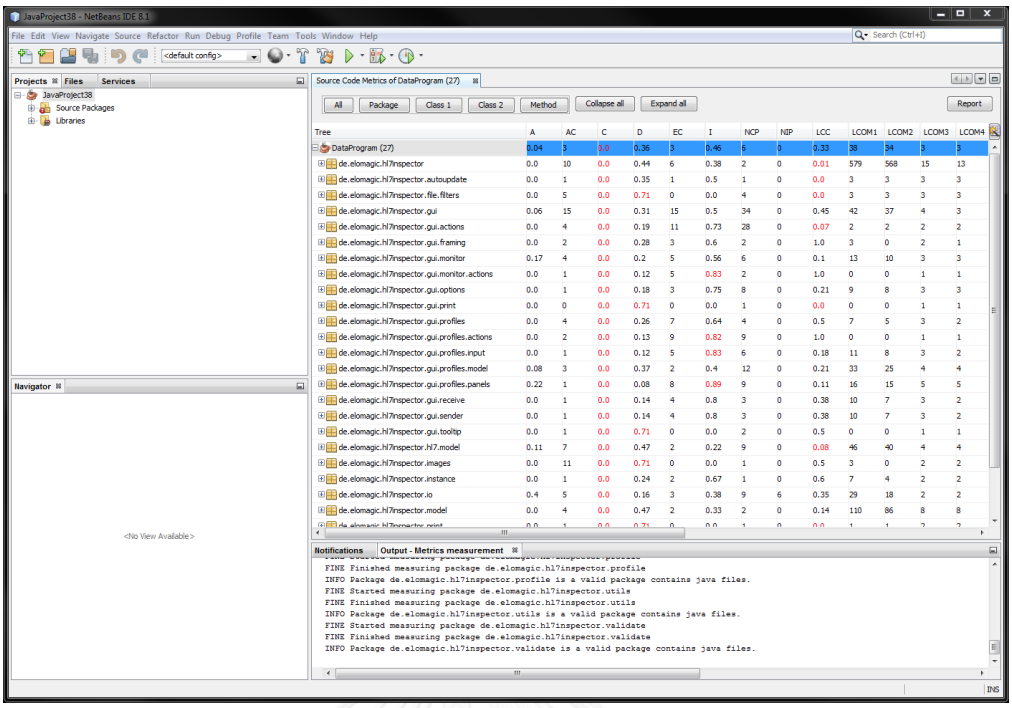


รูปที่ 54 หน้าต่างแสดงการเลือกคำสั่งการคำนวณค่ามาตรวัด

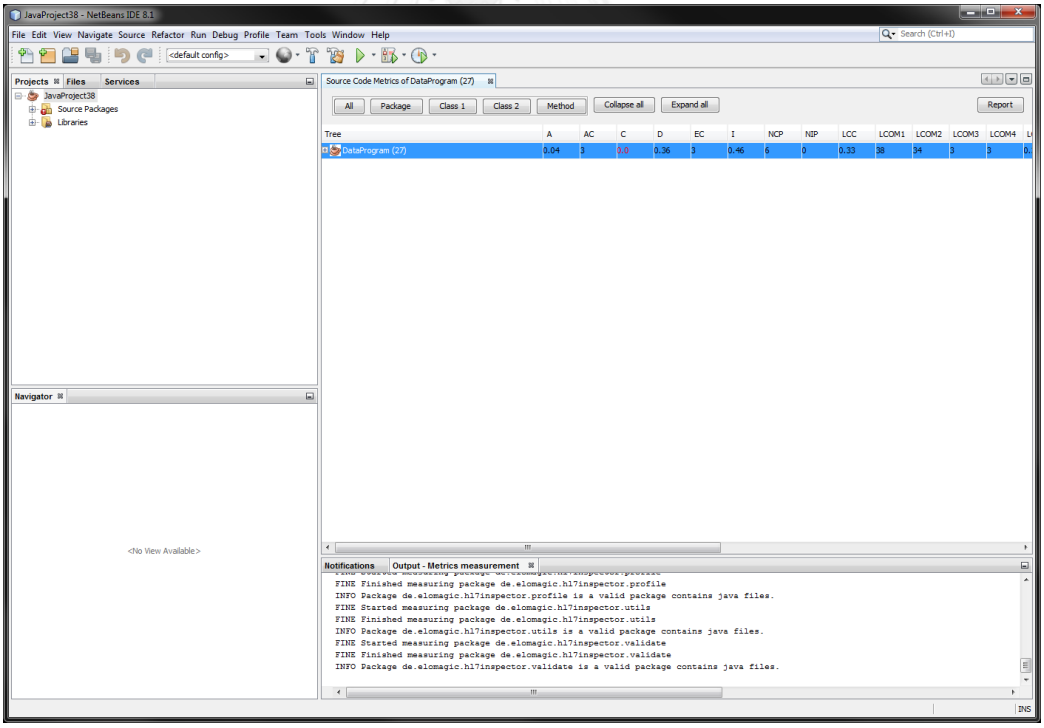
จากนั้นโดยโปรแกรมคำนวณหาค่ามาตรวัดของรหัสต้นฉบับดังรูปที่ 55 เมื่อคำนวณเสร็จเรียบร้อยจะแสดงดังรูปที่ 56 โดยจะประกอบด้วยค่ามาตรวัดต่างๆ และแสดงค่ามาตรวัดในระดับคลาส ระดับแพ็คเกจ และระดับโปรแกรม ซึ่งในงานวิจัยนี้จะเลือกค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์ในระดับของโปรแกรมดังรูปที่ 57



รูปที่ 55 หน้าต่างแสดงการประมวลผลการคำนวณค่ามาตรวัด



รูปที่ 56 หน้าต่างแสดงค่ามาตรวัดต่างๆ



รูปที่ 57 หน้าต่างแสดงค่ามาตรวัดในระดับโปรแกรม

เมื่อได้ค่ามาตรวัดความไม่เสถียรของซอฟต์แวร์แล้ว ก็จะนำไปใช้ในการประมาณระยะเวลาที่ใช้ในการแก้ไขข้อผิดพลาดของแบบจำลองที่ 3 ได้

ประวัติผู้เขียนวิทยานิพนธ์

นายปวินท์ อยู่พวง เกิดเมื่อวันที่ 21 กรกฎาคม พ.ศ. 2534 ณ จังหวัดนครปฐม วุฒิการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน สำเร็จการศึกษาในปี พ.ศ. 2555 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2556

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University