การประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล

นางสาวอลิสา แสงภู่วงษ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านบัณฑิตวิทยาลัย
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.

ESTIMATING STABILITY OF SOFTWARE DESIGN IN INCREMENTAL DEVELOPMENT

Miss Alisa Sangpuwong

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

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 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล
โดย	นางสาวอลิสา แสงภู่วงษ์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี)

.....กรรมการ

(รองศาสตราจารย์ ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร. มฑุปายาส ทองมาก)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

อลิสา แสงภู่วงษ์ : การประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล. (ESTIMATING STABILITY OF SOFTWARE DESIGN IN INCREMENTAL DEVELOPMENT) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. พรศิริ หมื่นไชยศรี, 110 หน้า.

ความเสถียรของซอฟต์แวร์เป็นตัวชี้วัดหนึ่งที่บ่งบอกถึงคุณภาพของซอฟต์แวร์ ซึ่งเป็นความต้านทานต่อผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้น ความเสถียรสามารถวัดค่าได้จากซอร์ซโค้ดในขั้นตอนของการบำรุงรักษาและประมาณค่าได้จากโมเดลการออกแบบในขั้นตอนของการออกแบบ โดยโมเดลสำหรับการประมาณค่าความเสถียรนั้นมีหลายโมเดล ซึ่งสร้างขึ้นด้วยความแตกต่างของปัจจัย ตัววัด และวิธีการในการสร้างโมเดล การเปลี่ยนแปลงของโครงสร้างการออกแบบซอฟต์แวร์นั้นเป็นปัจจัยสำคัญที่ส่งผลกระทบต่อความเสถียรของซอฟต์แวร์และคุณสมบัติอื่นๆที่บ่งบอกถึงคุณภาพซอฟต์แวร์ วิทยานิพนธ์นี้มุ่งเน้นการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล เมื่ออินคริเมนต์ปัจจุบันถูกเพิ่มเข้าไปในส่วนของอินคริเมนต์ก่อนหน้านี้ โครงสร้างการออกแบบอาจจะได้รับผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้น ด้วยเหตุนี้ค่าความเสถียรเชิงตรรกะอาจจะเพิ่มขึ้นหรือลดลง หากค่าความเสถียรเชิงตรรกะมีค่าลดลงก็จะสามารถเป็นตัวชี้วัดที่จะนำไปสู่การแก้ไขการออกแบบ ดังนั้นวิทยานิพนธ์นี้จึงนำเสนอโมเดลในการประมาณค่าความเสถียรเชิงตรรกะของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัลจากแผนภาพคลาสและแผนภาพลำดับ ซึ่งโมเดลถูกสร้างด้วย 4 วิธีการที่แตกต่างกัน ได้แก่ การวิเคราะห์ความถดถอยเชิงซ้อน การวิเคราะห์ส่วนประกอบหลัก การวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ และการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด จากนั้นทำการประเมินผลและเปรียบเทียบความแม่นยำของโมเดล ซึ่งผลการทดลองพบว่าโมเดลทั้งหมดนั้นเป็นที่ยอมรับในการนำไปประมาณค่าความเสถียร และโมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลักคือโมเดลที่มีความแม่นยำดีที่สุด คือ PRED(0.25)=0.9763 โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อนมีความแม่นยำเป็นอันดับที่สอง คือ PRED(0.25)=0.9594 โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบและใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดมีความแม่นยำเป็นอันดับที่สามคือ PRED(0.25)=0.9512 และโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบที่ไม่ได้ทำการคัดเลือกตัววัดด้วยวิธีการวิเคราะห์ส่วนประกอบหลักมีความแม่นยำเป็นอันดับสุดท้ายคือ PRED(0.25)=0.8571

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

5470459721 : MAJOR SOFTWARE ENGINEERING
KEYWORDS: STABILITY ESTIMATION / CLASS LOGICAL STABILITY / SOFTWARE DESIGN STABILITY / INCREMENTAL DEVELOPMENT

ALISA SANGPUWONG: ESTIMATING STABILITY OF SOFTWARE DESIGN IN INCREMENTAL DEVELOPMENT. ADVISOR: ASSOC. PROF. PORNSIRI MUENCHAISRI, Ph.D., 110 pp.

Software stability is indicative of software quality which resists to the effects of changes. Stability is measured from code at the maintenance phase and estimated from design models at the design phase. There are several models for estimating stability that are developed from a variety of factors, indicators, and methods. Changes of software design structure are one of the significant factors affecting not only the software stability but also other quality attributes of software. This research focuses on estimating logical stability in incremental development. When the current increment is added to the previous increment, the new design structure may be affected by the changes. As a result, the logical stability may be reduced or increased. If logical stability is reduced, it is possibly an indicator to redesign. So this research proposes models to estimate the logical stability of software design in incremental development from class diagrams and sequence diagrams. The models are constructed by using 4 different methods: multiple regression analysis, principal component analysis, design logical ripple effect analysis and design logical ripple effect analysis which uses principal component analysis to select indicators. Then the accuracy of the models are validated and compared. The experiment result shows that the models are acceptable for estimating stability and the model that is constructed by principal component analysis is the best model in terms of model accuracy with PRED(0.25)=0.9763. The model that is constructed by multiple regression analysis is the second best model in terms of model accuracy with PRED(0.25)=0.9594. The model that is constructed by design logical ripple effect analysis with selected indicators by principal component analysis is the third best model in term of model accuracy with PRED(0.25)=0.9512 and the model that is constructed by design logical ripple effect analysis without selected indicators by principal component analysis comes last in terms of model accuracy with PRED(0.25)=0.8571.

Department: Computer Engineering Student's Signature .....
Field of Study: Software Engineering Advisor's Signature .....
Academic Year: 2013

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี ที่ได้เสียสละเวลาในการช่วยเหลือในด้านความรู้ทางวิชาการและให้กำลังใจสนับสนุนให้งานวิจัยมีความก้าวหน้า รวมทั้งให้แนวทางและคำปรึกษาที่มีประโยชน์ในการทำงานวิจัยที่ดีมากมาโดยตลอด ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณคณะกรรมการในการสอบวิทยานิพนธ์ทุกท่าน รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ รองศาสตราจารย์ ดร.ทวีตีย์ เสนีวงศ์ ณ อยุธยา และผู้ช่วยศาสตราจารย์ ดร.มฑุปายาส ทองมาก ที่กรุณาเสียสละเวลาในการให้คำแนะนำเกี่ยวกับงานวิจัย และตรวจสอบความถูกต้องครบถ้วนสมบูรณ์ของวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณคณาจารย์ทุกท่านและเพื่อนๆทุกคนในสาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ความรู้ คำแนะนำในการเรียน และการทำวิจัย

สุดท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา และสมาชิกทุกคนในครอบครัวที่โอกาสในการศึกษา รวมทั้งการสนับสนุนให้คำแนะนำและกำลังใจและการทำงานวิจัยแก่ผู้วิจัยมาโดยตลอดจนสำเร็จการศึกษา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญภาพ ..... ฐ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของงานวิจัย 3

1.3 ขอบเขตของงานวิจัย 3

1.4 ประโยชน์ของงานวิจัย 4

1.5 ขั้นตอนและวิธีดำเนินการวิจัย 4

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์ 5

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6

2.1 ทฤษฎีที่เกี่ยวข้อง 6

2.1.1 ความเสถียรของซอฟต์แวร์ (Software Stability) 6

2.1.1.1 ความเสถียรของซอฟต์แวร์เชิงตรรกะ (Logical Stability) 6

2.1.1.2 ความเสถียรของซอฟต์แวร์เชิงประสิทธิภาพ (Performance Stability) 6

2.1.2 ผลกระเพื่อมหรือผลกระทบที่เกิดจากการเปลี่ยนแปลง (Ripple effect) 7

2.1.3 การเปลี่ยนแปลง (Changes) 7

2.1.4 การคำนวณค่าความเสถียร 9

2.1.4.1 การคำนวณค่าความเสถียรในขั้นตอนของการบำรุงรักษา 9

2.1.4.2 การคำนวณค่าความเสถียรในขั้นตอนของการออกแบบ 11

2.1.5 การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล (Incremental software development) . 14

2.1.6 แผนภาพคลาสและแผนภาพลำดับ 14

2.1.8 การวิเคราะห์ส่วนประกอบหลัก (Principal Component Analysis) 18

2.2 งานวิจัยที่เกี่ยวข้อง 21

หน้า

2.2.1 งานวิจัยที่เกี่ยวข้องกับความเสถียรของซอฟต์แวร์ 21
2.2.1.1 งานวิจัยที่นำเสนอการคำนวณความเสถียรเชิงตรรกะจากซอร์ซโค้ด 21
2.2.1.2 งานวิจัยที่นำเสนอการวัดความเสถียรของคลาสเชิงวัตถุ 22
2.2.1.3 งานวิจัยที่นำเสนอโมเดลการคำนวณค่าความเสถียรเชิงตรรกะของการออกแบบโปรแกรมเชิงโครงสร้าง 22
2.2.1.4 กลุ่มงานวิจัยที่ทำนำเสนอการสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาส 22
2.2.2 งานวิจัยที่เกี่ยวข้องกับการประเมินค่าในการพัฒนาแบบอินคริเมนทัล 23
2.2.2.1 ความน่าเชื่อถือ (Reliability) 23
2.2.2.1 ความพยายาม (Effort) 23
2.2.3 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ 23
บทที่ 3 การสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาส ในการพัฒนาแบบอินคริเมนทัล 25
3.1 ขั้นตอนการเตรียมชุดโปรแกรม 27
3.2 ขั้นตอนการคำนวณค่าความเสถียรเชิงตรรกะของคลาส 28
3.3 ขั้นตอนการสร้างแผนภาพคลาสและแผนภาพลำดับ 30
3.4 ขั้นตอนการวัดค่าตัววัดแผนภาพ 30
3.5 ขั้นตอนการสร้างโมเดล 30
3.5.1 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) 33
3.5.2 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) 33
3.5.3 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) 34
3.5.4 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) 39
3.6 ขั้นตอนการประเมินผลและเปรียบเทียบโมเดล 40
บทที่ 4 การสร้างและการประเมินผลโมเดลประมาณค่าความเสถียรเชิงตรรกะ 42

หน้า

4.1 การสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะ 44
4.1.1 การสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) 44
4.1.2 การสร้างโมเดลด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) 49
4.1.3 การสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) 53
4.1.4 การสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) 55
4.2 วัตถุประสงค์ของการประเมินผล 58
4.3 การประเมินโมเดลการประมาณค่าความเสถียร 58
4.3.1 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน 58
4.3.2 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ส่วนประกอบหลัก 59
4.3.3 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ 60
4.3.4 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด 61
4.4 การเปรียบเทียบการประเมินผลโมเดลการประมาณค่าความเสถียร 62
บทที่ 5 การออกแบบและพัฒนาเครื่องมือ สำหรับการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล 69
5.1 ความต้องการที่เป็นหน้าที่หลัก 69
5.1.1 การนำเข้าไฟล์ 69
5.1.2 การเตรียมข้อมูล 69
5.1.3 การแสดงผล 70
5.2 การออกแบบเครื่องมือ 70
5.2.1 แผนภาพยูสเคส 70
5.2.2 แผนภาพคลาส 74
5.2.3 แผนภาพลำดับ 77
5.2.4 แผนภาพกิจกรรม 79

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

5.2.5 เครื่องมือสนับสนุนที่ใช้ 81
5.3 การใช้งานเครื่องมือ 82
บทที่ 6 สรุปผลการวิจัย 84
6.1 สรุปผลการวิจัย 84
6.2 ข้อจำกัด 87
6.3 งานวิจัยในอนาคต 87
6.4 ผลงานตีพิมพ์จากวิทยานิพนธ์ 87
รายการอ้างอิง 88
ภาคผนวก 90
ภาคผนวก ก การคำนวณความเสถียรเชิงตรรกะของคลาสโดยโปรแกรม 91
ภาคผนวก ข การแปลงซอร์ซโค้ดเป็นแผนภาพคลาสและแผนภาพลำดับ 95
ภาคผนวก ค การวัดค่าตัววัดแผนภาพ 102
ภาคผนวก ง การใช้งานเครื่องมือประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล 105
ประวัติผู้เขียนวิทยานิพนธ์ 110

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 1 ประเภทของการเปลี่ยนแปลง 8
ตารางที่ 2 การออกแบบข้อมูลเข้าและข้อมูลออกของแต่ละมอดูล 12
ตารางที่ 3 สัญลักษณ์ความสัมพันธ์แบบต่างๆของแผนภาพคลาส 15
ตารางที่ 4 ตัววัดแผนภาพ 17
ตารางที่ 5 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลัก 19
ตารางที่ 6 ความหมายของผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลัก 19
ตารางที่ 7 เปรียบเทียบผลเมื่อไม่มีการหมุนแกนปัจจัยและเมื่อมีการหมุนแกนปัจจัยโดยวิธี Varimax 20
ตารางที่ 8 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ 24
ตารางที่ 9 ตัวอย่างซอร์ซโค้ดที่ใช้คำนวณความเสถียรเชิงตรรกะของคลาสต่อการเปลี่ยนแปลงที่กำหนด 29
ตารางที่ 10 ตัวอย่างการคำนวณค่าความเสถียรเชิงตรรกะของคลาสต่อการเปลี่ยนแปลงที่กำหนด . 29
ตารางที่ 11 การเปลี่ยนแปลงและการแทนค่าตัววัด จำแนกตามองค์ประกอบของแผนภาพ 37
ตารางที่ 12 ตัวอย่างการแทนค่าตัววัดของคลาส C1 38
ตารางที่ 13 ชุดโปรแกรมสำหรับสร้างและประเมินผลโมเดล 42
ตารางที่ 14 จำนวนคลาสในแต่ละเวอร์ชันของชุดโปรแกรม 44
ตารางที่ 15 ตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดแผนภาพที่ใช้ในการสร้างโมเดล (โปรแกรม Edastext) 45
ตารางที่ 16 สรุปข้อมูลทางสถิติสำหรับการวิเคราะห์ความถดถอยเชิงซ้อน 46
ตารางที่ 17 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน 46
ตารางที่ 18 ตัวอย่างข้อมูลซ้ำจากการรวบรวมข้อมูลชุดโปรแกรมสำหรับสร้างโมเดล 48
ตารางที่ 19 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลักของตัววัด 27 ตัว 50
ตารางที่ 20 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก 51
ตารางที่ 21 ตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดที่ใช้ในการปรับโมเดล 54
ตารางที่ 22 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลักของตัววัด 8 ตัววัด 56
ตารางที่ 23 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ 56

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

ตารางที่ 24 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน 59
ตารางที่ 25 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ส่วนประกอบหลัก 60
ตารางที่ 26 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ 61
ตารางที่ 27 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบโดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด .. 62
ตารางที่ 28 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียร 63
ตารางที่ 29 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียร 4 โมเดล ด้วยชุดโปรแกรมของแต่ละเวอร์ชัน 64
ตารางที่ 30 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียรในแต่ละเวอร์ชัน 64
ตารางที่ 31 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียรที่ระดับ PRED(0.25) แบ่งตามวิธีการสร้างในแต่ละเวอร์ชัน 65
ตารางที่ 32 เปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาสของโมเดลทั้ง 4 โมเดล 67
ตารางที่ 33 ขั้นตอนการทำงานของยูสเคส Import Diagram Metrics Value Files 72
ตารางที่ 34 ขั้นตอนการทำงานของยูสเคส Import Diagram Files 72
ตารางที่ 35 ขั้นตอนการทำงานของยูสเคส Estimate Class Logical Stability Using 3 Methods 73
ตารางที่ 36 ขั้นตอนการทำงานของยูสเคส Select Models for Estimating Class Logical Stability 74
ตารางที่ 37 ขั้นตอนการทำงานของยูสเคส Export Files 74

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญภาพ

หน้า

รูปที่ 1 การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล 1
รูปที่ 2 ตัวอย่างการออกแบบแผนผังโครงสร้างของมอดูล 12
รูปที่ 3 การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล 14
รูปที่ 4 ตัวอย่างแผนภาพคลาส 15
รูปที่ 5 ตัวอย่างแผนภาพลำดับ 16
รูปที่ 6 แนวคิดในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล 25
รูปที่ 7 ภาพรวมขั้นตอนของงานวิจัย 26
รูปที่ 8 ขั้นตอนการคำนวณค่าความเสถียรเชิงตรรกะของคลาส 28
รูปที่ 9 การสร้างแผนภาพคลาสและแผนภาพลำดับ 30
รูปที่ 10 ภาพรวมของกิจกรรมการสร้างโมเดลสำหรับประมาณค่าความเสถียร ด้วยวิธี MRA, PCA, DLREA และ DLREA(PCA) 31
รูปที่ 11 ภาพรวมของการสร้างโมเดลที่ใช้ตัววัดแผนภาพ (MRA, PCA) 32
รูปที่ 12 ตัวอย่างวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) 34
รูปที่ 13 แนวคิดของวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ 35
รูปที่ 14 ตัวอย่างแผนภาพคลาสและแผนภาพลำดับ 38
รูปที่ 15 กราฟแสดงค่าสัมประสิทธิ์การถดถอยของตัววัด 27 ตัววัด 47
รูปที่ 16 กราฟแสดงค่าสัมประสิทธิ์การถดถอยของตัววัด 13 ตัววัด 52
รูปที่ 17 กราฟแสดงค่าสัมประสิทธิ์การถดถอยของตัววัด 4 ตัววัด 56
รูปที่ 18 แผนภาพยูสเคสของเครื่องมือ 71
รูปที่ 19 แผนภาพคลาสแสดงองค์ประกอบของเครื่องมือ 75
รูปที่ 20 แผนภาพลำดับแสดงการนำเข้าไฟล์สู่ระบบงาน 77
รูปที่ 21 แผนภาพลำดับแสดงการเตรียมข้อมูล 78
รูปที่ 22 แผนภาพลำดับการแสดงผลค่าประมาณความเสถียรเชิงตรรกะของคลาส 79
รูปที่ 23 แผนภาพกิจกรรมของเครื่องมือการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล 80
รูปที่ 24 ส่วนต่อประสานกับผู้ใช้ของขั้นตอนการนำเข้าไฟล์ 82
รูปที่ 25 ส่วนต่อประสานกับผู้ใช้ของขั้นตอนการแสดงผล 83
รูปที่ 26 การตั้งค่าโอดีบีซีในหน้าต่าง ODBC Data Source Administrator 91

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

รูปที่ 27 การตั้งค่าโอดีบีซีในหน้าต่าง ODBC Microsoft Access Setup..... 91
รูปที่ 28 หน้าต่างของโปรแกรม Logical Stability Measure Tool..... 92
รูปที่ 29 หน้าต่างเลือกไฟล์ซอร์ซโค้ดที่ต้องการคำนวณค่าความเสถียรเชิงตรรกะของคลาส 92
รูปที่ 30 หน้าต่างแสดงไฟล์ที่นำมาประมวลผลเพื่อคำนวณค่า ความเสถียรเชิงตรรกะของคลาส 93
รูปที่ 31 หน้าต่างแสดงข้อความเมื่อมีการคำนวณและบันทึกลงในฐานข้อมูลเรียบร้อยแล้ว 93
รูปที่ 32 หน้าต่างแสดงตัวอย่างผลการคำนวณค่าความเสถียรเชิงตรรกะของคลาส 94
รูปที่ 33 หน้าต่างหลักของโปรแกรม MagicDraw UML..... 95
รูปที่ 34 หน้าต่างสร้างโครงการใหม่ของโปรแกรม MagicDraw UML..... 95
รูปที่ 35 หน้าต่างแสดงการสร้างชุดคำสั่งภาษาคอมพิวเตอร์..... 96
รูปที่ 36 หน้าต่างแสดงการแก้ไขชุดคำสั่งภาษาคอมพิวเตอร์..... 96
รูปที่ 37 หน้าต่างแสดงการจัดการซอร์ซโค้ดที่ต้องการแปลงกลับเป็นแผนภาพคลาส 97
รูปที่ 38 หน้าต่างแสดงรายการไฟล์ซอร์ซโค้ดภาษาจาวาที่ต้องการแปลงกลับ เป็นแผนภาพคลาส .. 97
รูปที่ 39 หน้าต่างแสดงรายการไฟล์ซอร์ซโค้ดภาษาจาวาในชุดคำสั่งภาษาคอมพิวเตอร์..... 98
รูปที่ 40 หน้าต่างแสดงการแปลงกลับจากซอร์ซโค้ดเป็นแผนภาพคลาส..... 98
รูปที่ 41 หน้าต่างตัวเลือกการแปลงกลับ 99
รูปที่ 42 หน้าต่างแสดงการตั้งค่าของตัวเลือกการแปลงกลับ 99
รูปที่ 43 หน้าต่างแสดงแผนภาพคลาสที่ได้จากการแปลงกลับ..... 100
รูปที่ 44 หน้าต่างแสดงการแปลงกลับแผนภาพลำดับ..... 100
รูปที่ 45 หน้าต่างแสดงแผนภาพลำดับที่ได้จากการแปลงกลับ..... 101
รูปที่ 46 หน้าต่างบันทึกโครงการเป็นไฟล์ xml..... 101
รูปที่ 47 หน้าต่างหลักของโปรแกรม SDMetrics เวอร์ชัน 2.3 102
รูปที่ 48 หน้าต่างแสดงการตั้งค่าโครงการ 102
รูปที่ 49 หน้าต่าง Project Settings..... 103
รูปที่ 50 หน้าต่างแสดงเมนูการคำนวณค่าตัววัดแผนภาพ..... 103
รูปที่ 51 หน้าต่างแสดงค่าตัววัดแผนภาพ 104
รูปที่ 52 หน้าต่างแสดงการบันทึกค่าตัววัดแผนภาพเป็นไฟล์ csv..... 104
รูปที่ 53 หน้าต่างหลักของเครื่องมือประมาณค่าความเสถียร..... 106
รูปที่ 54 ตัวอย่างขั้นตอนการนำเข้าไฟล์..... 106
รูปที่ 55 ตัวอย่างขั้นตอนการตรวจสอบไฟล์และเตรียมข้อมูล..... 107
รูปที่ 56 ตัวอย่างเมื่อเสร็จสิ้นขั้นตอนการตรวจสอบไฟล์และเตรียมข้อมูล 108

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

รูปที่ 57 ตัวอย่างของการแสดงผลกราฟและตารางค่าประมาณความเสถียร..... 109
รูปที่ 58 ไฟล์ทั้งหมดที่ได้จากการบันทึกไฟล์ค่าประมาณความเสถียรเชิงตรรกะของคลาส..... 109

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

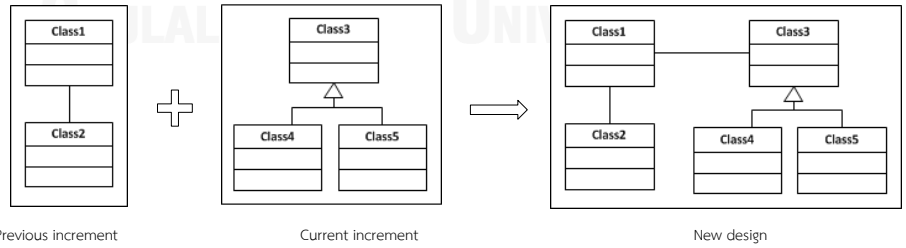
บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงภาพรวมของงานวิจัย ซึ่งประกอบด้วย ที่มาและความสำคัญ ปัญหา วัตถุประสงค์ ขอบเขต ประโยชน์ ขั้นตอนและวิธีดำเนินการวิจัย และโครงสร้างของเนื้อหางานวิจัย ซึ่งมีเนื้อหาดังต่อไปนี้

1.1 ที่มาและความสำคัญของปัญหา

ความเสถียรของซอฟต์แวร์ (Software stability) เป็นตัวชี้วัดหนึ่งที่บ่งบอกถึงคุณภาพของซอฟต์แวร์ (Software quality) ซึ่งเป็นค่าความต้านทานต่อการได้รับผลกระทบที่เกิดจากการเปลี่ยนแปลงซอฟต์แวร์ (Software change) [1, 2] ความเสถียรของซอฟต์แวร์แบ่งเป็น 2 ประเภท คือ ความเสถียรเชิงตรรกะ (Logical stability) ซึ่งเกี่ยวข้องกับเชิงโครงสร้างของซอฟต์แวร์ และความเสถียรเชิงประสิทธิภาพ (Performance stability) ซึ่งเกี่ยวข้องกับเชิงพฤติกรรมของซอฟต์แวร์ในขณะการทำงานจริงของซอฟต์แวร์

การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล (Incremental software development) เป็นกระบวนการการพัฒนาซอฟต์แวร์หนึ่งที่มีการแบ่งการซอฟต์แวร์ที่จะพัฒนาออกเป็นส่วนๆ เรียกว่า อินคริเมนต์ (Increment) แต่ละอินคริเมนต์แบ่งการทำงานออกเป็นขั้นตอน ประกอบด้วย วิเคราะห์ (Analysis) ออกแบบ (Design) พัฒนา (Develop) และทดสอบ (Test) โดยกระบวนการพัฒนาแบบอินคริเมนทัลนั้นจะมีการเก็บรวบรวมและวิเคราะห์ความต้องการของซอฟต์แวร์ (Software Requirements) ทั้งหมดก่อนแล้วจึงนำส่วนที่เป็นความต้องการหลักมาทำการออกแบบและพัฒนาก่อนในอินคริเมนต์แรก จากนั้นจึงพัฒนาตามความต้องการที่เหลือในในอินคริเมนต์ต่อไปจนเสร็จสิ้นครบทุกความต้องการ ซึ่งเมื่ออินคริเมนต์ปัจจุบัน (Current increment) ได้ถูกเพิ่มไปยังส่วนของการออกแบบของอินคริเมนต์ก่อนหน้า (Previous increment) ทำให้เกิดโครงสร้างการออกแบบใหม่ (New design) ดังรูปที่ 1 ซึ่งเป็นความเปลี่ยนแปลงที่เกิดขึ้นกับโครงสร้างการออกแบบของอินคริเมนต์ก่อนหน้า



รูปที่ 1 การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล

การวัดค่าความเสถียรของซอฟต์แวร์สามารถแบ่งออกเป็น 2 ขั้นตอน คือ ในขั้นตอนของการบำรุงรักษา (Maintenance phase) เป็นการวัดค่าความเสถียรจากซอร์ซโค้ด (Source code) และ

ในขั้นตอนการออกแบบ (Design phase) เป็นการประมาณค่าความเสถียรจากแผนภาพการออกแบบซอฟต์แวร์ ซึ่งส่วนใหญ่งานวิจัยที่ผ่านมาเป็นการนำเสนออัลกอริทึมและสูตรคำนวณค่าความเสถียร [1, 3] โดยทำการวัดค่าจากซอร์ซโค้ด หลังจากการพัฒนาซอฟต์แวร์เสร็จสิ้นแล้ว แต่ Yau และ Collofello [1] เกิดแนวคิดที่จะสร้างอัลกอริทึมเพื่อคำนวณความเสถียรในขั้นตอนของการออกแบบซอฟต์แวร์ ก่อนจะเข้าสู่ขั้นตอนของการพัฒนา (Development phase) เพื่อลดต้นทุนในขั้นตอนของการบำรุงรักษา เนื่องจากขั้นตอนการบำรุงรักษานั้นส่งผลกระทบมากต่อต้นทุนของการพัฒนาซอฟต์แวร์ จากแนวคิดดังกล่าวนี้จึงมีงานวิจัยที่นำเสนอแนวคิดในการสร้างโมเดลเพื่อการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ซึ่งจะช่วยให้ผู้พัฒนาสามารถนำค่าความเสถียรที่ได้ไปใช้ในการตัดสินใจแก้ไขโครงสร้างการออกแบบก่อนขั้นตอนการพัฒนา โมเดลการประมาณค่าความเสถียรนั้นมีจำนวนมากซึ่งมีความแตกต่างกันด้วย ปัจจัย (Factors) ตัววัด (Indicators) และวิธีการในการสร้างโมเดล (Methods) งานวิจัยที่ผ่านมาใช้ตัววัดที่แตกต่างกันคือ ตัววัดที่มีผู้วิจัยอื่นๆ ได้นำเสนอไว้ [4, 5, 6] และตัววัดที่กำหนดขึ้นเอง [2] นอกจากนี้ยังนำเสนอวิธีการสร้างโมเดลที่แตกต่างกัน เช่น การวิเคราะห์ความถดถอยเชิงซ้อน [4] การอุปมา [5] ข่ายงานประสาทเทียม [6] เป็นต้น

การเปลี่ยนแปลงของโครงสร้างการออกแบบซอฟต์แวร์นั้นเป็นปัจจัยหนึ่งที่มีผลกระทบต่อความเสถียรของซอฟต์แวร์และคุณสมบัติอื่นๆที่บ่งบอกถึงคุณภาพของซอฟต์แวร์ ซึ่งในการพัฒนาแบบอินคริเมนทัลมีการเปลี่ยนแปลงของโครงสร้างการออกแบบที่เกิดขึ้นเมื่อเพิ่มการพัฒนาในแต่ละอินคริเมนต์ การเปลี่ยนแปลงนี้อาจส่งผลกระทบต่อองค์ประกอบของอินคริเมนต์ก่อนหน้า ทำให้ค่าความเสถียรเปลี่ยนแปลงไป หากความเสถียรมีค่าลดลงหมายถึงการเพิ่มการพัฒนาในอินคริเมนต์นั้นๆ มีการเปลี่ยนแปลงที่อาจส่งผลกระทบทำให้การทำงานผิดพลาดหรือไม่สามารถทำงานได้ ซึ่งจะต้องนำค่าความเสถียรที่ได้ไปพิจารณาและทำการแก้ไขการออกแบบ (Re-design) ด้วยเหตุนี้จึงต้องมีการวัดค่าความเสถียรเมื่อมีการเพิ่มการพัฒนาในแต่ละอินคริเมนต์เพื่อเป็นตัวชี้วัดคุณภาพของซอฟต์แวร์ แต่หากทำการประมาณค่าความเสถียรตั้งแต่ขั้นตอนของการออกแบบเพื่อการแก้ไขโครงสร้างการออกแบบก่อนจะเข้าสู่ขั้นตอนของการพัฒนาก็จะช่วยลดต้นทุนในขั้นตอนของการบำรุงรักษา เพราะถ้ามีการแก้ไขเมื่อซอฟต์แวร์พัฒนาเสร็จเรียบร้อยแล้วก็จะเกิดความยุ่งยาก ซับซ้อน และใช้ต้นทุนสูงในการบำรุงรักษาซอฟต์แวร์

งานวิจัยนี้มีแนวคิดที่จะทำการประมาณค่าของความเสถียรตั้งแต่ขั้นตอนของการออกแบบเมื่อเพิ่มการพัฒนาในแต่ละอินคริเมนต์ โดยจะใช้ข้อมูลจากโครงสร้างการออกแบบหรือแผนภาพในการประมาณค่าความเสถียร และเสนอโมเดลในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล โดยใช้แผนภาพคลาสเพื่อแสดงการออกแบบเชิงโครงสร้างขององค์ประกอบและใช้แผนภาพลำดับเพื่อแสดงการออกแบบเชิงพฤติกรรม และเนื่องจากการเปลี่ยนแปลงในแต่ละอินคริเมนต์เกิดขึ้นในระดับของคลาสและระดับของระบบ ซึ่งส่งผลกระทบต่อคลาสใดๆ ภายในโปรแกรม ความเสถียรในที่นี้จึงหมายถึงความเสถียรเชิงตรรกะของคลาสซึ่งเป็นค่าความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงของคลาสใดๆในโปรแกรม โดยงานวิจัยนี้จะใช้ตัววัดที่แตกต่างกันคือ ตัววัดที่ได้นำเสนอโดยผู้วิจัยในอดีต [4, 5, 6] และตัววัดที่กำหนดขึ้นใหม่ภายใต้ข้อกำหนดของโมเดล นอกจากนี้ยังใช้วิธีการในการสร้างโมเดลที่แตกต่างกัน 4 วิธี คือ โมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (Multiple regression analysis) โมเดลที่ 2 สร้าง

ด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (Principal component analysis) โมเดลที่ 3 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (Design logical ripple effect analysis) และโมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล การสร้างโมเดลประมาณค่าความเสถียรด้วยวิธีที่แตกต่างกัน 4 วิธี มีแนวคิดจากงานวิจัยที่ผ่านมานั้นใช้ตัววัดที่มีผู้วิจัยในอดีตได้นำเสนอไว้และไม่มีการคัดเลือกตัววัดที่จะนำมาใช้ในการสร้างโมเดล ใน 2 โมเดลแรกของงานวิจัยนี้จึงทำการเปรียบเทียบระหว่างโมเดลที่ไม่มีการคัดเลือกตัววัดคือโมเดลที่ 1 ซึ่งนำตัววัดแผนภาพทั้งหมดมาสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน และโมเดลที่มีการคัดเลือกตัววัดที่มีความสัมพันธ์กับความเสถียรคือโมเดลที่ 2 ซึ่งนำวิธีการวิเคราะห์ส่วนประกอบหลักมาใช้ในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล แต่ตัววัดเหล่านี้เป็นตัววัดที่แสดงถึงจำนวนองค์ประกอบของโครงสร้างการออกแบบซึ่งไม่สอดคล้องกับนิยามของความเสถียรที่หมายถึงค่าความต้านทานผลกระทบที่เกิดจากการเปลี่ยนแปลงของซอฟต์แวร์ จึงมีแนวคิดที่จะกำหนดตัววัดขึ้นใหม่เพื่อให้สอดคล้องกับนิยามของความเสถียร ซึ่งพิจารณาจากผลกระทบจากการเปลี่ยนแปลงของโครงสร้างการออกแบบซอฟต์แวร์ จึงนำเสนอโมเดลที่ 3 และโมเดลที่ 4 ซึ่งสร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ แต่ในโมเดลที่ 4 นั้นนำวิธีการวิเคราะห์ส่วนประกอบหลักมาใช้ในการคัดเลือกตัววัดก่อนการสร้างโมเดล แล้วทำการเปรียบเทียบโมเดลทั้ง 4 โมเดลด้วยผลการประเมินผลโมเดล ซึ่งผลของการประเมินผลโมเดลพบว่าการใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนการสร้างโมเดลจะได้โมเดลที่ให้ค่าประมาณที่มีความแม่นยำมากขึ้น โมเดลที่ 2 ให้ค่าประมาณที่ใกล้เคียงค่าจริงมากกว่าโมเดลที่ 1 และโมเดลที่ 4 ให้ค่าประมาณที่มีความถูกต้องใกล้เคียงค่าจริงมากกว่าโมเดลที่ 3 ซึ่งจากทั้ง 4 โมเดลนี้ โมเดลที่ 2 ให้ค่าประมาณที่มีความถูกต้องใกล้เคียงค่าจริงมากที่สุด

1.2 วัตถุประสงค์ของงานวิจัย

1) นำเสนอแนวคิดในการสร้างโมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล จากแผนภาพคลาสและแผนภาพลำดับ
2) ออกแบบและพัฒนาเครื่องมือสำหรับการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล จากแผนภาพคลาสและแผนภาพลำดับตามโมเดลที่ได้นำเสนอ

1.3 ขอบเขตของงานวิจัย

1) การวัดความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ดนั้นวิเคราะห์จากการเปลี่ยนแปลงและผลกระทบที่กำหนดไว้ในวิทยานิพนธ์ฉบับนี้
2) งานวิจัยนี้ทำการเสนอโมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์จากแผนภาพคลาสและแผนภาพลำดับ 4 โมเดล คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (Multiple regression analysis โดยใช้คำย่อว่า MRA) โมเดลที่สร้างด้วย

วิธีการวิเคราะห์ส่วนประกอบหลัก (Principal component analysis โดยใช้คำย่อว่า PCA) โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (Design logical ripple effect analysis โดยใช้คำย่อว่า DLREA) และโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล (ใช้คำย่อว่า DLREA(PCA))

3) ตัววัดแผนภาพที่นำมาสร้างโมเดลมีทั้งหมด 27 ตัววัด ซึ่งอธิบายรายละเอียดไว้ในวิทยานิพนธ์ฉบับนี้

4) พัฒนาเครื่องมือสำหรับการประมาณค่าความเสถียรของการออกแบบจากแผนภาพคลาสและแผนภาพลำดับ โดยแสดงคลาสที่ควรนำไปพิจารณาปรับปรุงเท่านั้น มิได้เสนอแนวทางในการปรับปรุงแก้ไข ซึ่งพัฒนาด้วยภาษาจาวาและทดสอบบนระบบปฏิบัติการวินโดวส์ (Windows)

5) โมเดลในการประมาณค่าความเสถียรของการออกแบบในงานวิจัยนี้ใช้สำหรับซอฟต์แวร์เชิงวัตถุที่พัฒนาด้วยภาษาจาวา และออกแบบด้วยแผนภาพคลาสและแผนภาพลำดับของยูเอ็มแอลเวอร์ชัน 2.0

6) โมเดลในการประมาณค่าความเสถียรของการออกแบบในงานวิจัยนี้ ใช้ชุดโปรแกรมสำหรับสร้างโมเดลเป็นซอฟต์แวร์ที่พัฒนาแบบอินคริเมนทัลจำนวน 5 ซอฟต์แวร์ และประเมินผลโมเดลจากซอฟต์แวร์ที่พัฒนาแบบอินคริเมนทัลจำนวน 15 ซอฟต์แวร์ โดยแต่ละซอฟต์แวร์มีจำนวนอินคริเมนต์อย่างน้อย 2 อินคริเมนต์

1.4 ประโยชน์ของงานวิจัย

1) ได้โมเดลการประมาณค่าความเสถียรเชิงตรรกะของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัลไปปรับใช้ เพื่อเป็นแนวทางให้ผู้พัฒนาซอฟต์แวร์พิจารณาความเสถียรในขั้นตอนการออกแบบก่อนการนำไปพัฒนาซึ่งจะช่วยลดต้นทุนในการบำรุงรักษา

2) ได้เครื่องมือสำหรับการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล

1.5 ขั้นตอนและวิธีดำเนินการวิจัย

1) ศึกษาทฤษฎีพื้นฐานของความเสถียรของซอฟต์แวร์ การออกแบบและการพัฒนาซอฟต์แวร์แบบอินคริเมนทัล ตัววัดของแผนภาพคลาสและแผนภาพลำดับ และวิธีการในการสร้างโมเดลสำหรับการประมาณค่าจากหนังสือ และบทความวิชาการต่างๆ

2) วิเคราะห์ทฤษฎีที่จะนำมาใช้ในการประเมินค่าความเสถียรของการออกแบบซอฟต์แวร์

3) รวบรวมโปรแกรมภาษาจาวาจากแหล่งโอเพ่นซอร์ซซอฟต์แวร์ เพื่อนำมาใช้ในการสร้างและประเมินผลโมเดล

4) เตรียมข้อมูลความเสถียรเชิงตรรกะของคลาสที่คำนวณได้เป็นค่าจริงจากซอร์ซโค้ด และค่าของตัววัดแผนภาพที่ได้จากแผนภาพคลาสและแผนภาพลำดับของโปรแกรมที่รวบรวมไว้

5) สร้างโมเดลการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ทั้งหมด 4 วิธี คือวิธีการวิเคราะห์ความถดถอยเชิงซ้อน วิธีการวิเคราะห์ส่วนประกอบหลัก วิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ และวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล

6) ประเมินผลโมเดลด้วยซอฟต์แวร์ที่มีการพัฒนาแบบอินคริเมนทัล

7) เปรียบเทียบผลการประเมินผลโมเดล

8) วิเคราะห์ผลการทดลอง ออกแบบและพัฒนาเครื่องมือ

9) สรุปผล ข้อเสนอแนะ งานวิจัยในอนาคต และเรียบเรียงวิทยานิพนธ์

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 6 บทคือ บทที่ 1 เป็นบทนำ บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง บทที่ 3 นำเสนอแนวคิดของงานวิจัยและขั้นตอนในการสร้างโมเดลสำหรับการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล ซึ่งประกอบด้วยขั้นตอนต่างๆในการดำเนินการวิจัย รวมทั้งแนวคิดและรายละเอียดในการสร้างโมเดลด้วยวิธีการที่แตกต่างกันทั้ง 3 โมเดล คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) และโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล (DLREA(PCA)) บทที่ 4 เป็นบทที่แสดงผลการสร้างโมเดลและประเมินผลโมเดล พร้อมทั้งเปรียบเทียบโมเดลที่ได้นำเสนอทั้งหมด 3 โมเดล บทที่ 5 เป็นบทที่อธิบายการออกแบบและพัฒนาเครื่องมือตามโมเดลที่นำเสนอ และบทที่ 6 เป็นบทสุดท้าย ซึ่งจะเป็นบทสรุปของงานวิจัย ข้อจำกัด รวมทั้งข้อเสนอแนะงานวิจัยในอนาคต และบทความวิชาการที่ตีพิมพ์

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะอธิบายถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดดังนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

ในส่วนนี้อธิบายเกี่ยวกับความรู้พื้นฐาน ทฤษฎี เกี่ยวกับการสร้างโมเดลการคำนวณความเสถียร และนิยามของผลกระทบ การเปลี่ยนแปลงของคลาส และการพัฒนาซอฟต์แวร์แบบอินคริเมนทัลมาใช้ในการออกแบบการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล โดยมีทฤษฎีที่เกี่ยวข้อง ดังนี้

2.1.1 ความเสถียรของซอฟต์แวร์ (Software Stability)

ความเสถียรของซอฟต์แวร์เป็นความต้านทานต่อผลกระทบที่เกิดจากความเปลี่ยนแปลงที่เกิดขึ้นกับซอฟต์แวร์ ซึ่งเป็นคุณสมบัติหนึ่งที่บ่งบอกถึงคุณภาพของซอฟต์แวร์ [1, 2] แบ่งออกเป็น 2 ประเภท คือ

2.1.1.1 ความเสถียรของซอฟต์แวร์เชิงตรรกะ (Logical Stability)

ความเสถียรของซอฟต์แวร์เชิงตรรกะ คือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้การทำงานผิดพลาดหรือไม่สามารถทำงานได้ ซึ่งจำแนกออกเป็น 2 ประเภทคือ

1) ความเสถียรเชิงตรรกะของคลาส (Logical stability of a class) คือ ค่าความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงของคลาสใดๆในโปรแกรม เช่น คลาสหนึ่งมีค่าความเสถียรเชิงตรรกะเท่ากับ 0.85 จะหมายความว่าหากเกิดความเปลี่ยนแปลงขึ้นในคลาสอื่นๆ โอกาสที่คลาสนี้จะไม่ได้รับผลกระทบเชิงตรรกะจะมีค่าเท่ากับ 85% หรือในการเปลี่ยนแปลง 100 ครั้ง คลาสนี้จะยังทำงานได้คงเดิมและไม่ได้รับผลกระทบเชิงตรรกะ 85 ครั้ง เป็นต้น
2) ความเสถียรเชิงตรรกะของโปรแกรม (Logical stability of a program) คือ ค่าความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงของโปรแกรม

2.1.1.2 ความเสถียรของซอฟต์แวร์เชิงประสิทธิภาพ (Performance Stability)

ความเสถียรของซอฟต์แวร์เชิงประสิทธิภาพ คือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้ประสิทธิภาพในการทำงานเปลี่ยนไป การหาค่าความเสถียรของซอฟต์แวร์เชิงประสิทธิภาพต้องกำหนดประสิทธิภาพที่สนใจ มีความซับซ้อนสูงในการวิเคราะห์ จะต้องใช้การวิเคราะห์แบบพลวัต (Dynamic Analysis) ซึ่งเป็นการวิเคราะห์ระหว่างการทำงานจริงของซอฟต์แวร์

ในงานวิจัยนี้จะมุ่งเน้นเฉพาะความเสถียรเชิงตรรกะ ซึ่งจะพิจารณาทั้งโครงสร้างของภายในคลาส และคลาสอื่นในโปรแกรมเดียวกัน โดยใช้แผนภาพคลาสเพื่อให้ทราบถึงการออกแบบเชิง

โครงสร้างขององค์ประกอบ และใช้แผนภาพลำดับเพื่อให้ทราบถึงการออกแบบเชิงพฤติกรรม นำมาประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์

2.1.2 ผลกระเพื่อมหรือผลกระทบที่เกิดจากการเปลี่ยนแปลง (Ripple effect)

ผลกระเพื่อมหรือผลกระทบที่เกิดจากการเปลี่ยนแปลง(Ripple effect) [8] แบ่งออกเป็นหลายประเภท ได้แก่ ผลกระทบเชิงตรรกะ (Logical ripple effect), ผลกระทบเชิงประสิทธิภาพ (Performance ripple effect), ผลกระทบเชิงการเขียนโปรแกรม (Coding ripple effect), ผลกระทบเชิงข้อมูล (Data ripple effect), ผลกระทบเชิงเอกสาร (Document ripple effect), ผลกระทบเชิงส่วนต่อประสาน (Interface ripple effect), ผลกระทบเชิงสิ่งแวดล้อม (Environment ripple effect), ผลกระทบเชิงการจัดการ (Management ripple effect) เป็นต้น

ในงานวิจัยนี้จะมุ่งเน้นเฉพาะผลกระทบเชิงตรรกะ(Logical ripple effect) ซึ่งเป็นผลกระทบที่เกิดจากการเปลี่ยนแปลงของคลาส 1 คลาสส่งผลให้เกิดความไม่สอดคล้องของคลาสที่อยู่ในโปรแกรมเดียวกัน เช่น การเปลี่ยนชื่อของแอททริบิวต์ เป็นต้น

2.1.3 การเปลี่ยนแปลง (Changes)

การเปลี่ยนแปลง (Changes) เป็นการเปลี่ยนแปลงของคลาสที่ได้ทำการรวบรวม [9, 10] และนำมาปรับใช้ในงานวิจัยนี้แสดงดังตารางที่ 1 ซึ่งการเปลี่ยนแปลงที่เกิดขึ้นจะต้องส่งผลกระทบต่อคลาสนั้นๆ หรือคลาสอื่นๆในโปรแกรมเดียวกัน แบ่งออกเป็น 2 ประเภท คือ

1) การเปลี่ยนแปลงในระดับของระบบ (System level change) คือ การเปลี่ยนแปลงที่ส่งผลต่อความสัมพันธ์ของคลาสในโปรแกรม
2) การเปลี่ยนแปลงในระดับของคลาส (Class level change) คือ การเปลี่ยนแปลงที่ส่งผลต่อองค์ประกอบภายในคลาส แบ่งออกเป็น 2 ประเภท คือ
    1) การเปลี่ยนแปลงในระดับแอททริบิวต์ (change at attribute)
    2) การเปลี่ยนแปลงในระดับเมธอด (change at method)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 1 ประเภทของการเปลี่ยนแปลง

ประเภท	การเปลี่ยนแปลง		คลาสที่ได้รับผลกระทบ
การเปลี่ยนแปลงในระดับของคลาส	1	เพิ่มเมธอดของคลาส C	S (เมื่อขอบเขตเป็นพับบลิคเท่านั้น)
	2	ลบเมธอดของคลาส C (ขอบเขตเป็นพับบลิค/โพรเทคต์/ไพรเวต)	C, Y, S / C, S / C
	3	เปลี่ยนชื่อ/เปลี่ยนซิกเนเจอร์/เปลี่ยนชนิดข้อมูลคืนกลับ/เปลี่ยนรายละเอียดการทำงาน เมธอดของคลาส C (ขอบเขตเป็นพับบลิค/โพรเทคต์/ไพรเวต)	C, Y, S / C, S / C
	4	เปลี่ยนขอบเขตการเข้าถึงเมธอดของคลาส C - พับบลิคเป็นไพรเวต - พับบลิคเป็นโพรเทคต์ - โพรเทคต์เป็นไพรเวต - โพรเทคต์เป็นพับบลิค - ไพรเวตเป็นพับบลิค - ไพรเวตเป็นโพรเทคต์	 S, Y Y S - - -
	5	เพิ่มแอททริบิวต์ของคลาส C	S (เมื่อขอบเขตเป็นพับบลิคเท่านั้น)
	6	ลบแอททริบิวต์ของคลาส C (ขอบเขตเป็นพับบลิค/โพรเทคต์/ไพรเวต)	C, X, S / C, S / C
	7	ลบ/เปลี่ยนชื่อ/เปลี่ยนประเภท/เปลี่ยนค่าแอททริบิวต์ของคลาส C (ขอบเขตเป็นพับบลิค/โพรเทคต์/ไพรเวต)	C, X, S / C, S / C
	8	เปลี่ยนขอบเขตการเข้าถึงแอททริบิวต์ของคลาส C - พับบลิคเป็นไพรเวต - พับบลิคเป็นโพรเทคต์ - โพรเทคต์เป็นไพรเวต - โพรเทคต์เป็นพับบลิค - ไพรเวตเป็นพับบลิค - ไพรเวตเป็นโพรเทคต์	 S, X X S - - -

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 2 ประเภทของการเปลี่ยนแปลง (ต่อ)

ประเภท	การเปลี่ยนแปลง		คลาสที่ได้รับผลกระทบ
การเปลี่ยนแปลงในระดับของระบบ	9	เพิ่ม/ลบ ความสัมพันธ์คลาสแม่ของ C	C, S / C, S, R
	10	เพิ่ม/ลบ ความสัมพันธ์คลาสลูกของ C	- / -
	11	เพิ่ม/ลบ ความสัมพันธ์แอกกรีเกชันคลาสของ C	C, S / C, S, R
	12	เพิ่ม/ลบ ความสัมพันธ์แอสโซซิเอชันคลาสของ C	C, S / C, S, R
	13	เพิ่ม/ลบ คลาสของ C	- / C, S, R

หมายเหตุ C คือ คลาสที่สนใจ

S คือ คลาสลูกของ C

R คือ คลาสที่มีความสัมพันธ์กับ C ในฐานะผู้กระทำ เช่น เป็นแอสโซซิเอชันคลาสของ C

X คือ คลาสที่เรียกใช้แอททริบิวต์ของ C

Y คือ คลาสที่เรียกใช้เมธอดของ C

- คือ ไม่มีคลาสใดๆ (C, S, R, X, Y) ได้รับผลกระทบจากรูปแบบการเปลี่ยนแปลง นั้นๆ

2.1.4 การคำนวณค่าความเสถียร

การคำนวณความเสถียร มีการนำเสนอการคำนวณไว้หลากหลายและยังไม่มีการคำนวณที่เป็นที่ยอมรับและใช้อย่างแพร่หลาย จึงมีผู้วิจัยคิดค้นและพัฒนาจนถึงปัจจุบัน

2.1.4.1 การคำนวณค่าความเสถียรในขั้นตอนของการบำรุงรักษา

การคำนวณค่าความเสถียรในขั้นตอนของการบำรุงรักษาหรือเมื่อมีการพัฒนาซอฟต์แวร์เสร็จสิ้นแล้ว ซึ่งคำนวณค่าได้จากซอร์ซโค้ด มีการนำเสนอไว้หลากหลาย เช่น

1) การคำนวณค่าความเสถียรเชิงตรรกะของคลาส (Logical Stability of Class) ซึ่งเป็นสูตรคำนวณของ Elish and Rine [3]

$$CLS_i = 1 - \frac{NTE_i}{TNC} \quad (1)$$

โดยที่ CLS_i คือ ค่าความเสถียรเชิงตรรกะของคลาส C_i เมื่อ $i = 1,2,..,n$

NTE_i คือ จำนวนครั้งที่คลาส C_i ได้รับผลกระทบ เมื่อ $i = 1,2,..,n$

TNC คือ จำนวนรวมของการเปลี่ยนแปลง

2) สูตรคำนวณของ Alshayeb, naji, Elish, Al-ghamdi [7] เสนอการคำนวณการวัดความเสถียรของคลาสเชิงวัตถุ โดยจำแนกคุณสมบัติที่มีผลต่อความเสถียรของคลาส 8 คุณสมบัติ คือ

1) ระดับการเข้าถึงของคลาส (Class access-level)
2) ชื่อส่วนต่อประสานของคลาส (Class interface name)
3) ชื่อคลาสที่สืบทอดคุณสมบัติ (Inherited class name)
4) ตัวแปรคลาส (Class variable) หมายถึงตัวแปรที่ประกาศไว้ในคลาส
5) ระดับการเข้าถึงของตัวแปร (Class variable access-level)
6) รายการพารามิเตอร์ของเมธอด (Method signature)
7) ระดับการเข้าถึงของเมธอด (Method access-level)
8) คำสั่งต่างๆที่อยู่ในเมธอด (Method body)

เมื่อโปรแกรมเวอร์ชันต่างกันสามารถเปรียบเทียบการเปลี่ยนแปลงของคลาสเดียวกัน โดยประกอบด้วยการเปลี่ยนแปลง 4 ประเภทคือ

1) เพิ่ม คือ คุณสมบัตินั้นไม่ปรากฏในเวอร์ชัน i และถูกเพิ่มในเวอร์ชัน i+1 (i=1,2,...,n)
2) ลบ คือ คุณสมบัตินั้นปรากฏในเวอร์ชัน i และถูกลบออกในเวอร์ชัน i+1
3) แก้ไข คือ คุณสมบัตินั้นปรากฏในเวอร์ชัน i และถูกแก้ไขในเวอร์ชัน i+1
4) ไม่เปลี่ยนแปลง คือ คุณสมบัตินั้นปรากฏในเวอร์ชัน i และไม่มีการเปลี่ยนแปลงในเวอร์ชัน i+1

$$\text{Stability}_{\text{CLASS}} = \frac{(\text{Stab}_{\text{ClassAL}} + \text{Stab}_{\text{Interface}} + \text{Stab}_{\text{Inhr}} + \text{Stab}_{\text{Mthd}} + \text{Stab}_{\text{Var}} + \text{Stab}_{\text{varAL}} + \text{Stab}_{\text{MthdAL}} + \text{Stab}_{\text{Body}})}{\text{properties}_{\text{CLASS}}}$$

$$\text{Stab}_{\text{Property}} = \frac{\text{Unchanged}_{\text{Property}}}{\text{Number}_{\text{Property}}} \quad (2)$$

โดยที่ $\text{Stab}_{\text{Property}}$ คือ ความเสถียรของคุณสมบัติในแต่ละคุณสมบัติทั้ง 8 คุณสมบัติ ได้แก่ $\text{Stab}_{\text{ClassAL}}$, $\text{Stab}_{\text{Interface}}$, $\text{Stab}_{\text{Inhr}}$, $\text{Stab}_{\text{Mthd}}$, Stab_{Var}, $\text{Stab}_{\text{viral}}$, $\text{Stab}_{\text{MthdAL}}$, $\text{Stab}_{\text{Body}}$

$\text{Unchanged}_{\text{Property}}$ คือ จำนวนรายการที่ไม่มีการเปลี่ยนแปลงของคุณสมบัตินั้นๆ

$\text{Number}_{\text{Property}}$ คือ จำนวนรายการของคุณสมบัตินั้นๆ

$\text{properties}_{\text{CLASS}}$ คือ จำนวนของคุณสมบัติในคลาสนั้นๆ

2.1.4.2 การคำนวณค่าความเสถียรในขั้นตอนของการออกแบบ

การคำนวณค่าความเสถียรในขั้นตอนของการออกแบบ มีงานวิจัยนำเสนอสูตรคำนวณหรือโมเดลในการประมาณค่าโดยใช้ตัววัดที่มีผู้วิจัยในอดีตได้เสนอไว้และใช้วิธีการในการสร้างโมเดลที่แตกต่างกัน แต่มีผู้วิจัยบางส่วนได้ทำการกำหนดตัววัดหรือตัวแปรขึ้นมาใหม่ คือ Stephen S. Yau และ James S. Collofelo [2] ซึ่งใช้สำหรับการคำนวณความเสถียรสำหรับโปรแกรมเชิงโครงสร้าง โดยคำนวณจากผลกระทบเชิงตรรกะของการออกแบบ โดยผลลัพธ์ที่ได้จะเป็นค่าประมาณ ซึ่งเป็นวิธีการที่งานวิจัยนี้นำมาเป็นแนวคิดในการสร้างโมเดลประมาณค่าความเสถียรด้วยวิธี การวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

$$PDS = 1/(1+\sum_{x} DLRE_x) \quad (3)$$

$$DS_x = 1/(1+DLRE_x)$$

$$DLRE_x = TG_x + \sum_{y \in J_x} TP_{xy} + \sum_{y \in J'_x} TP'_{xy}$$

โดยที่ DS_x คือ ความเสถียรของการออกแบบมอดูล x

PDS คือ ความเสถียรของการออกแบบโปรแกรม

$DLRE_x$ คือ ผลกระทบเชิงตรรกะของการออกแบบมอดูล x

x คือ มอดูลในโปรแกรม

J_x คือ เซตของมอดูลที่เรียกมอดูล x

J'_x คือ เซตของมอดูลที่ถูกเรียกโดยมอดูล x

TP_{xy} คือ จำนวนรวมของการแทนค่าตัวแปรใน R_{xy} โดยมอดูล y

TP'_{xy} คือ จำนวนรวมของการแทนค่าตัวแปรใน R'_{xy} โดยมอดูล y

TG_x คือ จำนวนรวมของการแทนค่ารายการข้อมูลโกลบอลใน GD_x ที่ทำโดยมอดูลอื่น ๆ

R_{xy} คือ เซตของตัวแปรที่ถูกส่งผ่านกลับมาจากมอดูล x ไปยังมอดูล y โดยที่ $y \in J_x$

R'_{xy} คือ เซตของตัวแปรที่ถูกส่งผ่านจากมอดูล x ไปยังมอดูล y โดยที่ $y \in J'_x$

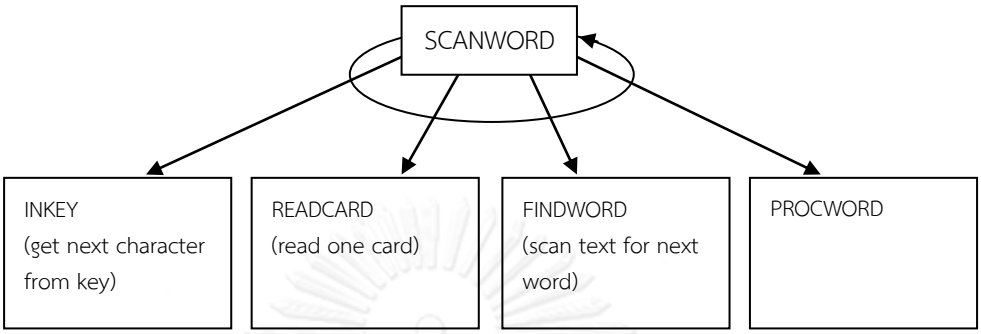
GR_x คือ เซตข้อมูลโกลบอลที่มีการอ้างอิงในมอดูล x

GD_x คือ เซตข้อมูลโกลบอลที่มีการกำหนดไว้ในมอดูล x

G คือ $\{x \mid i \in (GR_x \cup GD_x)\}$

ตัวอย่างการคำนวณความเสถียรของการออกแบบโปรแกรมจากผลกระทบเชิงตรรกะของการออกแบบ [2] แบ่งออกเป็นขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 วิเคราะห์โครงสร้างของโปรแกรมจากเอกสารการออกแบบ



รูปที่ 2 ตัวอย่างการออกแบบแผนผังโครงสร้างของมอดูล

ตารางที่ 2 การออกแบบข้อมูลเข้าและข้อมูลออกของแต่ละมอดูล

มอดูล	ข้อมูลเข้า	ข้อมูลออก
INKEY	-	character, end-of-transmission flag
READCARD	-	card image, last-card flag
FINDWORD	character, end-of-transmission flag, card image, last-card flag, source	word, end-of words flag, get-character flag, get-card flag, word-done flag
PROCWORD	Word	-

จากการออกแบบมอดูลในโปรแกรมดังรูปที่ 2 และตารางที่ 2 นำมาวิเคราะห์ ได้ดังนี้

$J_{SCANWORD} = \{\}$,

$J'_{SCANWORD} = \{INKEY, READCARD, FINDWORD, PROCWORD\}$,

$J_{INKEY} = J_{READCARD} = J_{FINDWORD} = J_{PROCWORD} = \{SCANWORD\}$,

$R_{INKEY, SCANWORD} = \{$character, end-of-transmission flag$\}$,

$R_{READCARD, SCANWORD} = \{$card image, last-card flag$\}$,

$R_{FINDWORD, SCANWORD} = \{$word, end-of-words flag, get character flag, get-card flag, word-done flag$\}$,

$R_{PROCWORD, SCANWORD} = \{\}$,

$R'_{SCANWORD, INKEY} = R'_{SCANWORD, READCARD} = \{\}$,

$R'_{SCANWORD, PROCWORD} = \{$word$\}$,

$R'_{SCANWORD,\ FINDWORD}$ = {character, end-of-transmission flag, card image, last-card flag, source}.

ขั้นตอนที่ 2 หาเซตข้อมูลโกลบอลที่มีการอ้างอิงหรือกำหนดไว้ แต่จากการออกแบบไม่มีข้อมูลโกลบอล

ขั้นตอนที่ 3 หา TP_{xy} หรือจำนวนรวมของการแทนค่าตัวแปรใน R_{xy} โดยมอดูล y

$TP_{INKEY,\ SCANWORD} = 1+1 = 2,$

$TP_{READCARD,\ SCANWORD} = 2+1 = 3$

$TP_{FINDWORD,\ SCANWORD} = 2+1+1+1+1 = 6$

$TP_{PROCWORD,\ SCANWORD} = 0$

ขั้นตอนที่ 4 หา TP'_{xy} หรือจำนวนรวมของการแทนค่าตัวแปรใน R'_{xy} โดยมอดูล y

$TP'_{SCANWORD,\ INKEY} = TP'_{SCANWORD,\ READCARD} = 0,$

$TP'_{SCANWORD,\ PROCWORD} = 2,$

$TP'_{SCANWORD,\ FINDWORD} = 1+1+2+1+1 = 6$

ขั้นตอนที่ 5 $TG = 0$ ในทุกมอดูล เนื่องจากไม่มีเซตข้อมูลโกลบอลที่กำหนดไว้ (จากขั้นตอนที่ 2)

ขั้นตอนที่ 6 คำนวณค่าผลกระทบเชิงตรรกะของการออกแบบมอดูลแต่ละมอดูล

$DLRE_{SCANWORD} = 8$

$DLRE_{INKEY} = 2$

$DLRE_{READCARD} = 3$

$DLRE_{FINDWORD} = 6$

$DLRE_{PROCWORD} = 0$

ขั้นตอนที่ 7 คำนวณค่าความเสถียรของการออกแบบมอดูลแต่ละมอดูล

$DS_{SCANWORD} = 1/9$

$DS_{INKEY} = 1/3$

$DS_{READCARD} = 1/4$

$DS_{FINDWORD} = 1/7$

$DS_{PROCWORD} = 1$

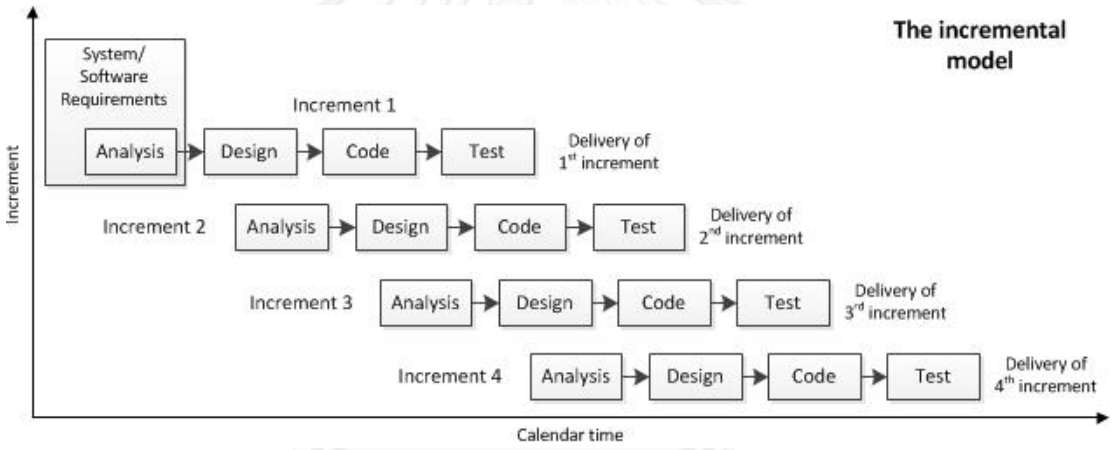
ขั้นตอนที่ 8 คำนวณค่าความเสถียรของการออกแบบโปรแกรมดังสมการที่ 3 ได้เท่ากับ $PDS = 1/20$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ในงานวิจัยนี้จะนำทฤษฎี วิธีการ และข้อจำกัด จากตัวอย่างการคำนวณค่าความเสถียรทั้งหมดในเบื้องต้น ไปใช้เพื่อเป็นแนวทางในการสร้างโมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์

2.1.5 การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล (Incremental software development)

การพัฒนาซอฟต์แวร์แบบอินคริเมนทัลเป็นกระบวนการการพัฒนาซอฟต์แวร์ที่แบ่งงานออกเป็นส่วนย่อยๆ [13] มีการพัฒนาเป็นรอบๆ หรือเรียกว่า อินคริเมนต์ (Increment) ซึ่งจะต้องทราบความต้องการทั้งหมดก่อน โดยจะให้ความสำคัญกับงานที่เป็นความต้องการหลักก่อนที่จะทำการพัฒนาในอินคริเมนต์ถัดไป ซึ่งเป็นกระบวนการการพัฒนาซอฟต์แวร์ที่พัฒนามาจากกระบวนการพัฒนาแบบ Waterfall และหลักการนี้มีการนำไปใช้ในกระบวนการการพัฒนาแบบเอจายล์ [12,13] ซึ่งจะมีการพัฒนาแบบวนซ้ำหากพัฒนาไม่ครบทุกฟังก์ชัน รองรับการเปลี่ยนแปลงของความต้องการตลอดเวลา และมีการส่งมอบงานอย่างต่อเนื่อง โมเดลการพัฒนาซอฟต์แวร์แบบอินคริเมนทัลแสดงดังรูปที่ 3 ซึ่งมีการปรับจากรูปแบบเดิม [15] เพื่อให้เห็นถึงความต้องการทั้งหมดก่อน



รูปที่ 3 การพัฒนาซอฟต์แวร์แบบอินคริเมนทัล

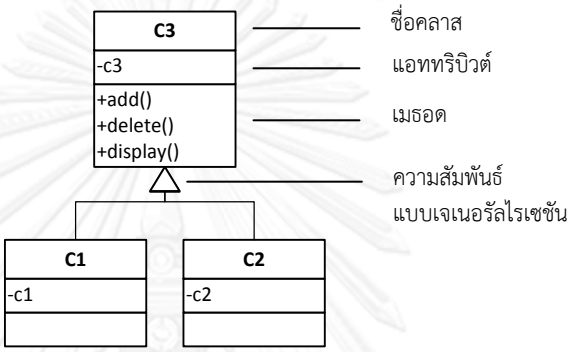
ซอฟต์แวร์ที่มีการพัฒนาแบบอินคริเมนทัลได้แก่ ซอฟต์แวร์ที่มีการปรับปรุงหรือมีฟีเจอร์เพิ่มขึ้น แอพพลิเคชั่นต่างๆ หรือซอฟต์แวร์ที่มีขนาดกลางไปถึงใหญ่ (Large System) ซึ่งกระบวนการการพัฒนาแบบอินคริเมนทัลได้มีการนำมาใช้ในช่วงกลางปี ค.ศ. 1950 และช่วงทศวรรษที่ผ่านมาได้รับความนิยมมากขึ้นเนื่องจากมีโครงการใหญ่ๆนำไปใช้แล้วประสบความสำเร็จ โดยมีผู้วิจัยจำนวนมากให้ความสำคัญกับปัจจัยหรือคุณสมบัติที่จะทำให้ซอฟต์แวร์ที่พัฒนาเพิ่มขึ้นมีคุณภาพจะต้องมีการประเมินค่าของคุณสมบัติต่างๆ ได้แก่ ความพยายาม [13] ความน่าเชื่อถือ [15] เป็นต้น

2.1.6 แผนภาพคลาสและแผนภาพลำดับ

ในงานวิจัยนี้จะนำเสนอโมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ซึ่งเป็นการออกแบบโปรแกรมเชิงวัตถุโดยจะต้องใช้แผนภาพคลาสและแผนภาพลำดับ จึงต้องมีการศึกษาโครงสร้างและส่วนประกอบของแผนภาพเพื่อนำไปใช้ในงานวิจัย

แผนภาพคลาส [16] เป็นแผนภาพที่แสดงโครงสร้างและองค์ประกอบทั้งหมดของระบบ ซึ่งประกอบด้วยส่วนต่างๆ ดังนี้

1) คลาส ภายในคลาสประกอบด้วย
2) ชื่อคลาส แสดงชื่อสิ่งที่เราสนใจในระบบ
3) แอททริบิวต์ แสดงถึงคุณลักษณะของคลาส
4) เมธอด แสดงถึงกิจกรรมหรือการทำงานของคลาส



รูปที่ 4 ตัวอย่างแผนภาพคลาส

ความสัมพันธ์ระหว่างคลาส มีดังนี้

ตารางที่ 3 สัญลักษณ์ความสัมพันธ์แบบต่างๆของแผนภาพคลาส

ความสัมพันธ์	สัญลักษณ์
แอสโซซิเอชัน	
เจเนอรัลไลเซชัน	
แอกกรีเกชันและคอมโพซิชัน	
ดีเพนเดนซี	
เรียลไลเซชัน	

1) ความสัมพันธ์แบบแอสโซซิเอชัน เป็นความสัมพันธ์ระหว่างคลาสสองคลาส ซึ่งมีอินสแตนซ์ของคลาสที่ได้ทำการส่งสารระหว่างกัน
2) ความสัมพันธ์แบบเจเนอรัลไลเซชัน เป็นความสัมพันธ์ระหว่างคลาสที่มีลักษณะทั่วไปและคลาสที่มีลักษณะเฉพาะ โดยคลาสที่มีลักษณะเฉพาะใช้โครงสร้างและพฤติกรรมร่วมกับคลาสที่มีลักษณะทั่วไป

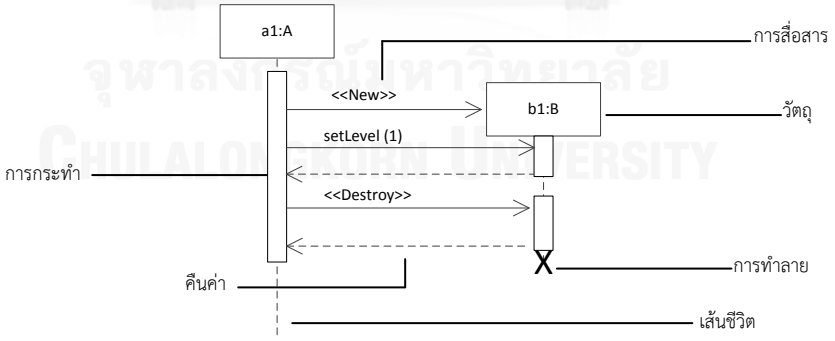
3) ความสัมพันธ์แบบแอกกรีเกชันและคอมโพซิชัน ความสัมพันธ์แบบแอกกรีเกชันเป็นความสัมพันธ์ของคลาสที่เป็นส่วนหนึ่งของอีกคลาส ส่วนความสัมพันธ์แบบคอมโพซิชันจะมีการแสดงความเป็นเจ้าของสูงกว่าความสัมพันธ์แบบแอกกรีเกชันคือ เมื่อลบคลาสที่ใหญ่กว่าคลาสที่เล็กกว่าจะถูกลบด้วย แต่ความสัมพันธ์แบบแอกกรีเกชันนั้นคลาสที่เล็กกว่าจะไม่ถูกลบไปด้วย

4) ความสัมพันธ์แบบดีเพนเดนซี เป็นความสัมพันธ์ที่ใช้ในกรณีที่คลาสหนึ่งมีการเปลี่ยนแปลงแล้วมีผลกระทบต่อคลาสอื่น

5) ความสัมพันธ์แบบเรียลไลเซชัน ใช้เชื่อมระหว่างส่วนส่วนต่อประสานและคลาสหรือส่วนประกอบ

แผนภาพลำดับ [18] เป็นแผนภาพแสดงพฤติกรรม ลำดับการโต้ตอบกันระหว่างวัตถุของเค้าโครงเรื่อง (Scenario) ของซอฟต์แวร์ ส่วนประกอบของแผนภาพลำดับ ประกอบด้วยส่วนต่างๆ ดังนี้

1) วัตถุ แสดงถึงวัตถุที่ถูกสร้างขึ้นจากคลาส
2) การสื่อสาร แสดงการสื่อสารหรือโต้ตอบกันระหว่างวัตถุ
3) เส้นชีวิต แสดงช่วงระยะเวลาการมีชีวิตอยู่ของวัตถุ
4) การกระทำ แสดงช่วงระยะเวลาที่วัตถุกำลังดำเนินการอยู่
5) คืนค่า แสดงการย้อนกลับไปยังวัตถุเดิม
6) การทำลาย แสดงจุดสิ้นสุดของเส้นชีวิต โดยได้รับสารเป็นการลบวัตถุนั้น



รูปที่ 5 ตัวอย่างแผนภาพลำดับ

2.1.7 ตัววัดแผนภาพ (Diagram metrics)

จากการศึกษาและรวบรวมตัววัดแผนภาพคลาสและแผนภาพลำดับจากงานวิจัยที่ผ่านมา [5, 6, 7] ซึ่งสามารถจำแนกออกตามประเภทของตัววัดแผนภาพเชิงโครงสร้างซึ่งวัดได้จากแผนภาพคลาสและตัววัดแผนภาพเชิงพฤติกรรมซึ่งวัดได้จากแผนภาพ ได้ดังนี้

ตารางที่ 4 ตัววัดแผนภาพ

ตัววัดแผนภาพ	**ประเภทของตัววัด**	**คำอธิบาย**
NOC	เชิงโครงสร้าง	ตัววัดจำนวนคลาสลูกของคลาสที่สนใจ
NOP	เชิงโครงสร้าง	ตัววัดจำนวนคลาสแม่ของคลาสที่สนใจ
OpsInh	เชิงโครงสร้าง	ตัววัดจำนวนของโอเปอร์เรเตอร์หรือเมธอดที่รับสืบทอดมาจากคลาสอื่น
AttrInh	เชิงโครงสร้าง	ตัววัดจำนวนของแอททริบิวต์ที่รับสืบทอดมาจากคลาสอื่น
NumDesc	เชิงโครงสร้าง	ตัววัดจำนวนคลาสลูกรวมไปถึงคลาสที่สืบทอดจากคลาสลูกและต่อไป
NumAnc	เชิงโครงสร้าง	ตัววัดจำนวนคลาสแม่และคลาสที่คลาสแม่สืบทอดมา
DIT	เชิงโครงสร้าง	ตัววัดระดับความลึกของคลาสที่สนใจในแผนผังต้นไม้การสืบทอด
CLD	เชิงโครงสร้าง	ตัววัดระดับของความลึกจากคลาสที่สนใจจนถึงระดับที่ลึกที่สุดในแผนผังต้นไม้การสืบทอด
IFImpl	เชิงโครงสร้าง	ตัววัดจำนวนของอินเตอร์เฟสที่คลาสที่สนใจนำมาใช้ในการสร้างคลาส
Assoc_out	เชิงโครงสร้าง	ตัววัดจำนวนความสัมพันธ์ขององค์ประกอบของคลาสที่สนใจกับภายนอกคลาส(ความสัมพันธ์ขาออก)
Assoc_In	เชิงโครงสร้าง	ตัววัดจำนวนความสัมพันธ์ขององค์ประกอบของคลาสที่สนใจกับภายในคลาส(ความสัมพันธ์ขาเข้า)
NumAss_Provider	เชิงโครงสร้าง	ตัววัดจำนวนความสัมพันธ์แอสโซซิเอชันของคลาสที่สนใจในแผนภาพคลาส เมื่อคลาสที่สนใจเป็นผู้ถูกใช้งานจากคลาสอื่น
EC_Par	เชิงโครงสร้าง	ตัววัดจำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นพารามิเตอร์ในคลาสอื่นและคลาสตัวเอง
IC_Par	เชิงโครงสร้าง	ตัววัดจำนวนของพารามิเตอร์ในคลาสที่สนใจที่มีชนิดเป็นคลาสอื่นๆ
EC_Attr	เชิงโครงสร้าง	ตัววัดจำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นแอททริบิวต์ในคลาสอื่นและคลาสตัวเอง
IC_Attr	เชิงโครงสร้าง	ตัววัดจำนวนของแอททริบิวต์ในคลาสที่สนใจที่มีชนิดเป็นคลาสอื่นๆ

ตารางที่ 4 ตัววัดแผนภาพ (ต่อ)

ตัววัดแผนภาพ	ประเภทของตัววัด	คำอธิบาย
MsgSent	เชิงพฤติกรรม	ตัววัดจำนวนครั้งในการส่งสารไปยังคลาสอื่นๆ
MsgRecv	เชิงพฤติกรรม	ตัววัดจำนวนครั้งที่ได้รับสารจากคลาสอื่นๆ
H	เชิงโครงสร้าง	ตัววัดจำนวนความสัมพันธ์ภายในต่อคลาส
ConnComp	เชิงโครงสร้าง	ตัววัดจำนวนการเชื่อมต่อขององค์ประกอบ
NumOps	เชิงโครงสร้าง	ตัววัดจำนวนของโอเปอร์เรเตอร์หรือเมธอดของคลาส
NumPubOps	เชิงโครงสร้าง	ตัววัดจำนวนของโอเปอร์เรเตอร์หรือเมธอดของคลาสที่มีขอบเขตการเข้าถึงเป็นแบบพับลิค
NumAttr	เชิงโครงสร้าง	ตัววัดจำนวนแอททริบิวต์ของคลาส
Nesting	เชิงโครงสร้าง	ตัววัดระดับความลึกคลาสภายในคลาสอื่น
Connectors	เชิงโครงสร้าง	ตัววัดจำนวนการเชื่อมต่อของคลาสที่สนใจ
MsgSelf	เชิงพฤติกรรม	ตัววัดจำนวนครั้งที่คลาสที่สนใจส่งสารไปยังตนเอง

2.1.8 การวิเคราะห์ส่วนประกอบหลัก (Principal Component Analysis)

การวิเคราะห์องค์ประกอบ(Factor Analysis) คือการอธิบายเมตริกซ์สหสัมพันธ์ (Correlation Matrix) ให้ง่ายขึ้น โดยการอธิบายในรูปของตัวแปรที่ลดลง ทำเมตริกซ์ที่ซับซ้อนให้ง่ายขึ้นโดยการลดตัวแปรในเมตริกซ์ ซึ่ง Spearman ได้ให้พื้นฐานการคำนวณอย่างง่ายไว้เมื่อ ค.ศ. 1904 และมีการใช้อย่างแพร่หลายจนกระทั่งมีโปรแกรมการคำนวณโดยคอมพิวเตอร์ และหลายวิธีการของการวิเคราะห์องค์ประกอบได้ใช้คอมพิวเตอร์ช่วยเช่นกัน

การวิเคราะห์ส่วนประกอบหลักเป็นหนึ่งวิธีการในการสกัดส่วนประกอบเพื่อลดจำนวนตัวแปร [17, 18] โดยทำการค้นหาจำนวนองค์ประกอบที่มีความสามารถเพียงพอในการอธิบายความสัมพันธ์ระหว่างตัวแปร ซึ่งกฎที่ดีที่สุดสำหรับการกำหนดจำนวนขององค์ประกอบคือ ค่าไอเกนมากกว่า 1 ค่าไอเกนเป็นค่าที่บ่งบอกถึงความสามารถขององค์ประกอบว่าจะอธิบายความแปรปรวนของกลุ่มตัวอย่างได้มากน้อยเพียงใด ซึ่งถ้าองค์ประกอบนั้นอธิบายความแปรปรวนของกลุ่มตัวอย่างได้น้อยกว่า 1 แล้วก็ไม่มีประโยชน์ที่จะนำองค์ประกอบนั้นมาใช้

ในที่นี้จะแสดงตัวอย่างการวิเคราะห์ส่วนประกอบหลักโดยจะใช้ค่าของตัววัดทั้งหมด 8 ตัววัด จากตัวอย่างโปรแกรมที่มีจำนวนคลาส 29 คลาส โดยจะใช้เครื่องมือในการวิเคราะห์ข้อมูลตัวอย่างดังกล่าว ซึ่งจะได้ผลลัพธ์ดังในตารางที่ 5 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลักนั้นประกอบด้วยค่าต่างๆ ซึ่งจะอธิบายรายละเอียดความหมายของแต่ละค่าในตารางที่ 6

ตารางที่ 5 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลัก

Total Variance Explained						
Component	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	**4.519**	**56.491**	**56.491**	**.519**	**56.491**	**56.491**
2	**1.779**	**22.241**	**78.732**	**.779**	**22.241**	**78.732**
3	.966	12.077	90.809			
4	.540	6.756	97.566			
5	.139	1.735	99.301			
6	.044	.548	99.849			
7	.012	.151	100.000			
8	-2.622E-018	-3.278E-017	100.000			

ตารางที่ 6 ความหมายของผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลัก

ผลลัพธ์	**ความหมาย**
1	Component หมายถึง ส่วนประกอบหรือปัจจัยโดยทั่วไปจะสกัดให้มีจำนวนปัจจัยเท่ากับจำนวนตัวแปร ในตัวอย่างนี้มี 8 ตัววัดจึงมี 8 ปัจจัยหรือ 8 ส่วนประกอบ
2	Eigenvalues หมายถึง ค่าไอเกนหรือค่าความผันแปรคือความแปรปรวนทั้งหมดในตัวัดเดิมที่สามารถอธิบายได้โดยองค์ประกอบหรือค่าไอเกนคือผลบวกค่าของสัมประสิทธิ์สหสัมพันธ์ยกกำลังสองของแต่ละตัววัดในองค์ประกอบหนึ่งๆ ดังนั้นจะไม่พิจารณาองค์ประกอบ ที่มีค่าไอเกนน้อยกว่า 1 จะพบว่ามีเพียงองค์ประกอบหรือส่วนประกอบที่ 1 และ 2 เท่านั้นที่มีค่าไอเกนมากกว่า 1
	% of Variance หมายถึง เปอร์เซ็นต์ที่แต่ละองค์ประกอบสามารถอธิบายความผันแปรได้ ซึ่งมีความผันแปรทั้งหมด = 8 เช่น - % of Variance ขององค์ประกอบที่ 1 =(4.519/8)*100 = 56.491 % หมายถึงองค์ประกอบสามารถอธิบายความผันแปรทั้งหมดได้ 56.491 % - % of Variance ขององค์ประกอบที่ 2 =(1.779/8)*100 = 22.241 % หมายถึงองค์ประกอบสามารถอธิบายความผันแปรทั้งหมดได้ 22.241 %
	Comulative% หมายถึง ผลบวกสะสมของ % of Variance Comulative % ของ 2 องค์ประกอบแรก หมายถึงองค์ประกอบที่ 1-2 อธิบายค่าความแปรปรวนทั้ง 8 ตัวได้ 56.491+22.241 = 78.732
3	วิธีการวิเคราะห์ส่วนประกอบหลักค่าไอเกนเริ่มต้นในข้อ 2 และค่า Extraction Sums of Squared Loading จะเท่ากัน และจะแสดงเฉพาะองค์ประกอบที่มีค่าไอเกนมากกว่า 1

ตารางที่ 7 เปรียบเทียบผลเมื่อไม่มีการหมุนแกนปัจจัยและเมื่อมีการหมุนแกนปัจจัยโดยวิธี Varimax

Component Matrix[a]

	Component	
	1	2
IFImpl	**.899**	
IC_Attr	**.895**	**-.360**
NumAttr	**.875**	**-.400**
NumOps	**.830**	**-.446**
NumPubOps	**.809**	
NumAnc	**.629**	**.739**
DIT	**.629**	**.739**
Nesting		**.398**

Rotated Component Matrix[a]

	Component	
	1	2
NumAttr	**.958**	
IC_A tr	**.954**	
NumOps	**.942**	
IFImpl	**.782**	**.444**
NumPubOps	**.597**	**.580**
NumAnc		**.956**
DIT		**.956**
Nesting	**-.282**	**.295**

จากตารางที่ 5 เป็นการสกัดปัจจัยโดยวิธีการวิเคราะห์ส่วนประกอบหลักมีรายละเอียดซึ่งพบว่าควรมีองค์ประกอบเพียง 2 องค์ประกอบเนื่องจากเฉพาะ 2 องค์ประกอบแรกเท่านั้นที่มีค่าไอเกนมากกว่า 1 โดยองค์ประกอบที่สำคัญที่สุดคือองค์ประกอบที่ 1 เนื่องจากอธิบายหรือดึงความแปรปรวนของข้อมูลได้มากที่สุด ในตัวอย่างนี้สามารถอธิบายได้ถึง 56.491 ส่วนองค์ประกอบที่ 2 จะสำคัญรองลงมา

จากตารางที่ 7 สามารถอธิบายได้ว่าเมื่อยังไม่มีการหมุนแกนปัจจัยค่าสัมประสิทธิ์สหสัมพันธ์ (Rotated Component Matrix) ระหว่างตัววัด IFmpl กับองค์ประกอบที่ 1 มีค่าเท่ากับ 0.899 ซึ่งในองค์ประกอบที่ 2 ไม่แสดงค่าเนื่องจากมีค่าสัมประสิทธิ์สหสัมพันธ์น้อยกว่า จึงสามารถสรุปได้ว่าตัววัด IFmpl อยู่ในองค์ประกอบที่ 1 และเมื่อมีการหมุนแกนปัจจัยโดยวิธี Varimax จะพบว่าค่าสัมประสิทธิ์สหสัมพันธ์เปลี่ยนแปลงไปซึ่งสาเหตุที่ต้องทำการหมุนแกนปัจจัยก็เพื่อให้สามารถจัดตัววัดได้ ถ้าค่าสัมประสิทธิ์สหสัมพันธ์ในองค์ประกอบต่างๆแตกต่างกันไม่ชัดเจน

การพิจารณาว่าตัววัดใดควรอยู่ในองค์ประกอบ (Component) ใดจะพิจารณาจากค่าสัมประสิทธิ์สหสัมพันธ์ ถ้าในองค์ประกอบใดมีค่าสัมประสิทธิ์สหสัมพันธ์ของตัววัดนั้นๆมีค่ามากคือค่าเข้าสู่ +1 หรือ -1 และในองค์ประกอบอื่นๆมีค่าสัมประสิทธิ์สหสัมพันธ์ต่ำคือเข้าสู่ 0 จึงจัดตัววัดนั้นให้อยู่ในองค์ประกอบที่มีค่าสัมประสิทธิ์สหสัมพันธ์สูง ซึ่งจากตารางที่ 7 นั้นสามารถจัดตัววัดได้ตามองค์ประกอบคือองค์ประกอบที่ 1 ประกอบด้วยตัววัด 5 ตัววัด ได้แก่ NumAttr, IC_Attr, NumOps, IFlmpl, NumPubOps และองค์ประกอบที่ 2 ประกอบด้วยตัววัด 3 ตัววัด ได้แก่ NumAnc, DIT,

Nesting ซึ่งสามารถสรุปได้ว่าตัววัดแผนภาพที่นำมาสร้างโมเดลมีทั้งหมด 5 ตัววัดคือ NumAttr, IC_Attr, NumOps, IFlmpl, NumPubOps

งานวิจัยนี้จะใช้การวิเคราะห์ส่วนประกอบหลักในการสกัดตัววัดแผนภาพจากทั้งหมด 27 ตัววัด (กำหนดไว้ในตารางที่ 4 หน้า 17) เพื่อให้ได้ตัววัดที่สามารถใช้อธิบายความเสถียรก่อนจะนำมาสร้างโมเดล แล้วนำตัววัดที่ได้ไปวิเคราะห์ความถดถอยเชิงซ้อนเพื่อให้ได้โมเดลในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ได้ศึกษางานวิจัยที่เกี่ยวข้องแบ่งออกเป็น 2 กลุ่ม คือ กลุ่มที่ 1 เป็นงานวิจัยที่เกี่ยวข้องกับความเสถียรของซอฟต์แวร์ และกลุ่มที่ 2 เป็นงานวิจัยที่เกี่ยวข้องกับการประเมินค่าในการพัฒนาแบบอินคริเมนทัล ซึ่งจุดประสงค์หลักของการศึกษางานวิจัยที่เกี่ยวข้อง เพื่อศึกษาแนวทางและวิธีคิดต่างๆ ของงานวิจัยก่อนหน้า เพื่อนำมาปรับใช้ในการทำงานวิจัยนี้ให้มีประสิทธิภาพยิ่งขึ้น โดยจะแสดงรายละเอียดงานวิจัยที่เกี่ยวข้องได้ดังต่อไปนี้

2.2.1 งานวิจัยที่เกี่ยวข้องกับความเสถียรของซอฟต์แวร์

งานวิจัยกลุ่มนี้เป็นการนำเสนอตัววัด แนวคิด และวิธีการในการสร้างโมเดล ซึ่งมีรายละเอียดดังนี้

2.2.1.1 งานวิจัยที่นำเสนอการคำนวณความเสถียรเชิงตรรกะจากซอร์ซโค้ด

Elish และ Rine ได้นำเสนออัลกอริทึมในการคำนวณความเสถียรเชิงตรรกะของคลาสในโปรแกรมเชิงวัตถุ [4] ซึ่งใช้ตัวชี้วัดที่ได้ถูกนำเสนอไปแล้วในการคำนวณการออกแบบเชิงวัตถุของ Chidamber และ Kemrer มีทั้งหมด 6 ตัวชี้วัดการออกแบบ ที่ทำการคำนวณจากซอร์ซโค้ดที่พัฒนาเสร็จสิ้นแล้ว ได้แก่

1) Weighted Methods per Class (WMC) คือ ตัวชี้วัดที่ใช้ในการวัดความซับซ้อนของคลาส ซึ่งเกิดจาก ผลรวมของความซับซ้อนของเมธอดทั้งหมดภายในคลาส
2) Depth of Inheritance Tree (DIT) คือ ตัวชี้วัดที่ใช้วัดความยาวสูงสุดจากคลาสไปยังรากคลาสของต้นไม้การสืบทอด
3) Number of Children (NOC) คือ ตัวชี้วัดที่ใช้ในการวัดจำนวนคลาสลูก
4) Coupling Between Object classes (CBO) คือ ตัวชี้วัดที่ใช้ในการนับจำนวนคลาสที่มีความขึ้นต่อกันของเมธอดหรือตัวแปรที่อยู่คนละคลาส
5) Response For a Class (RFC) คือ ตัวชี้วัดที่ใช้ในการนับจำนวนเมธอดทั้งหมดที่มีการสร้างไว้ใช้งานภายในคลาส และสามารถเรียกใช้จากคลาสภายนอก

6) Lack of Cohesion in Methods (LCOM) คือ ตัวชี้วัดที่ใช้ในการวัดจำนวนของเมธอดที่มีการเรียกใช้ตัวแปรที่กำหนดไว้ภายในคลาส

ผู้วิจัยกล่าวว่าความเสถียรการออกแบบ ประกอบด้วย โครงสร้างการออกแบบที่จะบ่งบอกถึงความเสถียรเชิงตรรกะ และพฤติกรรมการออกแบบที่จะบ่งบอกถึงความเสถียรเชิงสมรรถนะ โดยผู้วิจัยทำการวิจัยกับ 3 ซอฟต์แวร์ และ 6 ตัวชี้วัดการออกแบบ ซึ่งพบว่าตัวชี้วัดที่ดีของความเสถียรเชิงตรรกะคือ CBO และ RFC ซึ่งไม่พบความสัมพันธ์ระหว่าง NOC กับความเสถียรเชิงตรรกะของคลาส

2.2.1.2 งานวิจัยที่นำเสนอการวัดความเสถียรของคลาสเชิงวัตถุ

เป็นงานวิจัยที่เสนอการวัดความเสถียรของคลาสเชิงวัตถุ [7] โดยมีกระบวนการในการวัดแบ่งเป็นขั้นตอนคือ จำแนกปัจจัยที่มีผลต่อความเสถียรของคลาสซึ่งผู้วิจัยได้ระบุไว้ทั้งหมด 8 ข้อ จากนั้นทำการหาความเป็นไปได้สูงสุดของความเปลี่ยนแปลงของคลาสในแต่ละปัจจัยที่กำหนด แล้วจึงคำนวณค่าความเสถียรตามสูตรการคำนวณที่ผู้วิจัยได้พัฒนาขึ้น ซึ่งงานวิจัยนี้ทำให้ทราบถึงวิธีการนิยามหรือกำหนดปัจจัยที่จะนำไปสร้างโมเดลการประมาณค่าความเสถียร

2.2.1.3 งานวิจัยที่นำเสนอโมเดลการคำนวณค่าความเสถียรเชิงตรรกะของการออกแบบโปรแกรมเชิงโครงสร้าง

เป็นงานวิจัยที่นำเสนอโมเดลการคำนวณค่าความเสถียรเชิงตรรกะของการออกแบบโปรแกรมเชิงโครงสร้าง [2] โดยจะพิจารณาจากแผนผังโครงสร้างของมอดูล คำนวณจากสมมติฐานผลกระทบที่อาจเกิดขึ้นจากการปรับเปลี่ยนมอดูลหรือผลรวมของสมมติฐานที่กระทำโดยมอดูลอื่นๆ กับมอดูลที่ทำการวัด ซึ่งพิจารณาจากการเชื่อมต่อของมอดูล ลำดับชั้นมอดูลของโปรแกรม และส่วนประกอบของมอดูล โดยแสดงดังสมการที่ 3 หน้า 9 พร้อมตัวอย่างการคำนวณ ซึ่งเป็นแนวคิดที่จะนำมาสร้างโมเดลในการคำนวณค่าการออกแบบของโปรแกรมเชิงวัตถุในการพัฒนาแบบอินคริเมนทัล

2.2.1.4 กลุ่มงานวิจัยที่ทำนำเสนอการสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาส

กลุ่มงานวิจัยที่ทำการประมาณค่าความเสถียรเชิงตรรกะโดยทำการประมาณค่าในขั้นตอนการออกแบบ จากแผนภาพคลาสและแผนภาพลำดับ โดยเสนอโมเดลการประมาณค่าความเสถียร [4] และเทคนิคต่างๆได้แก่ การอุปมา (Analogy) [5] ข่ายงานประสาทเทียม (ANN) [6] ในการนำตัวชี้วัดแผนภาพมาใช้ในการประมาณค่า โดยทั้งหมดเป็นการนำซอร์ซโค้ดที่ได้จากการพัฒนาเรียบร้อยแล้วมาทำการสร้างย้อนกลับ (Reverse) เป็นแผนภาพคลาสและแผนภาพลำดับ เพื่อที่จะสามารถทำการเปรียบเทียบผลที่ได้ของค่าประมาณ และค่าจริงที่วัดได้จากซอร์ซโค้ดของโปรแกรมนั้นๆ ซึ่งโมเดลการประมาณค่าความเสถียรที่พัฒนาขึ้นนั้นได้มีการแบ่งการทดลองให้เหมาะสมกับโปรแกรม 3 ประเภท คือ โปรแกรมด้านการคำนวณ โปรแกรมด้านการจัดการข้อความ โปรแกรมด้านการจัดการภาพ เพราะแต่ละประเภทมีรูปแบบการทำงานที่แตกต่างกัน ซึ่งการใช้ตัววัดแผนภาพ การกำหนดความ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

เปลี่ยนแปลงและผลกระทบที่เกิดขึ้นกับคลาสของงานวิจัยกลุ่มนี้จะนำไปใช้เป็นแนวคิดในการสร้างโมเดลการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์จากแผนภาพคลาสและแผนภาพลำดับ

2.2.2 งานวิจัยที่เกี่ยวข้องกับการประเมินค่าในการพัฒนาแบบอินคริเมนทัล

งานวิจัยกลุ่มนี้เป็นการนำเสนอการประมาณค่าต่างๆ ที่บ่งบอกถึงคุณภาพของซอฟต์แวร์ ที่มีการพัฒนาแบบอินคริเมนทัล ซึ่งมีรายละเอียดดังนี้

2.2.2.1 ความน่าเชื่อถือ (Reliability)

ความน่าเชื่อถือ (Reliability) [15] แนวคิดของงานวิจัยคือการประเมินค่าความน่าเชื่อถือโดยใช้วิธีการในการประมาณค่าความน่าเชื่อถือซึ่งมีความเกี่ยวข้องกับการวัดในกระบวนการการพัฒนาแบบอินคริเมนทัล หลักการคือใช้ NHPP-based SRM (Non-Homogeneous Poisson Process based Software Reliability Model) เนื่องจากเป็นวิธีที่รู้จักกันดีและได้รับความนิยมเพื่อใช้ในการประมาณค่าความน่าเชื่อถือของซอฟต์แวร์ ซึ่งงานวิจัยนี้มุ่งเน้นไปที่การตรวจจับความผิดพลาดของ data ในขั้นตอนการทดสอบที่เพิ่มขึ้น ซอฟต์แวร์ที่นำมาใช้เป็นซอฟต์แวร์ที่ใช้ในการทดสอบประกอบไปด้วย การทดสอบมอดูล การทดสอบการรวม และการทดสอบระบบ ซึ่งงานวิจัยนี้สนใจเพียงการทดสอบระบบ เพราะเป็นจุดที่มีการปรับปรุงความน่าเชื่อถือ

2.2.2.1 ความพยายาม (Effort)

ความพยายาม(Effort) [13] งานวิจัยนี้นำเสนอวิธีการใหม่ในการประมาณค่าความพยายามในสภาวะการพัฒนาแบบเอจายล์ (Agile) และแบบวนซ้ำ (Iterative) โดยทำการเปรียบเทียบผลจากโมเดลการทำนายของ 2 โมเดล คือ รูปแบบการทำนายแบบโกลบอล (Global) และ รูปแบบการทำนายแบบอินคริเมนทัลซึ่งเป็นความคิดหลักที่ผู้วิจัยได้สร้างและนำเสนอซึ่งสามารถประมาณค่าได้ดีกว่ารูปแบบการทำนายแบบโกลบอล เพราะมีการปรับปรุงในทุกๆ ครั้งของการพัฒนาทวนซ้ำ ซึ่งแนวคิดในการสร้างโมเดลการทำนายแบบอินคริเมนทัลนี้จะนำไปใช้ในการปรับโมเดลการประเมินค่าความเสถียรของการออกแบบจากโมเดลเชิงโครงสร้างเป็นโมเดลเชิงวัตถุ

2.2.3 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ

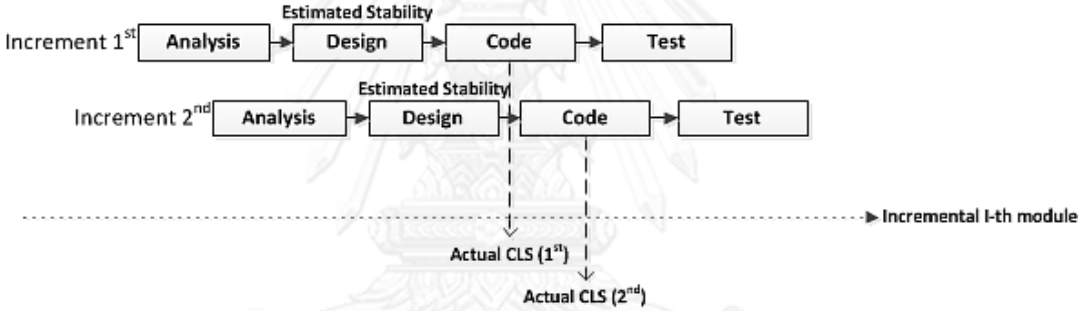
จากการศึกษางานวิจัยที่เกี่ยวข้องทั้งหมดที่กล่าวถึงข้างต้น ทุกงานวิจัยเป็นการนำเสนอโมเดล แต่จะมีส่วนที่แตกต่างกันคือวิธีการและเครื่องมือที่ใช้ในการพัฒนา จึงได้ทำการสรุปวิธีการและเครื่องมือที่ใช้ของแต่ละงานวิจัยที่เกี่ยวข้อง พร้อมกับเปรียบเทียบกับงานวิจัยที่นำเสนอ ดังแสดงในตารางที่ 8

ตารางที่ 8 เปรียบเทียบงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ

งานวิจัย	ขั้นตอน	วัดค่าจาก	ตัวแปรอิสระ	เทคนิคที่ใช้
งานวิจัยที่เกี่ยวข้องกับความเสถียรของซอฟต์แวร์				
2.2.1.1	การบำรุงรักษา	ซอร์ซโค้ด	ตัววัดการออกแบบเชิงวัตถุ 6 ตัววัด	อัลกอริทึมคำนวณความเสถียรเชิงตรรกะของคลาส
2.2.1.2	การบำรุงรักษา	ซอร์ซโค้ด	ปัจจัยที่มีผลต่อความเสถียร 8 ปัจจัย	ความเป็นไปได้สูงสุด
2.2.1.3	การออกแบบ	แผนผังโครงสร้างของมอดูล	ส่วนประกอบและลำดับชั้นของการออกแบบโปรแกรมเชิงโครงสร้าง	การนับจำนวนการเชื่อมต่อของมอดูล
2.2.1.4	การออกแบบ	แผนภาพคลาสและแผนภาพลำดับ	ตัววัดแผนภาพ 21ตัววัด	การวิเคราะห์ความถดถอยเชิงซ้อน
	การออกแบบ	แผนภาพคลาสและแผนภาพลำดับ	ตัววัดแผนภาพ 21 ตัววัด	การอุปมา , การวิเคราะห์ความถดถอยเชิงซ้อน
	การออกแบบ	แผนภาพคลาสและแผนภาพลำดับ	ตัววัดแผนภาพ 30 ตัววัด	ข่ายงานประสาทเทียม
งานวิจัยที่เกี่ยวข้องกับการประเมินค่าในการพัฒนาซอฟต์แวร์แบบอินคริเมนทัล				
2.2.2.1	การทดสอบระบบ	ผลลัพธ์ของข้อมูล	ความผิดพลาดของข้อมูล	การวิเคราะห์ทางสถิติสำหรับตัวแบบความเชื่อถือได้ของซอฟต์แวร์
2.2.2.2	การออกแบบ	ข้อมูลที่เกี่ยวข้องในอดีต	ปัจจัยต่างๆ เช่น ขนาด, การออกแบบ, คุณภาพและปริมาณของทรัพยากร, ปัจจัยทางเทคโนโลยี	ข่ายงานประสาทเทียม , การวิเคราะห์ความถดถอยเชิงซ้อน
งานวิจัยนี้	การออกแบบ	แผนภาพคลาสและแผนภาพลำดับ	ตัววัดแผนภาพที่กำหนดจากผลกระทบเชิงตรรกะของการออกแบบ, ตัววัดแผนภาพ 27 ตัววัด	การวิเคราะห์ความถดถอยเชิงซ้อน, การวิเคราะห์ส่วนประกอบหลัก, การวิเคราะห์การผลกระทบเชิงตรรกะของการออกแบบ

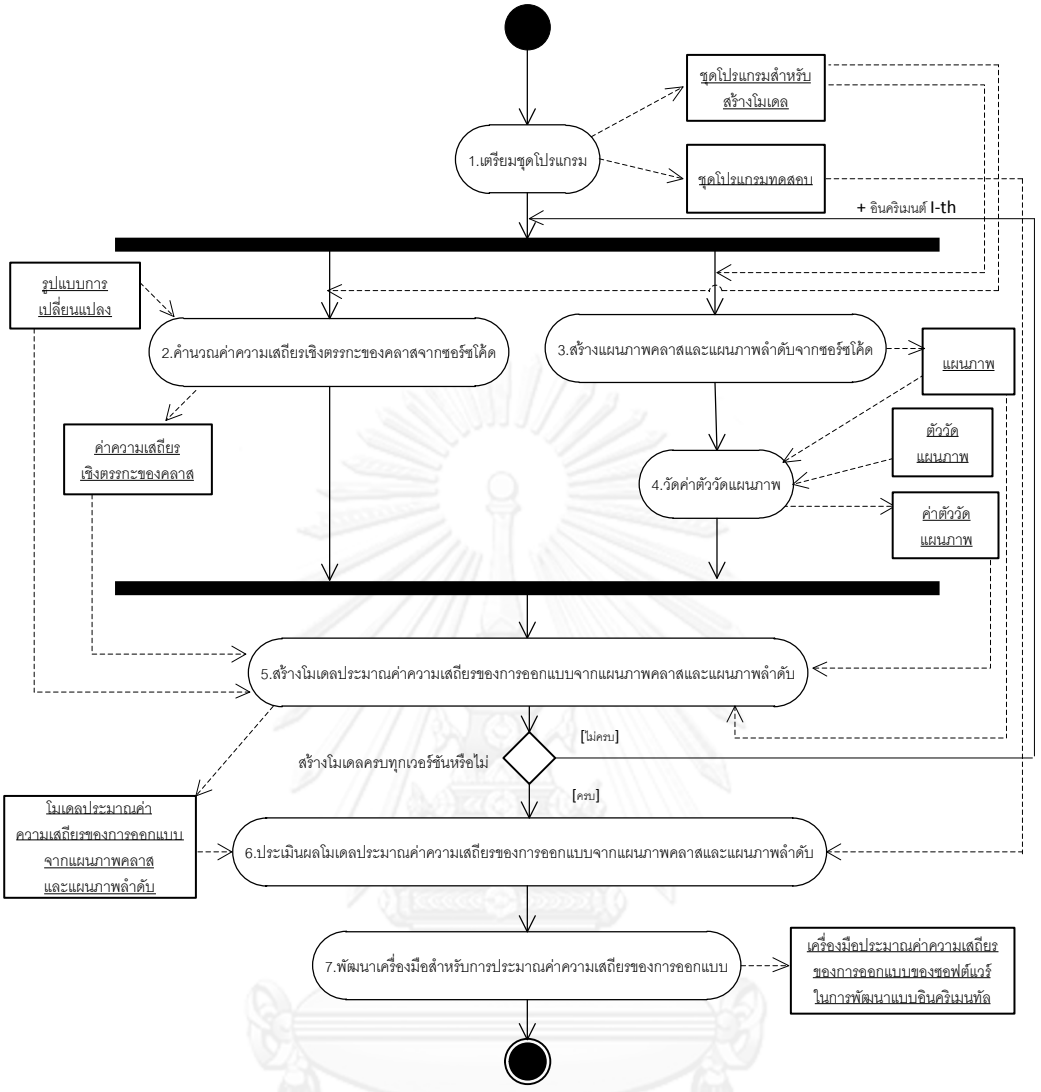
บทที่ 3
การสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาส ในการพัฒนาแบบอินคริเมนทัล

ความเสถียรเชิงตรรกะของคลาสคือความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงของซอฟต์แวร์ โดยสามารถวัดได้เมื่อการพัฒนาซอฟต์แวร์เสร็จสิ้นแล้ว ซึ่งการพัฒนาแบบอินคริเมนทัลนั้นจะมีการเปลี่ยนแปลงเกิดขึ้นกับซอฟต์แวร์เมื่อมีการพัฒนาในอินคริเมนต์ถัดไปทำให้เกิดโครงสร้างการออกแบบใหม่ ซึ่งอาจจะมีคลาสของอินคริเมนต์ก่อนหน้าจะได้รับผลกระทบจากการเปลี่ยนแปลงนี้ และอาจส่งผลกระทบต่อความเสถียรเชิงตรรกะของคลาส หากค่าความเสถียรเชิงตรรกะของคลาสลดลงก็เป็นส่วนที่จะนำไปพิจารณาในการปรับปรุงแก้ไขโครงสร้างการออกแบบใหม่ ซึ่งถ้าสามารถคาดการณ์ล่วงหน้าในขั้นตอนของการออกแบบได้ก็จะทำให้สามารถลดความซับซ้อนและยุ่งยากในขั้นตอนของการบำรุงรักษาและไม่ต้องรอพิจารณาปรับปรุงโมเดลของการอ ออกแบบในภายหลังการพัฒนาเสร็จสิ้น



รูปที่ 6 แนวคิดในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล

จากที่กล่าวมาข้างต้นแสดงได้ดังรูปที่ 6 ซึ่งเป็นแนวคิดของงานวิจัยนี้คือการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ที่พัฒนาแบบอินคริเมนทัล ซึ่งแบ่งการพัฒนาออกเป็นอินคริเมนต์ ซึ่งการเปลี่ยนแปลงจะเกิดขึ้นเมื่อมีการเพิ่มการพัฒนาในอินคริเมนต์ถัดไปโครงสร้างการออกแบบก็จะเปลี่ยนไป หากการเปลี่ยนแปลงเหล่านั้นส่งผลกระทบเชิงตรรกะต่อคลาสใดๆ ในอินคริเมนต์ก่อนหน้าก็จะเกิดผลกระทบต่อความเสถียรเชิงตรรกะของคลาสนั้นๆด้วย ซึ่งหากค่าความเสถียรเชิงตรรกะของคลาสลดลงควรมีการแก้ไขโครงสร้างการออกแบบใหม่ การประมาณค่าความเสถียรเชิงตรรกะของคลาสในขั้นตอนของการออกแบบนั้นสามารถนำไปใช้ในการตัดสินใจแก้ไขการออกแบบ (Re-design) ของแต่ละอินคริเมนต์ก่อนที่จะเข้าสู่ขั้นตอนต่อไปของการพัฒนาซอฟต์แวร์จนเสร็จสิ้น ซึ่งจะสามารถคำนวณค่าความเสถียรเชิงตรรกะหรือค่าจริง (Actual CLS) โครงสร้างของการออกแบบที่ใช้ในงานวิจัยนี้จะประกอบด้วย แผนภาพคลาส และแผนภาพลำดับ ซึ่งจะต้องสร้างโมเดลสำหรับการประมาณค่าความเสถียรเชิงตรรกะของคลาส งานวิจัยนี้นำเสนอการสร้างโมเดลสำหรับการประมาณค่าความเสถียรเชิงตรรกะของคลาสโดยอธิบายรายละเอียดภาพรวมของงานวิจัยเป็นขั้นตอนที่แบ่งออกเป็น 7 ขั้นตอนดังรูปที่ 7 ได้ดังนี้



รูปที่ 7 ภาพรวมขั้นตอนของงานวิจัย

ขั้นตอนที่ 1 คือการเตรียมชุดโปรแกรมโอเพ่นซอร์ซ ซึ่งจะได้ผลลัพธ์ของขั้นตอนแบ่งออกเป็น 2 กลุ่ม คือ ชุดโปรแกรมสำหรับสร้างโมเดล และชุดโปรแกรมทดสอบ

ขั้นตอนที่ 2 คือการคำนวณค่าความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ด ซึ่งใช้ซอร์ซโค้ดของชุดโปรแกรมสำหรับสร้างโมเดลในขั้นตอนที่ 1 โดยรูปแบบการเปลี่ยนแปลงที่งานวิจัยได้กำหนดไว้ในตารางที่ 1 ในบทที่ 2 ซึ่งผลลัพธ์ของขั้นตอนนี้คือ ค่าความเสถียรเชิงตรรกะของคลาสทุกคลาสในแต่ละอินคริเมนต์ที่คำนวณได้จากซอร์ซโค้ด

ขั้นตอนที่ 3 คือการสร้างแผนภาพคลาสและแผนภาพลำดับจากซอร์ซโค้ด โดยใช้เครื่องมือ [22] แปลงซอร์ซโค้ดของชุดโปรแกรมสำหรับสร้างโมเดลในขั้นตอนที่ 1 ซึ่งผลลัพธ์ของขั้นตอนนี้จะได้แผนภาพคลาสของแต่ละอินคริเมนต์และแผนภาพลำดับของการทำงานของทุกคลาสในแต่ละเมธอด

ขั้นตอนที่ 4 คือการวัดค่าตัววัดแผนภาพที่ได้จากขั้นตอนที่ 3 โดยใช้เครื่องมือ [23] ในการวัดค่าตัววัดแผนภาพทั้งหมด 27 ตัววัดตามที่งานวิจัยได้ทำการกำหนดไว้ในตารางที่ 5 ในบทที่ 2 ผลลัพธ์ที่ได้ก็คือค่าตัววัดแผนภาพทั้งหมด 27 ตัวของทุกคลาสในแต่ละอินคริเมนต์

ขั้นตอนที่ 5 คือการสร้างโมเดลประมาณค่าความเสถียรของการออกแบบในการพัฒนาแบบอินคริเมนทัล จากแผนภาพคลาสและแผนภาพลำดับ โดยข้อมูลที่ใช้คือค่าความเสถียรเชิงตรรกะของคลาส และค่าตัววัดแผนภาพของทุกคลาสในแต่ละอินคริเมนต์ ในงานวิจัยนี้จะทำการสร้างโมเดลด้วยวิธีการที่แตกต่างกันทั้งหมด 4 วิธี ได้แก่ การวิเคราะห์ความถดถอยเชิงซ้อน (MRA) การวิเคราะห์ส่วนประกอบหลัก (PCA) การวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) และการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล (DLREA(PCA)) แล้วนำผลที่ได้มาเปรียบเทียบ วิเคราะห์ผล และสรุปผล ซึ่งจะอธิบายอย่างละเอียดถึงขั้นตอนและวิธีการในลำดับถัดไป

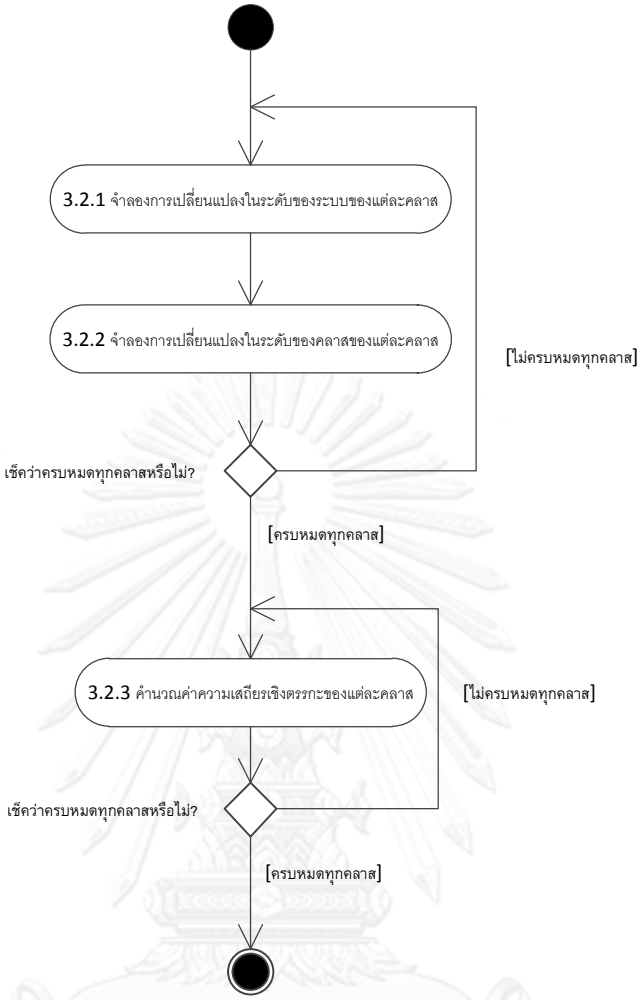
ขั้นตอนที่ 6 คือการประเมินผลโมเดลประมาณค่าความเสถียรของการออกแบบในการพัฒนาแบบอินคริเมนทัล จากแผนภาพคลาสและแผนภาพลำดับ โดยจะใช้โมเดลที่สร้างจากขั้นตอนที่ 5 มาประมาณค่าความเสถียรของชุดโปรแกรมทดสอบ แล้วนำผลของค่าประมาณที่ได้และค่าจริงที่คำนวณได้จากซอร์ซโค้ดมาวิเคราะห์ความถูกต้องแม่นยำว่าสามารถยอมรับโมเดลนั้นๆในการนำมาประมาณค่าความเสถียรเชิงตรรกะของคลาสได้หรือไม่ซึ่งจะนำเสนอในบทที่ 4

ขั้นตอนที่ 7 คือการพัฒนาเครื่องมือสำหรับการประมาณค่าความเสถียรของการออกแบบในการพัฒนาแบบอินคริเมนทัล ซึ่งส่วนของรายละเอียดเกี่ยวกับเครื่องมือนี้จะนำเสนอไว้ในบทที่ 5

จากที่ได้กล่าวมานี้ส่วนสำคัญของงานวิจัยคือ ขั้นตอนที่ 5 คือการสร้างโมเดลประมาณค่าความเสถียรของการออกแบบในการพัฒนาแบบอินคริเมนทัลด้วยวิธีการที่แตกต่างกัน จากแผนภาพคลาสและแผนภาพ ซึ่งสามารถอธิบายถึงลำดับขั้นตอนในการสร้างโมเดลอย่างละเอียด ดังนี้

3.1 ขั้นตอนการเตรียมชุดโปรแกรม

ค้นหาโปรแกรมที่เขียนด้วยภาษาจาวาจากโอเพ่นซอร์ซ [20, 21] โดยพิจารณาเลือกโปรแกรมจากหมายเลขเวอร์ชันที่ระบุไว้ มีโครงสร้าง x.y.z โดยที่ x คือตำแหน่งหมายเลขแทนการเพิ่มฟังก์ชัน เกิดการเปลี่ยนแปลงมาก y คือตำแหน่งหมายเลขแทนการเปลี่ยนแปลงเล็กน้อย เพื่อความสะดวกในการใช้งาน และ z คือตำแหน่งหมายเลขแทนจำนวนการสร้าง จำนวนครั้งการแก้ไขข้อผิดพลาด โดยไม่ทำให้ฟังก์ชันการทำงานเปลี่ยนไป ในการเลือกจะให้ความสำคัญกับตำแหน่งแรกคือ x ที่เปลี่ยนไป เช่น 1.0.0, 2.0.0, 3.0.0 เป็นต้น โดยให้ความสำคัญกับส่วนของฟังก์ชันการทำงานเป็นหลัก ซึ่งเป็นการพัฒนาแบบอินคริเมนทัล เนื่องจากในแต่ละเวอร์ชันจะมีการระบุการเปลี่ยนแปลงและการเพิ่มของฟังก์ชันไว้ เวอร์ชันถัดไปจะมีการเพิ่มฟังก์ชันซึ่งเป็นการรวมกันของอินคริเมนต์ก่อนหน้าและอินคริเมนต์ถัดไป ซึ่งจะได้ผลลัพธ์ของขั้นตอนนี้แบ่งออกเป็นชุดโปรแกรมสำหรับสร้างโมเดล 5 โปรแกรม และชุดโปรแกรมสำหรับประเมินผลโมเดล 15 โปรแกรม รายละเอียดของแต่ละโปรแกรมแสดงในบทที่ 4



รูปที่ 8 ขั้นตอนการคำนวณค่าความเสถียรเชิงตรรกะของคลาส

3.2 ขั้นตอนการคำนวณค่าความเสถียรเชิงตรรกะของคลาส

การคำนวณค่าความเสถียรเชิงตรรกะของคลาสในทุกๆคลาสในแต่ละเวอร์ชัน แสดงขั้นตอนดังรูปที่ 9 โดยจะคำนวณจากกรณีการเปลี่ยนแปลงในระดับของระบบและระดับของคลาสที่สามารถเกิดขึ้นได้ในแต่ละคลาส ซึ่งกำหนดไว้ในตารางที่ 1 จากนั้นเก็บค่าจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมดและจำนวนครั้งที่คลาสจะได้รับผลกระทบจากการเปลี่ยนแปลงนั้นๆ แล้วนำไปคำนวณความเสถียรตามสมการที่ 1 ซึ่งค่าที่ได้จากขั้นตอนนี้จะถูกเก็บไว้ เพื่อนำไปใช้ในการสร้างโมเดลต่อไป

ในขั้นตอนนี้จะใช้เครื่องมือที่ชื่อว่า ChangeImpact ซึ่งผู้พัฒนาคือ นายศุภวัชร์ รังสิยวัฒน์ [4] มาใช้ในการคำนวณค่าความเสถียรเชิงตรรกะของคลาส โดยมีหลักการทำงานจากสมการคำนวณค่าความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ดตามข้อกำหนดดังสมการที่ 1 ในบทที่ 1

ตัวอย่างการคำนวณค่าความเสถียรเชิงตรรกะตามสมการที่ 1 แสดงดังตารางที่ 9 และ 10 โดยใช้การเปลี่ยนแปลงตามที่กำหนดไว้ในตารางที่ 1 ในบทที่ 1 ซึ่งแสดงตัวอย่างเพื่อต้องการแสดง

ขั้นตอนวิธีการคำนวณค่าความเสถียรเชิงตรรกะของคลาสโดยแสดงการเปลี่ยนแปลงเพียงบางส่วนของรูปแบบการเปลี่ยนแปลง

ตารางที่ 9 ตัวอย่างซอร์ซโค้ดที่ใช้คำนวณความเสถียรเชิงตรรกะของคลาสต่อการเปลี่ยนแปลงที่กำหนด

```
public class C1{
public int a1;
private C2 a2;
public int mC1(){
  C3 c = new C3();
  a1 = c.mC3();
  return a1; } }
```

```
public class C2{
public int b1;
public void mC2(){
  C3 c = new C3();
  B1 = c.mC3();
  } }
```

```
public class C3 extends C1{
public int c1;
public int mC3(){
  c1 = 10;
  return c1;
  } }
```

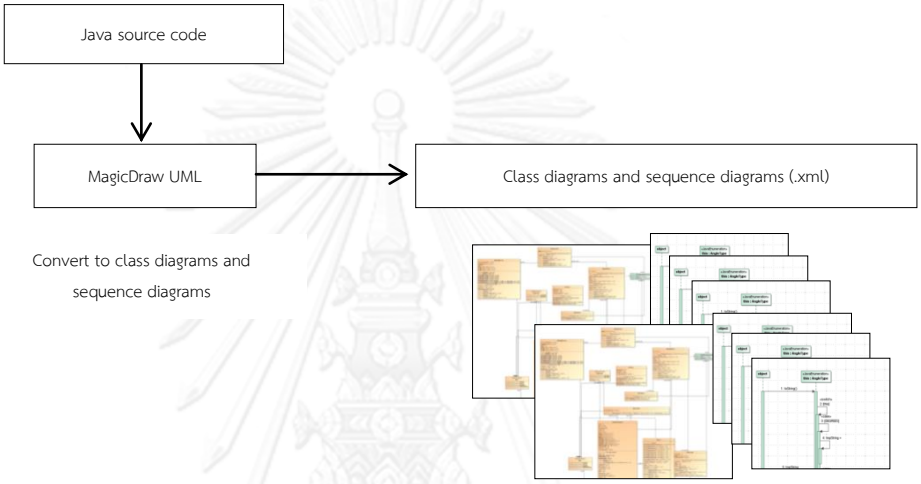
ตารางที่ 10 ตัวอย่างการคำนวณค่าความเสถียรเชิงตรรกะของคลาสต่อการเปลี่ยนแปลงที่กำหนด

การเปลี่ยนแปลงที่กำหนด	Class C1	Class C2	Class C3
ลบคลาส C1	X		X
ลบคลาส C2	X	X	
ลบคลาส C3	X	X	X
เพิ่มความสัมพันธ์แอสโซซิเอชันคลาสของคลาส C1	X		X
ลบแอททริบิวต์ a1	X		
เปลี่ยนขอบเขต a1 เป็นไพรเวต	X		X
เปลี่ยนขอบเขต a1 เป็นโพรเทคต์		X	
เปลี่ยนชื่อเมธอด mC3()	X	X	X
เปลี่ยนเมธอด mC3()	X	X	X
เปลี่ยนขอบเขต mC3() เป็นโพรเทคต์	X	X	
จำนวนครั้งที่ได้รับผลกระทบ	9	6	6
ความเสถียรเชิงตรรกะของคลาส	1-(9/10)=0.1	1-(6/10)=0.4	1-(6/10)=0.4

จากตารางข้างต้น หากค่าที่ได้มีค่าเท่ากับ 0 ถึง 1 (มีความเสถียรมากขึ้นเมื่อค่าเข้าใกล้ 1) มีค่าเท่ากับ 1หมายถึงเสถียร (เป็นค่าความเสถียรที่มากที่สุด) แต่หากค่าที่ได้มีค่าเท่ากับ 0 หมายถึงไม่เสถียร ซึ่งจากตารางที่ 10 ทำให้ทราบค่าความเสถียรเชิงตรรกะของคลาสต่อการเปลี่ยนแปลงที่กำหนดขึ้น 10 การเปลี่ยนแปลงดังตาราง คือ ค่าความเสถียรเชิงตรรกะของ C1 เท่ากับ 0.1 ค่าความเสถียรเชิงตรรกะของ C2 เท่ากับ 0.4 และค่าความเสถียรเชิงตรรกะของ C3 เท่ากับ 0.4

3.3 ขั้นตอนการสร้างแผนภาพคลาสและแผนภาพลำดับ

การสร้างแผนภาพจากการนำซอร์ซโค้ดที่ได้จากขั้นตอน 3.1 มาทำการแปลงกลับเป็นแผนภาพคลาสและแผนภาพลำดับด้วยโปรแกรม MagicDraw UML [22] ดังรูปที่ 10 โดยทำการแปลงกลับเป็นแผนภาพคลาสให้ครบทุกคลาส และแปลงกลับเป็นแผนภาพลำดับให้ครบทุกเมธอดของทุกคลาสให้ครบทุกเวอร์ชันที่มีการเพิ่มอินคริเมนต์ถัดไป แล้วบันทึกข้อมูลแผนภาพของแต่ละอินคริเมนต์ให้อยู่ในรูปแบบไฟล์เอ็กซ์เอ็มแอล (xml) เพื่อการนำไปใช้ในขั้นตอนต่อไป ซึ่งขั้นตอนการใช้เครื่องมือในการสร้างแผนภาพคลาสและแผนภาพลำดับจากซอร์ซโค้ดแสดงไว้ในภาคผนวก ข



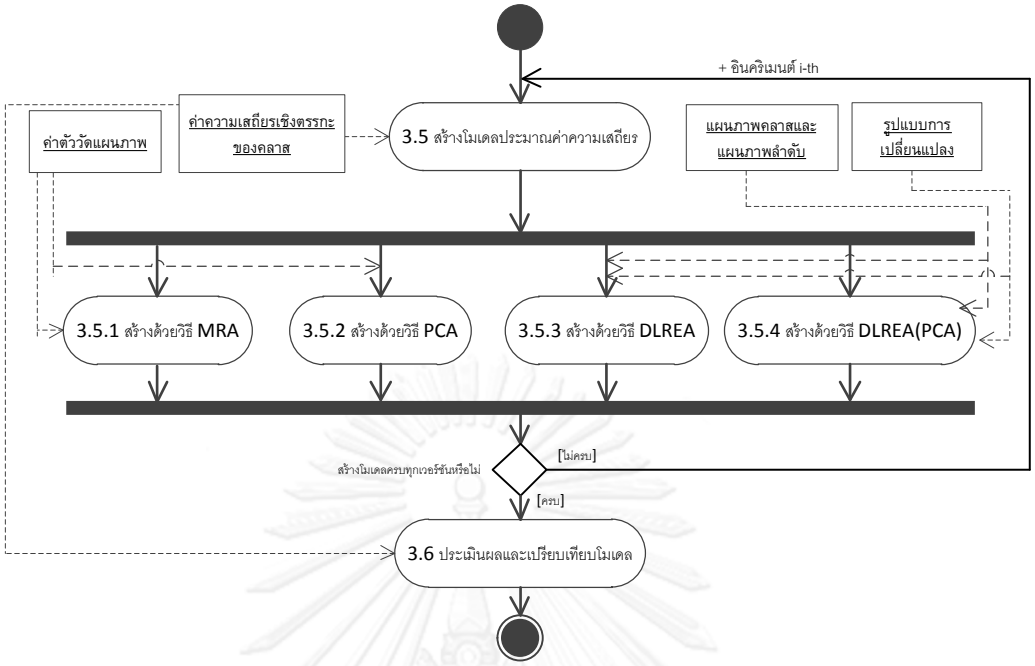
รูปที่ 9 การสร้างแผนภาพคลาสและแผนภาพลำดับ

3.4 ขั้นตอนการวัดค่าตัววัดแผนภาพ

การวัดค่าตัววัดแผนภาพจะใช้โปรแกรม SDMetrics โดยนำเข้าไฟล์แผนภาพที่อยู่ในรูปแบบไฟล์เอ็กซ์เอ็มแอล (xml) ที่ได้จากการแปลงจากซอร์ซโค้ดในขั้นตอนที่ 3.3 โดยโปรแกรมจะประมวลผลและแสดงผลค่าของตัววัดแผนภาพของแต่ละคลาสในรูปแบบของตาราง (csv) ซึ่งจะประกอบด้วยค่าของตัววัดต่างๆ ซึ่งจะต้องเลือกค่าตามตัววัดที่งานวิจัยได้กำหนดทั้งหมด 27 ตัววัด เพื่อนำไปใช้ในขั้นตอนของการสร้างโมเดล คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน และโมเดลที่สร้างด้วยวิธีวิเคราะห์ส่วนประกอบหลัก

3.5 ขั้นตอนการสร้างโมเดล

งานวิจัยนี้นำเสนอโมเดลในการประมาณค่าความเสถียรของการออกแบบ จากแผนภาพคลาสและแผนภาพลำดับในการพัฒนาแบบอินคริเมนทัลทั้งหมด 4 โมเดล คือ โมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) โมเดลที่ 2 สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โมเดลที่ 3 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) และโมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดก่อนจะนำมาสร้างโมเดล (DLREA(PCA))

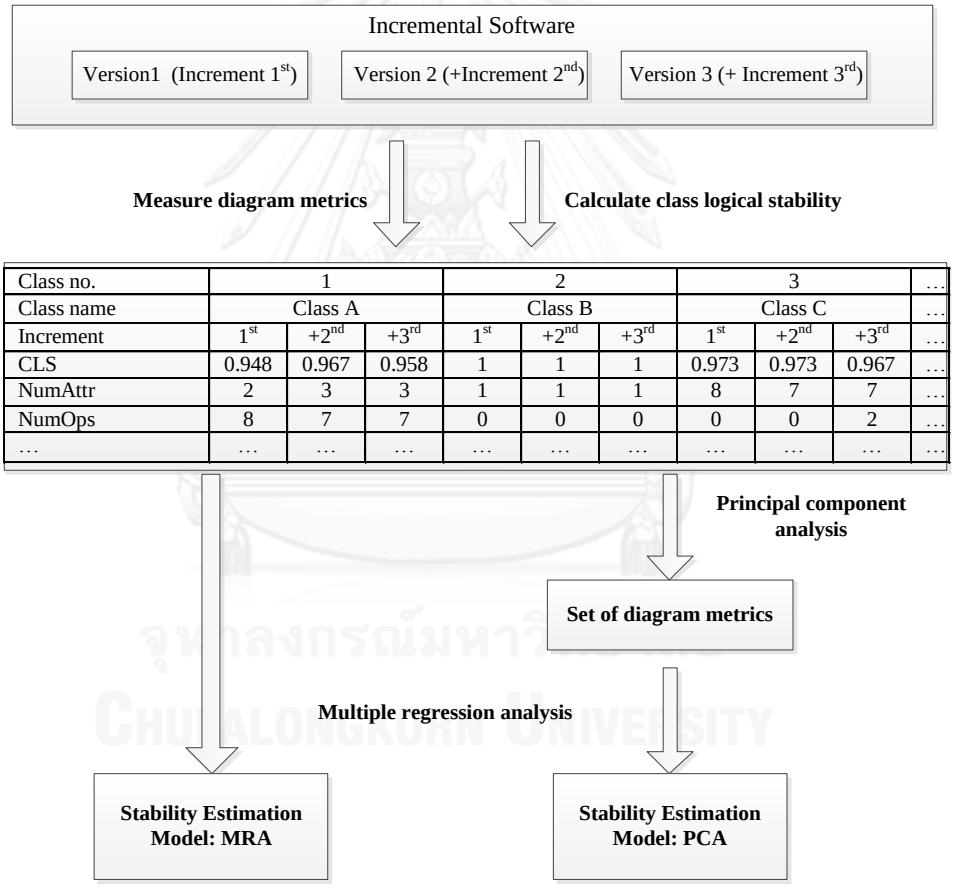


รูปที่ 10 ภาพรวมของกิจกรรมการสร้างโมเดลสำหรับประมาณค่าความเสถียร
ด้วยวิธี MRA, PCA, DLREA และ DLREA(PCA)

จากรูปที่ 10 แสดงภาพรวมของกิจกรรมการสร้างโมเดลสำหรับประมาณค่าความเสถียร ซึ่งแตกต่างกันที่วิธีการในการสร้างโมเดล ข้อมูลที่ใช้ในการสร้างโมเดล รวมทั้งข้อกำหนดและตัววัด โดยโมเดลที่ 1 และโมเดลที่ 2 จะใช้ข้อมูลที่เหมือนกันคือค่าตัววัดแผนภาพ ส่วนโมเดลที่ 3 และ 4 จะใช้แผนภาพคลาสและแผนภาพลำดับในการนำมาวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งจะพิจารณาตามรูปแบบการเปลี่ยนแปลงและผลกระทบที่กำหนดขึ้นจากตารางที่ 1 ในบทที่ 2 จากนั้นเมื่อเสร็จสิ้นการสร้างโมเดลทั้งหมดก็จะต้องประเมินผลโมเดลและเปรียบเทียบโมเดล ด้วยการเปรียบเทียบค่าประมาณและค่าจริงที่ได้จากการคำนวณค่าความเสถียรเชิงตรรกะของคลาส

การสร้างโมเดลในแต่ละวิธีนั้นมีแนวคิดมาจากการศึกษางานวิจัยในอดีตที่โมเดลประมาณค่าความเสถียรนั้นสร้างจากตัววัดที่มีผู้วิจัยในอดีตเป็นผู้กำหนดขึ้นและนำมาสร้างโมเดลด้วยวิธีที่แตกต่างกัน โดยไม่มีการระบุวิธีการในการเลือกตัววัดที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาสก่อนจะนำมาสร้างโมเดล ซึ่งอาจมีผลต่อความแม่นยำและถูกต้องของค่าประมาณที่ได้จากโมเดล ดังนั้นเพื่อการเปรียบเทียบและวิเคราะห์ผลจึงนำตัววัดแผนภาพทั้ง 27 ตัววัดที่รวบรวมจากงานวิจัยในอดีตมาสร้างโมเดลที่ 1 ด้วยการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) ซึ่งเป็นวิธีพื้นฐานในการสร้างโมเดลทั่วไป จากนั้นโมเดลที่ 2 นำการวิเคราะห์ส่วนประกอบหลัก (PCA) มาใช้ในการเลือกตัววัดแผนภาพจาก 27 ตัววัดเพื่อหาตัววัดที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาสก่อนนำมาสร้างโมเดล โดยการวิเคราะห์ความถดถอยเชิงซ้อนแสดงดังรูปที่ 11 ข้อมูลประกอบด้วยค่าความเสถียรเชิงตรรกะของคลาสและค่าของตัววัดที่คำนวณจากไฟล์แผนภาพ (xml) โดยเครื่องมือ [23] และเนื่องจากข้อกำหนดของตัววัดในแต่ละโมเดลประมาณค่าของงานวิจัยที่ผ่านมาเป็นจำนวนองค์ประกอบต่างๆ ในแผนภาพ [4, 5, 6] ซึ่งไม่สอดคล้องกับคำนิยามของความเสถียรซึ่งหมายถึงความต้านทาน

ผลกระทบที่เกิดจากการเปลี่ยนแปลงของซอฟต์แวร์ รวมทั้งพบว่ามีผู้วิจัยในอดีต [2] ได้กำหนดตัววัดที่ใช้สร้างโมเดลขึ้นเอง แต่เป็นโมเดลสำหรับการประมาณค่าความเสถียรเชิงตรรกะของมอดูลโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบโปรแกรมในเชิงโครงสร้าง ในโมเดลที่ 3 จึงนำเสนอการสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) โปรแกรมในเชิงวัตถุ โดยพิจารณาผลกระทบจากการเปลี่ยนแปลงของโครงสร้างการออกแบบของคลาสและแผนภาพลำดับ ซึ่งจะใช้รูปแบบของการเปลี่ยนแปลงตามที่งานวิจัยได้กำหนดขึ้นตามตารางที่ 1 ในบทที่ 2 และโมเดลที่ 4 นั้นจะสร้างด้วยวิธีการเดียวกับโมเดลที่ 3 คือการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบแต่จะใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดที่มีความสัมพันธ์กับความเสถียรก่อนจะนำมาสร้างโมเดล (DLREA(PCA)) ซึ่งโมเดลที่ 3 และ 4 นี้จะไม่ใช้ตัววัดแผนภาพทั้ง 27 ตัววัดแต่จะเป็นการกำหนดตัววัดขึ้นมาใหม่ตามข้อกำหนดของโมเดล รายละเอียดของแต่ละวิธีในการสร้างโมเดลสามารถอธิบายได้ ดังต่อไปนี้



Class no.	1			2			3			…
Class name	Class A			Class B			Class C			…
Increment	1st	+2nd	+3rd	1st	+2nd	+3rd	1st	+2nd	+3rd	…
CLS	0.948	0.967	0.958	1	1	1	0.973	0.973	0.967	…
NumAttr	2	3	3	1	1	1	8	7	7	…
NumOps	8	7	7	0	0	0	0	0	2	…
…	…	…	…	…	…	…	…	…	…	…

รูปที่ 11 ภาพรวมของการสร้างโมเดลที่ใช้ตัววัดแผนภาพ (MRA, PCA)

3.5.1 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA)

การสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อนเป็นวิธีพื้นฐานทั่วไปในการสร้างโมเดล การวิเคราะห์ความถดถอยเชิงซ้อนจะสามารถเลือกตัวแปรอิสระมากกว่า 1 ตัวที่มีความสัมพันธ์กับตัวแปรตามซึ่งในงานวิจัยนี้ ตัวแปรอิสระก็คือตัววัดแผนภาพทั้งหมด 27 ตัว และตัวแปรตามก็คือความเสถียรเชิงตรรกะของคลาส โดยวิธีนี้จะเลือกตัววัดที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาสอย่างมีนัยสำคัญที่สุดเข้าสมการหรือโมเดลทีละตัว โดยดูจากค่าสัมประสิทธิ์สหสัมพันธ์เชิงส่วน (Partial correlation) และตัดตัววัดที่มีความสัมพันธ์กับตัววัดอื่นๆออกจนกระทั่งไม่สามารถนำตัววัดเข้าสู่สมการได้หรือไม่มีตัววัดที่สามารถตัดออกได้อีก ซึ่งลักษณะของโมเดลจะมีรูปแบบตามสมการที่ 4

$$y = a + b_1x_1 + b_2x_2 + \ldots + b_ix_i + e \quad (4)$$

โดยที่ y คือ ค่าประมาณความเสถียรเชิงตรรกะของคลาส

a คือ ค่าคงที่

b คือ ค่าสัมประสิทธิ์สหสัมพันธ์ของตัววัดแผนภาพ เมื่อ $i = 1,2,..,n$

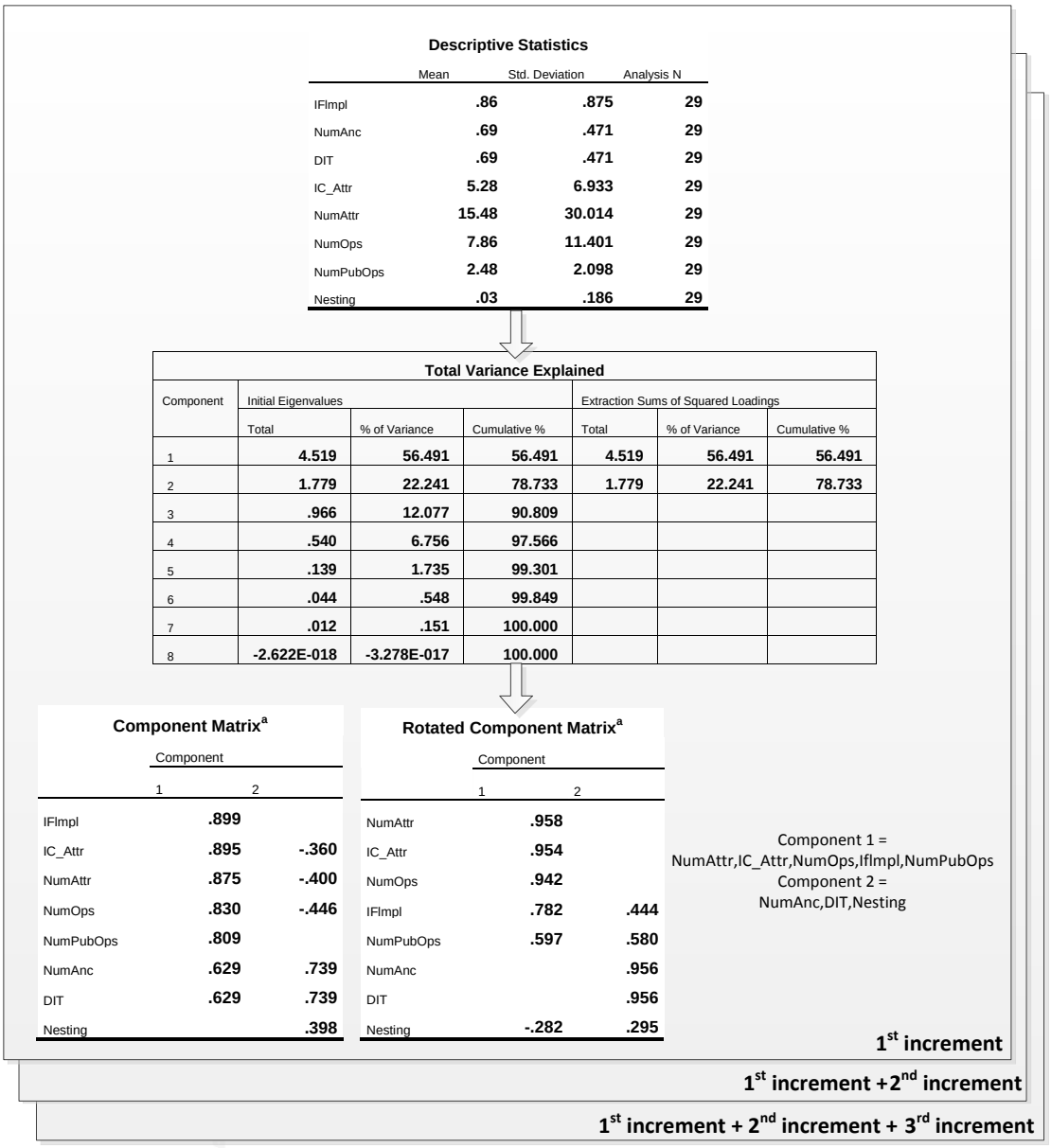
x คือ ตัววัดแผนภาพ เมื่อ $i = 1,2,..,n$

e คือ ค่าความคลาดเคลื่อน

3.5.2 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA)

การวิเคราะห์ส่วนประกอบหลักเป็นหนึ่งวิธีการในการสกัดส่วนประกอบเพื่อลดจำนวนตัวแปร โดยทำการค้นหาจำนวนองค์ประกอบที่มีความสามารถเพียงพอในการอธิบายความสัมพันธ์ระหว่างตัวแปรการสร้างโมเดลด้วยวิธีการวิเคราะห์ส่วนประกอบหลักเป็นแนวคิดที่จะทำการสกัดตัววัดแผนใน 27 ตัววัดที่มีความสัมพันธ์กันและสามารถนำไปประมาณค่าความเสถียรเชิงตรรกะของคลาสและตัดตัววัดที่ไม่มีความสัมพันธ์กันออกไป ก่อนที่จะนำตัววัดที่ได้มาสร้างโมเดลซึ่งจะมีลักษณะรูปแบบโมเดลเหมือนกับวิธีวิเคราะห์ความถดถอยเชิงซ้อนในสมการที่ 4

ในรูปที่ 12 เป็นตัวอย่างที่แสดงให้เห็นถึงวิธีการวิเคราะห์ส่วนประกอบหลักโดยใช้ค่าตัววัดแผนภาพของทุกๆคลาสในแต่ละอินคริเมนต์ ที่ผลลัพธ์ขั้นตอนสุดท้ายจะได้กลุ่มของตัววัดที่มีความสัมพันธ์กันที่สามารถนำไปอธิบายความเสถียรเชิงตรรกะของคลาสได้ จากนั้นจะเลือกกลุ่มของตัววัดที่มีค่าความสัมพันธ์สูงสุด ซึ่งจากตัวอย่างจะมี 2 กลุ่ม คือ กลุ่มที่ 1 (Component 1) ประกอบด้วย NumAttr, IC_Attr, NumOps, IFImpl, NumPubOps และกลุ่มที่ 2 (Component 2) ประกอบด้วย NumAnc, DIT, Nesting ซึ่งในกรณีนี้จะเลือกตัววัดจากกลุ่มที่ 1 เนื่องจากมีค่าความสัมพันธ์สูงสุดคือ 51.491% ซึ่งหมายถึงองค์ประกอบสามารถอธิบายความผันแปรทั้งหมดได้ 51.491% นำมาทำการวิเคราะห์ความถดถอยเชิงซ้อนเพื่อสร้างเป็นสมการหรือโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาส



Descriptive Statistics

	Mean	Std. Deviation	Analysis N
IFImpl	.86	.875	29
NumAnc	.69	.471	29
DIT	.69	.471	29
IC_Attr	5.28	6.933	29
NumAttr	15.48	30.014	29
NumOps	7.86	11.401	29
NumPubOps	2.48	2.098	29
Nesting	.03	.186	29

Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	4.519	56.491	56.491	4.519	56.491	56.491
2	1.779	22.241	78.733	1.779	22.241	78.733
3	.966	12.077	90.809			
4	.540	6.756	97.566			
5	.139	1.735	99.301			
6	.044	.548	99.849			
7	.012	.151	100.000			
8	-2.622E-018	-3.278E-017	100.000			

Component Matrix[a]

	Component 1	Component 2
IFImpl	.899	
IC_Attr	.895	-.360
NumAttr	.875	-.400
NumOps	.830	-.446
NumPubOps	.809	
NumAnc	.629	.739
DIT	.629	.739
Nesting		.398

Rotated Component Matrix[a]

	Component 1	Component 2
NumAttr	.958	
IC_Attr	.954	
NumOps	.942	
IFImpl	.782	.444
NumPubOps	.597	.580
NumAnc		.956
DIT		.956
Nesting	-.282	.295

Component 1 = NumAttr,IC_Attr,NumOps,IfImpl,NumPubOps
Component 2 = NumAnc,DIT,Nesting

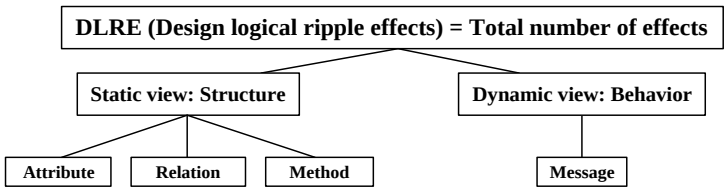
CHULALONGKORN UNIVERSITY

รูปที่ 12 ตัวอย่างวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA)

3.5.3 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA)

การสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบนั้น ในงานวิจัยนี้จะเป็นการออกแบบโปรแกรมเชิงวัตถุ ซึ่งจะใช้แผนภาพคลาสและแผนภาพลำดับในการนำมาวิเคราะห์ผลกระทบเชิงตรรกะ โดยมีแนวคิดมาจากการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบโปรแกรมเชิงโครงสร้าง ซึ่งจะใช้แผนผังโครงสร้างของมอดูลนำมาวิเคราะห์องค์ประกอบของมอดูลซึ่งประกอบด้วย ข้อมูลโกลบอล และพารามิเตอร์ เพื่อหาจำนวนผลกระทบจากการ

เปลี่ยนแปลงที่มีโอกาสเกิดขึ้น แต่งานวิจัยนี้จะนำแผนภาพคลาสและแผนภาพลำดับมาวิเคราะห์องค์ประกอบที่สามารถเกิดการเปลี่ยนแปลงแล้วจะส่งผลกระทบต่อโครงสร้างการออกแบบและความเสถียรเชิงตรรกะของคลาสนั้นๆ เพื่อหาจำนวนรวมของผลกระทบทั้งหมดของคลาสนั้นๆ



รูปที่ 13 แนวคิดของวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

จากรูปที่ 13 แสดงถึงแนวคิดของวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบจากแผนภาพคลาสซึ่งเป็นแผนภาพที่แสดงโครงสร้างรายละเอียดของคลาส และแผนภาพลำดับซึ่งเป็นแผนภาพที่แสดงพฤติกรรมของออบเจ็กต์ โดยสามารถแยกองค์ประกอบของแผนภาพทั้ง 2 แผนภาพได้แก่ แอททริบิวต์ (Attribute) ความสัมพันธ์ (Relation) เมธอด (Method) และการรับส่งสาร (Message) ดังนั้นผลกระทบเชิงตรรกะของการออกแบบจะเท่ากับผลรวมของ จำนวนรวมของผลกระทบจากแอททริบิวต์ (TA) จำนวนรวมของผลกระทบจากความสัมพันธ์ (TR) จำนวนรวมของผลกระทบจากเมธอด และจำนวนรวมของผลกระทบจากการรับส่งสาร (TS) โดยที่ b คือ ค่าสัมประสิทธิ์สหสัมพันธ์ของตัววัดจำนวนรวมของผลกระทบ ดังสมการที่ 5

$$DS_X = 1/(1+DLRE_X)$$
$$DLRE_X = a + b_i TA + b_{i+1} TR + b_{i+2} TM + b_{i+3} TS \quad (5)$$

โดยที่ a คือ ค่าคงที่

b คือ ค่าสัมประสิทธิ์ของตัววัด

จากงานวิจัยที่นำเสนอโมเดลประมาณค่าความเสถียรของการออกแบบมอดูลที่ได้กล่าวถึงในบทที่ 2 สมการที่ 3 นำมาเป็นแนวคิดของข้อกำหนดโมเดลในงานวิจัยนี้ซึ่งเป็นไปตามองค์ประกอบของโครงสร้างการออกแบบ โดยความเสถียรเชิงตรรกะของคลาสขึ้นอยู่กับจำนวนรวมของสมมติฐานผลกระทบเชิงตรรกะของการออกแบบคลาส ซึ่งก็คือจำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ เมธอด ความสัมพันธ์ และการรับส่งสาร หากผลกระทบเชิงตรรกะมากก็จะทำให้ความเสถียรเชิงตรรกะของคลาสนั้นๆลดลง และความเสถียรจะมีค่าเป็น 1 ถ้าไม่มีผลกระทบเชิงตรรกะเกิดขึ้นเมื่อมีการเปลี่ยนแปลง

เมื่อนำข้อกำหนดที่ได้มาสร้างตัววัดเข้าสู่โมเดล เพื่อให้ได้จำนวนรวมของผลกระทบเชิงตรรกะของการออกแบบ ซึ่งเรียกว่า การวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ โดยจะทำการวิเคราะห์เจาะจงไป จากความเปลี่ยนแปลงที่สามารถเกิดขึ้นกับซอฟต์แวร์ที่กำหนดไว้ในบทที่ 2 ตารางที่ 1 ซึ่งโมเดลที่ได้มีจะลักษณะดังสมการที่ 6 และพร้อมทั้งคำอธิบายตัววัดแต่ละตัวด้านล่างโมเดลดังนี้

$$DLRE_x = a + b_0 TA_x + b_1 \sum_{y \in J_{xy}} TA_{xy} + b_2 TR_x + b_3 \sum_{y \in J_{xy}} TR_{xy} + b_4 TM_x + b_5 \sum_{y \in J_{xy}} TM_{xy} + b_6 \sum_{y \in J_{xy}} TS_{xy} + b_7 \sum_{y \in J_{xy}} TS'_{xy} \quad (6)$$

โดยที่ DS_x คือ ความเสถียรของการออกแบบของคลาส x

$DLRE_x$ คือ ผลกระทบเชิงตรรกะของการออกแบบของคลาส x

x คือ คลาสที่สนใจในโปรแกรม

y คือ คลาสอื่นๆ ในโปรแกรม ที่มีความสัมพันธ์กับคลาส x

J_x คือ เซตของคลาสลูกของคลาส x

J_{xy} คือ เซตของคลาสที่มีความสัมพันธ์กับคลาส x

TA_x คือ จำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ที่ประกาศโดยคลาส x

TA_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ที่ประกาศโดยคลาส y และถูกใช้โดยคลาส x

TR_x คือ จำนวนรวมของผลกระทบที่เกิดจากความสัมพันธ์ของคลาสใน J_x

TR_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากความสัมพันธ์ของคลาสใน J_{xy}-J_x

TM_x คือ จำนวนรวมของผลกระทบที่เกิดจากเมธอดของคลาส x

TM_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากเมธอดของคลาส y โดยที่คลาส x สามารถเรียกใช้งาน

TS_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากการส่งสารจากคลาส x ไปยังคลาส y

TS'_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากการรับสารที่ถูกส่งจากคลาส y มายังคลาส x

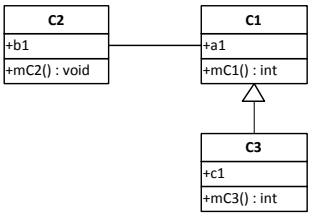
จากข้อกำหนดตัววัดเบื้องต้นนั้น การแทนค่าของตัววัดจะใช้การเปลี่ยนแปลงที่แสดงไว้ดังตารางที่ 1 ในบทที่ 2 ซึ่งใช้ในการคำนวณค่าความเสถียรจากซอร์ซโค้ด นำมาจำลองการเปลี่ยนแปลงกับแผนภาพคลาสและแผนภาพลำดับ ซึ่งแบ่งเป็นการเปลี่ยนแปลงของแอททริบิวต์ (Attribute) ความสัมพันธ์ (Relation) เมธอด (Method) และการรับส่งสาร (Message) ซึ่งในตารางที่ 1 นั้นจะสามารถจำลองการเปลี่ยนแปลงได้เพียงแผนภาพคลาส ซึ่งประกอบด้วย แอททริบิวต์ ความสัมพันธ์ และเมธอด ส่วนแผนภาพลำดับได้กำหนดเพิ่มเติมโดยการเปลี่ยนแปลงประกอบด้วย

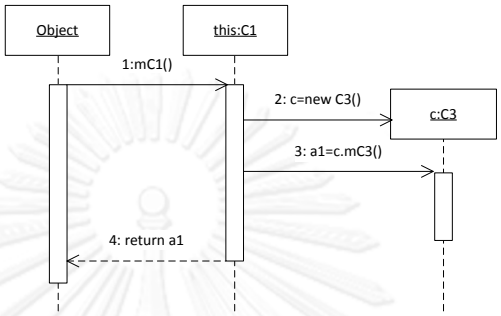
1) เพิ่มการส่งสารจากคลาส x ไปยังคลาส y
2) ลบการส่งสารจากคลาส x ไปยังคลาส y
3) เพิ่มการรับสารที่ถูกส่งจากคลาส y มายังคลาส x
4) ลบการรับสารที่ถูกส่งจากคลาส y มายังคลาส x

จากการเปลี่ยนแปลงดังกล่าวสามารถรวบรวมและจำแนกความเปลี่ยนแปลงและผลกระทบ เพื่อนำมาใช้เป็นต้นแบบในการจำลองความเปลี่ยนแปลงและแทนค่าตัววัดทั้ง 8 ตัววัด ดังตารางที่ 11

ตารางที่ 11 การเปลี่ยนแปลงและการแทนค่าตัววัด จำแนกตามองค์ประกอบของแผนภาพ

แอททริบิวต์ (Attribute)		**เมธอด (Method)**	
การเปลี่ยนแปลง	**แทนค่าตัววัด**	**การเปลี่ยนแปลง**	**แทนค่าตัววัด**
Add attribute (when scope is public)	TA_{xy}	Add method (when scope is public)	TM_{xy}
Delete attribute		*Delete method*	
Scope is public	TA_x, TA_{xy}	Scope is public	TM_x, TM_{xy}
Scope is private	TA_x	Scope is private	TM_x
Scope is protect	TA_x	Scope is protect	TM_x
Change attribute value		*Change method name*	
Scope is public	TA_x, TA_{xy}	Scope is public	TM_x, TM_{xy}
Scope is private	TA_x	Scope is private	TM_x
Scope is protect	TA_x	Scope is protect	TM_x
Change attribute type		*Change method signature*	
Scope is public	TA_x, TA_{xy}	Scope is public	TM_x, TM_{xy}
Scope is private	TA_x	Scope is private	TM_x
Scope is protect	TA_x	Scope is protect	TM_x
Change attribute name		*Change return type*	
Scope is public	TA_x, TA_{xy}	Scope is public	TM_x, TM_{xy}
Scope is private	TA_x	Scope is private	TM_x
Scope is protect	TA_x	Scope is protect	TM_x
Change attribute scope		*Change method description*	
From public to private	TA_{xy}	Scope is public	TM_x, TM_{xy}
From protect to private	TA_{xy}	Scope is private	TM_x
From private to public	TA_{xy}	Scope is protect	TM_x
From private to protect	TA_{xy}	*Change method scope*	
ความสัมพันธ์ (Relation)		From public to private	TM_{xy}
การเปลี่ยนแปลง	**แทนค่าตัววัด**	From protect to private	TM_{xy}
Add super class of y	TR_x, TR_{xy}	From private to public	TM_{xy}
Delete super class of y	TR_x, TR_{xy}	From private to protect	TM_{xy}
Add aggregate class relation of y	TR_x	**การรับส่งสาร (Message)**	
Delete aggregate class relation of y	TR_x, TR_{xy}	**การเปลี่ยนแปลง**	**แทนค่าตัววัด**
Add association class relation of y	TR_x	Add return message	TS'_{xy}
Delete association class relation of y	TR_x, TR_{xy}	Delete return message	TS'_{xy}
Add class of y	TR_x, TR_{xy}	Add send message	TS_{xy}
Delete class of y	TR_x, TR_{xy}	Delete send message	TS_{xy}





รูปที่ 14 ตัวอย่างแผนภาพคลาสและแผนภาพลำดับ

ตารางที่ 12 ตัวอย่างการแทนค่าตัววัดของคลาส C1

แผนภาพ	องค์ประกอบ	การเปลี่ยนแปลง	นับค่าตัววัดของคลาส C1
แผนภาพคลาส	แอททริบิวต์ (Attribute)	ลบ a1	TA_x
		เปลี่ยนขอบเขต a1 เป็น private	TA_x
		เปลี่ยนขอบเขต a1 เป็น protect	TA_x
		เพิ่มแอททริบิวต์ public ใน C2	TA_{xy}
	ความสัมพันธ์ (Relation)	ลบความสัมพันธ์ association ของ C2	TR_{xy}
		ลบคลาส C2	TR_{xy}
		ลบคลาส C3	TR_x
		เพิ่มความสัมพันธ์ association ของ C3	TR_x
		เพิ่มความสัมพันธ์ aggregation ของ C3	TR_x
	เมธอด (Method)	ลบ mC1()	TM_x
		เปลี่ยนขอบเขต mC1() เป็น private	TM_x
		ลบ mC3()	TM_{xy}
		เปลี่ยนขอบเขต mC3() เป็น private	TM_{xy}
		เปลี่ยนขอบเขต mC3() เป็น protected	TM_{xy}
แผนภาพลำดับ	การรับส่งสาร (Message)	ลบ send message 2:c=new C3()	TS_{xy}
		ลบ send message 3:a1=c.mC3()	TS_{xy}
		เพิ่ม return message หลังจาก send message 3:a1=c.mC3()	TS'_{xy}

จากตัวอย่างดังกล่าวเป็นเพียงการจำลองการเปลี่ยนแปลงบางส่วนที่มีความเป็นไปได้ ซึ่งจะสามารถแทนค่าตัววัดของคลาส C1 โดยจากโมเดล สามารถจำแนกตามตัววัดได้ดังนี้

- คลาสที่สนใจ *x* คือ *C1*
- คลาสอื่นๆในโปรแกรม ที่มีความสัมพันธ์กับคลาส *C1* คือ *y={C2, C3}*
- คลาสลูกของคลาส *C1* คือ $J_x = \{C3\}$
- คลาสอื่นๆที่มีความสัมพันธ์กับคลาส C1 คือ $J_{xy} = \{C2, C3\}$
- จำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ที่ประกาศโดยคลาส *C1* คือ $TA_x = 3$
- จำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ที่ประกาศโดยคลาส *y* และถูกใช้โดยคลาส *C1* คือ $TA_{xy} = 1$
- จำนวนรวมของผลกระทบที่เกิดจากความสัมพันธ์ของคลาสใน J_x คือ $TR_x = 3$
- จำนวนรวมของผลกระทบที่เกิดจากความสัมพันธ์ของคลาสใน J_{xy} คือ $TR_{xy} = 2$
- จำนวนรวมของผลกระทบที่เกิดจากเมธอดของคลาส *C1* คือ $TM_x = 2$
- จำนวนรวมของผลกระทบที่เกิดจากเมธอดของคลาส *y* โดยที่คลาส *C1* สามารถเรียกใช้งานคือ $TM_{xy} = 3$
- จำนวนรวมของผลกระทบที่เกิดจากการส่งสารจากคลาส *C1* ไปยังคลาส *y* คือ $TS_{xy} = 2$
- จำนวนรวมของผลกระทบที่เกิดจากการรับสารที่ถูกส่งจากคลาส *y* มายังคลาส *C1* คือ $TS'_{xy} = 1$

จากวิธีการแทนค่าตัววัดนี้จะนำไปใช้ในการหาค่าตัววัดในแต่ละคลาสของชุดโปรแกรมสำหรับสร้างโมเดลและชุดโปรแกรมสำหรับประเมินผลโมเดล ในการสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะ (DLREA) และโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA))

3.5.4 การสร้างโมเดลการประมาณค่าความเสถียรด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA))

โมเดลที่ 4 นี้มีแนวคิดและวิธีการมาจากโมเดลที่ 3 เพียงแต่จะนำตัววัดที่กำหนดขึ้นทั้งหมด 8 ตัว ตามสมการที่ 6 มาคัดเลือกตัววัดที่มีความสัมพันธ์กับความเสถียรของการออกแบบโดยใช้วิเคราะห์ส่วนประกอบหลักแล้วนำไปสร้างโมเดลและหาค่าสัมประสิทธิ์ของตัววัดด้วยการวิเคราะห์ความถดถอยเชิงซ้อน ซึ่งวิธีการนี้ก็จะเป็นการนำองค์ความรู้ที่ได้จากการสร้างโมเดลที่ 1,2 และ 3 นำมาประยุกต์และปรับโมเดลที่ 4 ให้ได้โมเดลที่มีประสิทธิภาพในการประมาณค่ามากขึ้นจากโมเดลที่ 3 ซึ่งไม่มีการคัดเลือกตัววัดในการสร้างโมเดล โดยข้อมูลในการสร้างโมเดลก็จะใช้ชุดเดียวกับโมเดลที่ 3 ซึ่งจะอธิบายรายละเอียดของโมเดลที่ได้ในบทที่ 4

3.6 ขั้นตอนการประเมินผลและเปรียบเทียบโมเดล

เมื่อได้โมเดลสำหรับการประมาณค่าความเสถียรของการออกแบบจากแผนภาพคลาสและแผนภาพลำดับแล้ว ประเมินผลโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสด้วยการนำชุดโปรแกรมทดสอบมาหาค่าประมาณความเสถียรเชิงตรรกะของคลาส เพื่อเปรียบเทียบกับค่าที่ได้จากการคำนวณความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ด โดยใช้การหาค่าเฉลี่ยผิดพลาดสัมพัทธ์โดยใช้สมการที่ 7 และการทำนายที่ระดับแอล [19] ดังสมการที่ 8 ซึ่งทั้ง 2 วิธีนี้เป็นวิธีการที่ใช้ในการประเมินผลระบบการประมาณค่า ซึ่งมีรายละเอียดดังนี้

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \tag{7}$$

$$MRE = |RE| = \left|\frac{E - \hat{E}}{E}\right|$$

โดยที่ $MMRE$ คือ ค่าเฉลี่ยความผิดพลาดสัมพัทธ์

MRE คือ ขนาดความผิดพลาดสัมพัทธ์

RE คือ ความผิดพลาดสัมพัทธ์

E คือ ค่าความเสถียรเชิงตรรกะ

$\hat{E}$ คือ ค่าประมาณความเสถียรของการออกแบบ

$$PRED(l) = \frac{k}{n} \tag{8}$$

โดยที่ $PRED(l)$ คือ การทำนายที่ระดับแอล

l คือ ช่วงระดับความคลาดเคลื่อนมีค่าตั้งแต่ 0-1

k คือ จำนวนหน่วยตัวอย่างที่มีค่า $MRE \le l$

n คือ จำนวนหน่วยตัวอย่างทั้งหมด

โดยผลจากการประเมินผลสามารถอธิบายได้เช่น ถ้า *PRED*(0.15)=0.80 หมายความว่า 80% ของค่าที่ประมาณได้จากโมเดลจะมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 15% และการทำนายโดยทั่วไปจะยอมรับโมเดลประมาณค่าที่ผลการทำนายที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 [19] ซึ่งหมายความว่าค่าประมาณที่ได้ต้องคลาดเคลื่อนจากค่าจริงไม่เกิน 25% เป็นจำนวน 75% ขึ้นไปจากการประมาณค่าทั้งหมด

การสร้างโมเดลเพื่อการประมาณค่าจะต้องมีการประเมินผลโมเดลเพื่อให้ทราบว่าโมเดลนั้นๆ มีความแม่นยำ สามารถที่จะยอมรับและนำไปใช้ในการประมาณค่าแล้วให้ผลที่ใกล้เคียงค่าจริง งานวิจัยนี้ใช้การประเมินผลโมเดลและยอมรับโมเดลตามมาตรฐานทั่วไปที่ได้อธิบายไว้เบื้องต้นแล้ว และการเปรียบเทียบโมเดลทั้ง 4 โมเดลนั้นก็จะทำการเปรียบเทียบจากผลการการประเมินโมเดล ซึ่ง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หากโมเดลใดมีค่าผลการประเมินที่ดีที่สุดหรือความคลาดเคลื่อนต่ำสุด ก็จะสามารถนำไปใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาสได้ใกล้เคียงกับค่าจริงมากที่สุด

ในงานวิจัยนี้ใช้ชุดโปรแกรมสำหรับสร้างโมเดลและสำหรับประเมินผลโมเดลที่มีการพัฒนาแบบอินคริเมนทัล ซึ่งทำการเลือกจากหมายเลขเวอร์ชันที่แสดงถึงการเพิ่มฟีเจอร์หรือฟังก์ชันของเวอร์ชันถัดไป โดยเวอร์ชันที่ 1 จะหมายถึงอินคริเมนต์ที่ 1 และเวอร์ชันที่ 2 จะหมายถึงการรวมโครงสร้างของอินคริเมนต์ที่ 1 กับอินคริเมนต์ที่ 2 ซึ่งสิ่งที่นอกเหนือจากการสร้างโมเดลด้วยวิธีการทั้งหมด 4 วิธีการนั้น คือการเพิ่มเติมของการสร้างโมเดลโดยการแบ่งชุดโปรแกรมสำหรับสร้างโมเดลตามหมายเลขเวอร์ชัน ซึ่งในงานวิจัยมีจำนวนสูงสุด 3 เวอร์ชัน ดังนั้นในแต่ละวิธีของการสร้างโมเดลจะประกอบด้วยโมเดลอีกจำนวน 3 โมเดล และนำโมเดลที่ได้นั้นไปประเมินผลด้วยชุดโปรแกรมสำหรับประเมินผลซึ่งแบ่งออกตามเวอร์ชันจำนวน 3 เวอร์ชันเช่นเดียวกัน จากนั้นนำโมเดลที่ได้ไปประมาณค่าความเสถียรเชิงตรรกะของคลาสที่แบ่งตามเวอร์ชันจากชุดโปรแกรมสำหรับประเมินผลเพื่อนำค่าประมาณที่ได้ไปประเมินผลโมเดลและเปรียบเทียบผลการประเมินโมเดลที่สร้างด้วยวิธีการต่างๆ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4

การสร้างและการประเมินผลโมเดลประมาณค่าความเสถียรเชิงตรรกะ

ในบทนี้จะกล่าวถึงการสร้างโมเดล การประเมินผล และการเปรียบเทียบโมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล ตามที่นำเสนอวิธีการในการสร้างโมเดลทั้งหมด 3 วิธีการในบทที่ 3 โดยจะแบ่งหัวข้อออกเป็นส่วนหลักๆ คือ การสร้างโมเดล วัตถุประสงค์ของการประเมินผลโมเดล การประเมินผลโมเดล และการเปรียบเทียบการประเมินผลโมเดลการประมาณค่าความเสถียร

สำหรับชุดโปรแกรมสำหรับสร้างและทดสอบหรือประเมินผลโมเดล แสดงรายละเอียดชื่อโปรแกรม ที่มา จำนวนคลาสในแต่ละเวอร์ชัน และรายละเอียดดังตารางที่ 13 ซึ่งแบ่งออกเป็น ชุดโปรแกรมสำหรับสร้างโมเดลจำนวน 5 โปรแกรม และชุดโปรแกรมสำหรับประเมินผลโมเดลจำนวน 15 โปรแกรม รวมจำนวนคลาสในแต่ละเวอร์ชันดังตารางที่ 14

ในตารางที่ 13 และ 14 ลำดับเวอร์ชันมีความหมายคือ เวอร์ชันที่ 1 เป็นจำนวนคลาสของอินคริเมนต์แรก เวอร์ชันที่ 2 เป็นจำนวนคลาสของอินคริเมนต์แรกรวมกับจำนวนคลาสของอินคริเมนต์ที่สอง และเวอร์ชันที่ 3 เป็นจำนวนคลาสของอินคริเมนต์ทั้งสามอินคริเมนต์รวมกัน

ตารางที่ 13 ชุดโปรแกรมสำหรับสร้างและประเมินผลโมเดล

ลำดับที่	ชื่อโปรแกรม	ที่มา	จำนวนคลาส			รายละเอียด
			เวอร์ชันที่ 1	เวอร์ชันที่ 2	เวอร์ชันที่ 3	
ชุดโปรแกรมสำหรับสร้างโมเดล						
1	Edastext	Sourceforge	1	7	18	โปรแกรมสำหรับการจัดการข้อความ
2	RedImage	Sourceforge	16	25	-	โปรแกรมตกแต่งรูปภาพ
3	Java Game Maker	Sourceforge	8	29	41	โปรแกรมสำหรับการสร้างเกมภาษาจาวา
4	TextTrix	Sourceforge	22	67	-	โปรแกรมสำหรับการจัดการข้อความ
5	Encounter	CodePlex	1	5	16	โปรแกรมสำหรับการคำนวณ
ชุดโปรแกรมสำหรับประเมินผลโมเดล						
1	MsmqJava	CodePlex	4	7	-	โปรแกรมสำหรับการเข้าถึง MSMQ
2	SWeDiT	CodePlex	1	223	-	โปรแกรมสำหรับการจัดการข้อความ

ตารางที่ 13 ชุดโปรแกรมสำหรับสร้างและประเมินผลโมเดล (ต่อ)

ลำดับที่	ชื่อโปรแกรม	ที่มา	จำนวนคลาส			รายละเอียด
			เลเวลชั้นที่ 1	เลเวลชั้นที่ 2	เลเวลชั้นที่ 3	
ชุดโปรแกรมสำหรับประเมินผลโมเดล						
3	MasterMind Solver	CodePlex	4	5	-	เกม(ภาษาจาวา)
4	Jscicalc	Sourceforge	22	113	-	โปรแกรมสำหรับการคำนวณ
5	Proguard	Sourceforge	95	199	368	โปรแกรมสำหรับสร้างโปรแกรมภาษาจาวา
6	Hodoku	Sourceforge	60	65	111	โปรแกรมสำหรับการสร้าง แก้ไข และวิเคราะห์การเขียนโปรแกรมภาษาจาวา
7	sahi	Sourceforge	141	345	-	โปรแกรมสำหรับการทดสอบการใช้งานเว็บแอปพลิเคชัน
8	jetris	Sourceforge	7	18	-	เกม(ภาษาจาวา)
9	FreeCell Java Solver	Sourceforge	11	18	-	เกม(ภาษาจาวา)
10	Krut Computer Recorder	Sourceforge	13	15	111	โปรแกรมสำหรับการบันทึกเสียงและวิดีโอจากหน้าจอคอมพิวเตอร์
11	jMemorize	Sourceforge	20	77	166	โปรแกรมสำหรับการบันทึกเตือนความจำ
12	Ibiiztera	Sourceforge	160	382	-	โปรแกรมสำหรับจัดการรูปภาพ
13	SubToText	Sourceforge	7	229	-	โปรแกรมสำหรับการแปลงไฟล์ DVD เป็น SRT
14	Tumblr Image Downloader	Sourceforge	1	2	225	โปรแกรมสำหรับการดาวน์โหลดรูปภาพจากเว็บไซต์ www.tumblr.com
15	Raw2vmdk	Sourceforge	4	173	-	โปรแกรมสำหรับจัดการไฟล์รูปภาพ

ตารางที่ 14 จำนวนคลาสในแต่ละเวอร์ชันของชุดโปรแกรม

ชุดโปรแกรม	จำนวนโปรแกรม	จำนวนรวมคลาส		
		เวอร์ชัน 1	เวอร์ชัน 2	เวอร์ชัน 3
สำหรับสร้างโมเดล	5	48	133	75
สำหรับประเมินผลโมเดล	15	550	1,871	981

4.1 การสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะ

จากขั้นตอนการสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะที่นำเสนอไว้ในบทที่ 3 สามารถอธิบายรายละเอียดในการสร้างโมเดลตามวิธีการที่แตกต่างกัน ดังนี้

4.1.1 การสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA)

เมื่อทำการเก็บค่าตัววัดแผนภาพทั้งหมด 27 ตัววัด จากชุดโปรแกรมสำหรับสร้างโมเดลทั้งหมด 5 โปรแกรม โดยแต่ละโปรแกรมประกอบด้วย 2 ถึง 3 เวอร์ชัน จำนวนรวมคลาสทั้งหมด 256 คลาส นำไปประมวลผลในเครื่องมือด้วยการวิเคราะห์ความถดถอยเชิงซ้อน ในตารางที่ 15 แสดงตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดแผนภาพที่ใช้ในการสร้างโมเดล ซึ่งโปรแกรมนี้ประกอบด้วย 3 เวอร์ชัน โดยเวอร์ชันที่ 1 คือ อินคริเมนต์ที่ 1 (1^{st}) เวอร์ชันที่ 2 คือ เมื่อเพิ่มส่วนการออกแบบของอินคริเมนต์ที่ 2 ($+2^{nd}$) และเวอร์ชันที่ 3 คือ เมื่อเพิ่มส่วนการออกแบบของอินคริเมนต์ที่ 3 ($+3^{rd}$) ในตารางแสดงค่าตัววัดแผนภาพเพียง 11 ตัววัด ซึ่งในความเป็นจริงงานวิจัยนี้จะเก็บค่าตัววัดแผนภาพทั้งหมด 27 ตัววัด และเครื่องมือจะวิเคราะห์ผลโดยใช้ข้อมูลทางสถิติที่แสดงดังตารางที่ 16 พร้อมทั้งแสดงกราฟดังรูปที่ 15 และรูปแบบของโมเดลดังสมการที่ 9

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 15 ตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดแผนภาพที่ใช้ในการสร้างโมเดล (โปรแกรม Edastext)

Increment	ID	Class Name	Class Logical Stability	NOC	NOP	IFImpl	NumDesc	NumAnc	NumAttr	NumOps	NumPubOps	Nesting	Connectors	MsgSelf	...
1st	1	EdasTexter	0.18519	0	0	2	0	1	50	17	3	0	0	0	...
+ 2nd	1	EdasTexter	0.53595	0	0	2	0	1	90	31	4	0	0	0	...
	2	ShowEdasTexterInfo	0.91755	0	0	1	0	1	7	4	2	0	0	0	...
	3	EdasPrinter	0.97507	0	0	0	0	1	3	2	1	0	0	0	...
	4	EdasPrinterPlus	0.96261	0	0	1	0	0	6	3	3	0	0	0	...
	5	EdasProxySetter	0.93289	0	0	2	0	1	11	6	6	0	0	0	...
	6	EdasUrlReader	0.97795	0	0	0	0	0	4	1	0	0	0	0	...
	7	ProxyObject	0.94247	0	0	0	0	0	3	9	1	0	0	0	...
+ 3rd	1	EdasTexter	0.74499	0	0	3	0	1	140	58	7	0	0	0	...
	2	ShowEdasTexterInfo	0.95469	0	0	1	0	1	7	4	2	0	0	0	...
	3	EdasPrinter	0.98174	0	0	0	0	1	4	2	1	0	0	0	...
	4	EdasPrinterPlus	0.97260	0	0	1	0	0	6	3	3	0	0	0	...
	5	EdasProxySetter	0.95083	0	0	2	0	1	11	6	6	0	0	0	...
	6	EdasUrlReader	0.98384	0	0	0	0	0	4	1	0	0	0	0	...
	7	ProxyObject	0.95785	0	0	0	0	0	3	8	0	0	0	0	...
	8	EdasColorChooser	0.95680	0	0	2	0	1	27	7	6	0	0	0	...
	9	EdasFinder	0.98911	0	0	0	0	0	0	5	0	0	0	0	...
	10	EdaSplash	0.98946	0	0	0	0	1	1	4	3	0	0	0	...
	11	EdasPrinterSetup	0.96558	0	0	1	0	1	17	8	2	0	0	0	...
	12	EdasUndoSetup	0.97295	0	0	1	0	1	12	4	4	0	0	0	...
	13	EIntField	0.99227	0	0	0	0	1	0	3	3	0	0	0	...
	14	ETextArea	0.96347	0	0	0	0	1	0	3	2	1	0	0	...
	15	PrinterObject	0.97612	0	0	0	0	0	4	10	0	0	0	0	...
	16	ShowJavaVersion	0.98314	0	0	1	0	1	7	4	2	0	0	0	...
	17	ShowShortKeys	0.98103	0	0	1	0	1	9	4	2	0	0	0	...
	18	ShowUserInfo	0.98314	0	0	1	0	1	7	4	2	0	0	0	...

ตารางที่ 16 สรุปข้อมูลทางสถิติสำหรับการวิเคราะห์ความถดถอยเชิงซ้อน

Metrics	Obs.	Obs. with missing data	Obs. without missing data	Min.	Max.	Mean	Std. deviation
CLS	256	0	256	0.19	1.00	0.94	0.10
NOC	256	0	256	0.00	1.00	0.01	0.11
NOP	256	0	256	0.00	0.00	0.00	0.00
OpsInh	256	0	256	0.00	699.00	21.19	100.35
AttrInh	256	0	256	0.00	100.00	4.99	15.57
NumDesc	256	0	256	0.00	1.00	0.01	0.11
NumAnc	256	0	256	0.00	5.00	0.66	0.82
DIT	256	0	256	0.00	5.00	0.66	0.82
CLD	256	0	256	0.00	1.00	0.01	0.11
IFImpl	256	0	256	0.00	5.00	0.41	0.77
Assoc_out	256	0	256	0.00	0.00	0.00	0.00
Assoc_In	256	0	256	0.00	0.00	0.00	0.00
NumAss_User	256	0	256	0.00	0.00	0.00	0.00
NumAss_Provider	256	0	256	0.00	0.00	0.00	0.00
EC_Par	256	0	256	0.00	28.00	0.15	1.83
IC_Par	256	0	256	0.00	44.00	0.51	4.09
EC_Attr	256	0	256	0.00	22.00	0.37	2.23
IC_Attr	256	0	256	0.00	31.00	2.14	3.83
MsgSent	256	0	256	0.00	159.00	1.09	10.69
MsgRecv	256	0	256	0.00	63.00	0.61	4.85

เมื่อนำข้อมูลของชุดโปรแกรมสำหรับสร้างโมเดลทั้งหมดที่สรุปข้อมูลทางสถิติทั้งหมดในตารางที่ 16 มาวิเคราะห์ความถดถอยเชิงซ้อนจะได้ค่าสัมประสิทธิ์ความถดถอยของตัววัด ซึ่งจะปรากฏในโมเดลที่ได้ดังตารางที่ 17

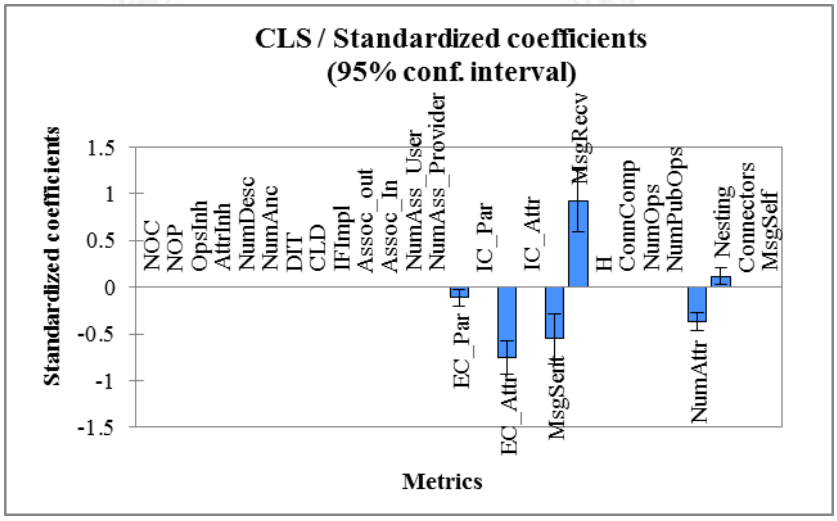
ตารางที่ 17 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน

Metrics	Value	Standard error	t	Pr > |t|	Lower bound (95%)	Upper bound (95%)
NOC	0.000	0.000				
NOP	0.000	0.000				
OpsInh	0.000	0.000				
AttrInh	0.000	0.000				
NumDesc	0.000	0.000				
NumAnc	0.000	0.000				
DIT	0.000	0.000				
CLD	0.000	0.000				
IFImpl	0.000	0.000				

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 17 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (ต่อ)

Metrics	Value	Standard error	t	Pr > |t|	Lower bound (95%)	Upper bound (95%)
Assoc_out	0.000	0.000				
Assoc_In	0.000	0.000				
NumAss_User	0.000	0.000				
NumAss_Provider	0.000	0.000				
EC_Par	**-0.110**	**0.047**	**-2.355**	**0.019**	**-0.202**	**-0.018**
IC_Par	0.000	0.000				
EC_Attr	**-0.749**	**0.092**	**-8.184**	**< 0.0001**	**-0.929**	**-0.569**
IC_Attr	0.000	0.000				
MsgSent	**-0.553**	**0.133**	**-4.142**	**< 0.0001**	**-0.816**	**-0.290**
MsgRecv	**0.920**	**0.166**	**5.541**	**< 0.0001**	**0.593**	**1.246**
H	0.000	0.000				
ConnComp	0.000	0.000				
NumOps	0.000	0.000				
NumPubOps	0.000	0.000				
NumAttr	**-0.371**	**0.049**	**-7.595**	**< 0.0001**	**-0.467**	**-0.275**
Nesting	**0.119**	**0.047**	**2.514**	**0.013**	**0.026**	**0.212**
Connectors	0.000	0.000				
MsgSelf	0.000	0.000				



รูปที่ 15 กราฟแสดงค่าสัมประสิทธิ์การถดถอยของตัววัด 27 ตัววัด

จากการวิเคราะห์ความถดถอยเชิงซ้อนจะได้โมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสซึ่งประกอบด้วยตัววัดแผนภาพ 6 ตัววัด ได้แก่ EC_Par คือ ตัววัดจำนวนครั้งที่คลาสที่สนใจถูก

นำไปใช้เป็นพารามิเตอร์ในคลาสอื่นและคลาสตัวเอง EC_Attr คือ ตัววัดจำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นแอททริบิวต์ในคลาสอื่นและคลาสตัวเอง MsgSent คือ ตัววัดจำนวนครั้งในการส่งสารไปยังคลาสอื่นๆ MsgRecv คือ ตัววัดจำนวนครั้งที่ได้รับสารจากคลาสอื่นๆ NumAttr คือ ตัววัดจำนวนแอททริบิวต์ของคลาส และ Nesting คือ ตัววัดระดับความลึกคลาสภายในคลาสอื่น

$$CLS = 0.962795875 - (0.006160384)EC_Par - (0.034378862)EC_Attr - (0.005303055)MsgSent + (0.019420127)MsgRecv - (0.002725783)NumAttr + (0.037487872)Nesting \quad (9)$$

จากโมเดลที่ได้จะพบว่าความเสถียรเชิงตรรกะของคลาสใดๆ จะมีค่าเพิ่มขึ้นเมื่อมีจำนวนการรับสารมาก หรือมีระดับความลึกคลาสภายในคลาสอื่นมาก ในขณะที่ความเสถียรเชิงตรรกะของคลาสจะมีค่าลดลงเมื่อจำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นพารามิเตอร์ในคลาสอื่นและคลาสตัวเองมาก จำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นแอททริบิวต์ในคลาสอื่นและคลาสตัวเองมาก จำนวนครั้งในการส่งสารไปยังคลาสอื่นมาก หรือจำนวนแอททริบิวต์ของคลาสมาก

ตารางที่ 18 ตัวอย่างข้อมูลซ้ำจากการรวบรวมข้อมูลชุดโปรแกรมสำหรับสร้างโมเดล

Increment	ID	Class Name	Class Logical Stability	NOC	NOP	IFImpl	NumDesc	NumAnc	NumAttr	NumOps	NumPubOps	Nesting	Connectors	MsgSelf	...
1st	1	ColorChooser	0.9621	0	0	1	0	1	4	8	8	0	0	0	...
	2	Element	0.9721	0	0	0	0	0	2	5	5	0	0	0	...
	3	GridFrame	0.8349	0	0	1	0	1	14	8	4	0	0	0	...
	4	FileMenuHelper	0.8349	0	0	0	0	5	21	17	1	0	0	0	...
+ 2nd	1	ColorChooser	0.9621	0	0	1	0	1	4	8	8	0	0	0	...
	2	Element	0.9721	0	0	0	0	0	2	5	5	0	0	0	...
	3	GridFrame	0.8255	0	0	1	0	1	19	10	4	0	0	0	...
	4	FileMenuHelper	0.8060	1	0	0	1	0	12	16	1	0	0	1	...
	5	GridLines	0.9672	0	0	0	0	1	3	7	7	0	0	0	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

จากตารางที่ 18 แสดงให้เห็นว่าการรวบรวมข้อมูลชุดโปรแกรมสำหรับสร้างโมเดลในแต่ละเวอร์ชัน เพื่อนำมาสร้างโมเดลจะมีโอกาสในการซ้ำของข้อมูล จากตัวอย่างข้อมูลที่ซ้ำซ้อนกันนั้นประกอบด้วย ค่าความเสถียรเชิงตรรกะของคลาส และค่าของตัววัดแผนภาพที่มีค่าเท่ากัน เนื่องจากจะมีคลาสบางส่วนในเวอร์ชันก่อนหน้า ไม่มีความสัมพันธ์ใดๆ หรือไม่ได้รับผลกระทบจากการเปลี่ยนแปลงเมื่อเพิ่มอินคริเมนต์ถัดไป ซึ่งในทางปฏิบัติของการสร้างโมเดลนั้นจะไม่นำข้อมูลที่มีค่าซ้ำกันมาสร้างโมเดล จึงทำการศึกษาและทดลองเพิ่มเติมเพื่อสร้างโมเดลโดยแยกชุดข้อมูลตามเวอร์ชัน

เมื่อแบ่งชุดโปรแกรมตามเวอร์ชันเพื่อสร้างโมเดลด้วยการวิเคราะห์ความถดถอยเชิงซ้อนของแต่ละเวอร์ชัน โดยจำนวนข้อมูลของแต่ละเวอร์ชันแสดงจำนวนดังตารางที่ 14 ประกอบด้วย 3 เวอร์ชัน ซึ่งขั้นตอนในการสร้างมีลักษณะเช่นเดียวกันกับการสร้างโมเดลในสมการที่ 9 โดยแสดงรูปแบบของโมเดลที่ได้ในแต่ละเวอร์ชันดังสมการที่ 10, 11 และ 12 ดังนี้

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (MRA) ในเวอร์ชันที่ 1

$$CLS = 0.909641823 + 0.075957712 * IFImpl - 0.038981618 * EC_Attr - 0.003920525 * NumAttr \quad (10)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (MRA) ในเวอร์ชันที่ 2

$$CLS = 0.985276536066338 + 0.000439833 * OpsInh - 0.001664176 * AttrInh - 0.021771789 * NumAnc - 0.007766751 * EC_Par + 0.004454723 * IC_Par - 0.002909957 * IC_Attr - 0.003842317 * MsgSent - 0.006435456 * NumOps + 0.005294336 * NumPubOps + 0.041103053 * Nesting + 0.001032558 * MsgSelf \quad (11)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (MRA) ในเวอร์ชันที่ 3

$$CLS = 1.009801661 - 0.001506319 * OpsInh - 0.007942732 * IC_Attr - 0.000823225 * NumOps \quad (12)$$

จากโมเดลที่ได้จะพบว่าหากชุดข้อมูลที่ใช้ในการสร้างโมเดลต่างกันก็จะได้ตัววัดที่ปรากฏในโมเดลต่างกัน เนื่องจากค่าของตัววัดในแต่ละคลาสของแต่ละเวอร์ชันมีค่าที่แตกต่างกัน ดังนั้นการสร้างโมเดลในการประมาณค่าความเสถียรจึงต้องสร้างเพื่อให้มีความเหมาะสมกับการนำไปใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาส โดยเป็นโมเดลที่มีรูปแบบเฉพาะเจาะจงกับโปรแกรมในแต่ละเวอร์ชัน

4.1.2 การสร้างโมเดลด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA)

ข้อมูลที่จะใช้ในการสร้างโมเดลด้วยวิธีการวิเคราะห์ส่วนประกอบหลักนี้เป็นข้อมูลจากชุดโปรแกรมสำหรับสร้างโมเดลเหมือนกับข้อมูลที่ใช้ในการสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อนมีลักษณะดังตารางที่ 15 และเครื่องมือจะวิเคราะห์ผลโดยใช้ข้อมูลทางสถิติตามตารางที่ 16 และจะแสดงผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลัก ดังตารางที่ 19 ซึ่งจะแสดงตัววัดที่สกัดได้จากตัววัดทั้งหมด 27 ตัววัด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 19 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลักของตัววัด 27 ตัว

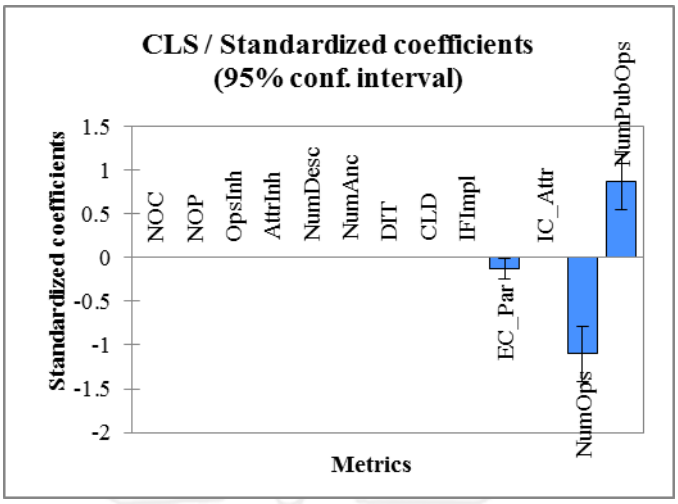
Rotation: Varimax (Kaiser normalization)		
Rotation matrix:		
Component	% of Variance	Cumulative %
D1	**54.473**	54.473
D2	38.051	92.524
Rotated Component Matrix:		
Metrics	D1	D2
NOC	**0.261**	0.007
NOP	**0.214**	0.023
OpsInh	**0.500**	0.001
AttrInh	**0.526**	0.000
NumDesc	**0.261**	0.007
NumAnc	**0.651**	0.000
DIT	**0.651**	0.000
CLD	**0.261**	0.007
IFImpl	**0.001**	0.000
Assoc_out	0.000	0.000
Assoc_In	0.000	0.000
NumAss_User	0.000	0.000
NumAss_Provider	0.000	0.000
EC_Par	**0.137**	0.001
IC_Par	0.000	**0.002**
EC_Attr	0.039	**0.722**
IC_Attr	**0.224**	0.031
MsgSent	0.042	**0.830**
MsgRecv	0.042	**0.872**
H	0.000	0.000
ConnComp	0.000	0.000
NumOps	**0.131**	0.094
NumPubOps	**0.098**	0.034
NumAttr	0.120	**0.169**
Nesting	0.012	**0.019**
Connectors	0.000	0.000
MsgSelf	0.035	**0.871**

จากตารางที่ 19 สามารถจัดกลุ่มตัววัดที่มีความสัมพันธ์กันได้ 2 องค์ประกอบ คือ องค์ประกอบที่ 1 ประกอบด้วยตัววัด 13 ตัววัด ได้แก่ NOC, NOP, OpsInh, AttrInh, NumDesc, NumAnc, DIT, CLD, IFImpl, EC_Par, IC_Attr, NumOps, NumPubOps และองค์ประกอบที่ 2 ประกอบด้วยตัววัด 7 ตัววัด ได้แก่ IC_Par, EC_Attr, MsgSent, MsgRecv, NumAttr, Nesting, MsgSelf ซึ่งจะเลือกองค์ประกอบที่ 1 มาสร้างโมเดลเนื่องจากองค์ประกอบสามารถอธิบายความผันแปรทั้งหมดได้ 54.473 % โดยนำตัววัดทั้ง 13 ตัวมาวิเคราะห์ความถดถอยเชิงซ้อน

ตารางที่ 20 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก

Metrics	Value	Standard error	t	Pr > |t|	Lower bound (95%)	Upper bound (95%)
NOC	0.000	0.000				
NOP	0.000	0.000				
OpsInh	0.000	0.000				
AttrInh	0.000	0.000				
NumDesc	0.000	0.000				
NumAnc	0.000	0.000				
DIT	0.000	0.000				
CLD	0.000	0.000				
IFImpl	0.000	0.000				
Assoc_out	**-0.127**	**0.058**	**-2.192**	**0.029**	**-0.241**	**-0.013**
Assoc_In	0.000	0.000				
NumAss_User	**-1.101**	**0.163**	**-6.772**	**< 0.0001**	**-1.421**	**-0.781**
NumAss_Provider	**0.862**	**0.163**	**5.303**	**< 0.0001**	**0.542**	**1.182**
EC_Par	0.000	0.000				
IC_Par	0.000	0.000				
EC_Attr	0.000	0.000				
IC_Attr	0.000	0.000				
MsgSent	0.000	0.000				
MsgRecv	0.000	0.000				
H	0.000	0.000				
ConnComp	0.000	0.000				
NumOps	0.000	0.000				
NumPubOps	**-0.127**	**0.058**	**-2.192**	**0.029**	**-0.241**	**-0.013**
NumAttr	0.000	0.000				
Nesting	**-1.101**	**0.163**	**-6.772**	**< 0.0001**	**-1.421**	**-0.781**
Connectors	0.000	0.000				
MsgSelf	0.000	0.000				

เมื่อนำข้อมูลของชุดโปรแกรมสำหรับสร้างโมเดลทั้งหมดที่สรุปข้อมูลทางสถิติทั้งหมดในตารางที่ 15 ซึ่งนำข้อมูลมาเพียง 13 ตัววัดที่ได้จากการวิเคราะห์ส่วนประกอบหลักในตารางที่ 19 มาวิเคราะห์ความถดถอยเชิงซ้อนจะได้ค่าสัมประสิทธิ์ความถดถอยของตัววัด ซึ่งจะปรากฏในโมเดลดังตารางที่ 20



รูปที่ 16 กราฟแสดงค่าสัมประสิทธิ์การถดถอยของตัววัด 13 ตัววัด

จากการวิเคราะห์ความถดถอยเชิงซ้อนจะได้โมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสจะได้ผลลัพธ์ของค่าสัมประสิทธิ์ของตัววัดดังรูปที่ 16 ซึ่งประกอบด้วยตัววัดแผนภาพ 3 ตัววัด ได้แก่ EC_Par คือ ตัววัดจำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นพารามิเตอร์ในคลาสอื่นและคลาสตัวเอง NumOps คือ ตัววัดจำนวนของโอเปอร์เรเตอร์หรือเมธอดของคลาส และ NumPubOps คือ ตัววัดจำนวนของโอเปอร์เรเตอร์หรือเมธอดของคลาสที่มีขอบเขตการเข้าถึงเป็นแบบพับลิค

$$CLS = 0.965242396 - (0.007123259)EC_Par - (0.008152249)NumOps + (0.007201201)NumPubOps \quad (13)$$

จากโมเดลที่ได้จะพบว่าความเสถียรเชิงตรรกะของคลาสใดๆ จะมีค่าเพิ่มขึ้นเมื่อมีจำนวนโอเปอร์เรเตอร์ของคลาสที่มีขอบเขตการเข้าถึงเป็นแบบพับลิคมาก ในขณะที่ความเสถียรเชิงตรรกะของคลาสจะมีค่าลดลงเมื่อจำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นพารามิเตอร์ในคลาสอื่นและคลาสตัวเองมาก หรือจำนวนของโอเปอร์เรเตอร์ของคลาสมาก

เมื่อแบ่งชุดโปรแกรมตามเวอร์ชันเพื่อสร้างโมเดลด้วยการวิเคราะห์ส่วนประกอบหลักของแต่ละเวอร์ชัน โดยจำนวนข้อมูลของแต่ละเวอร์ชันแสดงจำนวนดังตารางที่ 14 ประกอบด้วย 3 เวอร์ชัน ซึ่งขั้นตอนในการสร้างมีลักษณะเช่นเดียวกันกับการสร้างโมเดลในสมการที่ 13 โดยแสดงรูปแบบของโมเดลที่ได้ในแต่ละเวอร์ชันดังสมการที่ 14, 15 และ 16 ดังนี้

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (PCA) ในเวอร์ชันที่ 1

$$CLS = 0.885901366 + 0.0416481680 * IC_Attr - 0.011707289 * NumAttr \quad (14)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (PCA) ในเวอร์ชันที่ 2

$$CLS = 0.970267566 + 0.000555531 * OpsInh - 0.003129559 * AttrInh - 0.007512544 * EC_Par - 0.006298115 * IC_Attr - 0.008894980 * NumOps + 0.009569400 * NumPubOps \quad (15)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (PCA) ในเวอร์ชันที่ 3

$$CLS = 0.997750964 - 0.006909493 * IC_Attr - 0.000966608 * NumPubOps \quad (16)$$

จากโมเดลที่ได้จะพบว่าหากชุดข้อมูลที่ใช้ในการสร้างโมเดลต่างกันก็จะได้ตัววัดที่ปรากฏในโมเดลต่างกัน เนื่องจากค่าของตัววัดในแต่ละคลาสของแต่ละเวอร์ชันมีค่าที่แตกต่างกัน ดังนั้นการสร้างโมเดลในการประมาณค่าความเสถียรจึงต้องสร้างเพื่อให้มีความเหมาะสมกับการนำไปใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาส โดยเป็นโมเดลที่มีรูปแบบเฉพาะเจาะจงกับโปรแกรมในแต่ละเวอร์ชัน

4.1.3 การสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA)

ข้อมูลที่จะใช้ในการสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบนั้นไม่เหมือนกับโมเดลที่สร้างด้วยวิธการวิเคราะห์ความถดถอยเชิงซ้อนและวิธีการวิเคราะห์ส่วนประกอบหลัก เนื่องจากโมเดลนี้จะไม่ใช้ข้อมูลตัววัดแผนภาพทั้ง 27 ตัววัด แต่จะต้องวัดค่าตามตัววัดที่กำหนดขึ้นใหม่แล้วนำมาปรับค่าสัมประสิทธิ์ของตัววัดในโมเดลตามที่ระบุไว้ในสมการที่ 6 บทที่ 3 ซึ่งการสร้างโมเดลจะต้องมีการปรับโมเดล คือ การทำให้ค่าประมาณมีค่าเข้าใกล้ค่าจริงมากขึ้น เป็นวิธีที่ใช้ในงานวิจัยทั่วที่มีการสร้างโมเดล เช่น การประมาณค่าความพยายามด้วยการสร้างโมเดลการทำนายแบบอินคริเมนทัลที่พัฒนาจากโมเดลการทำนายแบบโกลบอล [14] เพื่อใช้ในการประมาณค่าความพยายาม (effort) เป็นต้น โดยกระบวนการปรับมี 2 ขั้นตอนคือ ขั้นตอนที่ 1 ทำให้มั่นใจว่าค่าที่นำไปสร้างโมเดลสอดคล้องกับข้อกำหนดของโมเดล และขั้นตอนที่ 2 คือปรับค่าสัมประสิทธิ์ของโมเดล อาจจะต้องดึงตัววัดบางตัวออก ปรับตัววัดที่มีอยู่แล้วหรือนำตัววัดตัวใหม่เข้าสู่โมเดล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 21 ตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดที่ใช้ในการปรับโมเดล

Increment	ID	Class Name	CLS	TA_x	TA_{xy}	TR_x	TR_{xy}	TM_x	TM_{xy}	TS_{xy}	TS'_{xy}
1st	1	EdasTexter	0.18518	50	17	1	0	17	5	0	0
	1	EdasTexter	0.53595	90	23	1	0	31	6	0	0
	2	ShowEdasTexterInfo	0.91754	7	4	1	0	4	3	0	0
	3	EdasPrinter	0.97507	3	1	1	0	2	1	0	0
+2nd	4	EdasPrinterPlus	0.96260	6	3	0	0	3	4	0	0
	5	EdasProxySetter	0.93288	11	6	1	0	6	8	0	0
	6	EdasUrlReader	0.97794	4	4	0	0	1	0	0	0
	7	ProxyObject	0.94247	3	1	0	0	9	1	0	0
	1	EdasTexter	0.74499	140	31	1	0	58	10	0	0
	2	ShowEdasTexterInfo	0.95468	7	4	1	0	4	3	0	0
	3	EdasPrinter	0.98173	4	2	1	0	2	1	0	0
	4	EdasPrinterPlus	0.97260	6	3	0	0	3	4	0	0
	5	EdasProxySetter	0.95082	11	6	1	0	6	8	0	0
	6	EdasUrlReader	0.98384	4	4	0	0	1	0	0	0
	7	ProxyObject	0.95785	3	1	0	0	8	0	0	0
	8	EdasColorChooser	0.95679	27	5	1	0	7	8	0	0
	9	EdasFinder	0.98911	0	0	0	0	5	0	0	0
+3rd	10	EdaSplash	0.98946	1	1	1	0	4	3	0	0
	11	EdasPrinterSetup	0.96557	17	7	1	0	8	3	0	0
	12	EdasUndoSetup	0.97295	12	5	1	0	4	5	0	0
	13	EIntField	0.99227	0	0	1	0	3	3	0	0
	14	ETextArea	0.96347	0	0	1	0	3	2	0	0
	15	EMouseListener	1	9	5	1	0	14	8	0	0
	16	PrinterObject	0.97611	4	2	0	0	10	0	0	0
	17	ShowJavaVersion	0.98314	7	4	1	0	4	3	0	0
	18	ShowShortKeys	0.98103	9	6	1	0	4	3	0	0
	19	ShowUserInfo	0.98314	7	4	1	0	4	3	0	0

การสร้างโมเดลจะต้องนำค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดวัดจากชุดโปรแกรมสำหรับสร้างโมเดล ซึ่งมีรายละเอียดตามตัวอย่างในตารางที่ 21 นำมาปรับค่าสัมประสิทธิ์ของตัววัดทั้ง 8 ตัววัดที่ได้จากการสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบจะได้รูปแบบของโมเดลตามสมการที่ 17

$$DS_x = 1 / (1 + DLRE_x)$$

$$DLRE_x = TA_x + TA_{xy} + TR_x + TR_{xy} + TM_x + TM_{xy} + TS_{xy} + TS'_{xy}$$

$$DLRE_x = 0.070539196 + (0.009749294)TA_x + (0.002898403)TA_{xy} + (-0.018457601)TR_x + (0.038095932)TR_{xy} \quad (17)$$

$$+(-0.003645325)TM_x + (-0.000432215)TM_{xy} + (0.007921985)TS_{xy} + (-0.006162119)TS'_{xy}$$

จากโมเดลข้างต้นที่ได้จะพบว่าความเสถียรเชิงตรรกะของคลาสใดๆ จะมีค่าเพิ่มขึ้น เมื่อผลกระทบเชิงตรรกะของการออกแบบของคลาสใดๆ มีค่าน้อย ซึ่งหมายถึงจำนวนรวมของผลกระทบ

ที่เกิดจากองค์ประกอบของการออกแบบมีค่าน้อย ได้แก่ จำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ จำนวนรวมของผลกระทบที่เกิดจากคลาส จำนวนรวมของผลกระทบที่เกิดจากเมธอด และจำนวนรวมของผลกระทบที่เกิดจากการรับส่งสาร

เมื่อแบ่งชุดโปรแกรมตามเวอร์ชันเพื่อสร้างโมเดลด้วยการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบในแต่ละเวอร์ชัน เชิงซ้อน โดยชุดโปรแกรมสำหรับสร้างโมเดลประกอบด้วย 3 เวอร์ชัน ซึ่งขั้นตอนในการสร้างมีลักษณะเช่นเดียวกันกับการสร้างโมเดลในสมการที 17 โดยแสดงรูปแบบของโมเดลที่ได้ในแต่ละเวอร์ชันดังสมการที่ 18, 19 และ 20 ดังนี้

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (DLREA) ในเวอร์ชันที่ 1

$$CLS = 0.853599028 - 0.009166459 * TA_x + 0.010518549 * TA_{xy} + 0.106503121 * TR_x + 0.995959720 * TR_{xy} + 0.016830598 * TM_x - 0.012180363 * TM_{xy} + 0.063219717 * TS_{xy} \quad (18)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (DLREA) ในเวอร์ชันที่ 2

$$CLS = 0.989576342 - 0.003175249 * TA_x + 0.000015210 * TA_{xy} - 0.0378242364 * TR_x - 0.145348081 * TR_{xy} - 0.000298922 * TM_x + 0.000263357 * TM_{xy} - 0.001878076 * TS_{yx} - 0.001544067 * TS'_{xy} \quad (19)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (DLREA) ในเวอร์ชันที่ 3

$$CLS = 1.007642850 - 0.001109574 * TA_x + 0.000786941 * TA_{xy} + 0.004527938 * TR_x - 0.000072784 * TM_x - 0.001908593 * TM_{xy} \quad (20)$$

จากโมเดลที่ได้จะพบว่าหากชุดข้อมูลที่ใช้ในการสร้างโมเดลต่างกันก็จะได้ตัววัดที่ปรากฏในโมเดลต่างกัน เนื่องจากค่าของตัววัดในแต่ละคลาสของแต่ละเวอร์ชันมีค่าที่แตกต่างกัน ดังนั้นการสร้างโมเดลในการประมาณค่าความเสถียรจึงต้องสร้างเพื่อให้มีความเหมาะสมกับการนำไปใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาส โดยเป็นโมเดลที่มีรูปแบบเฉพาะเจาะจงกับโปรแกรมในแต่ละเวอร์ชัน

4.1.4 การสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA))

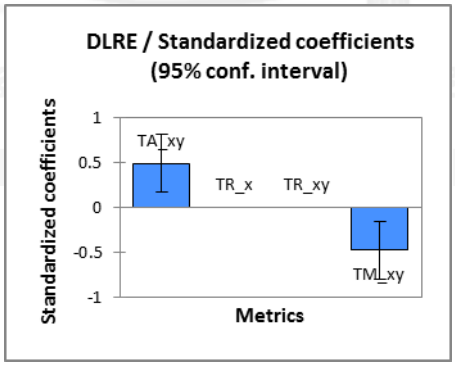
เมื่อกำหนดตัววัดที่จะใช้ในการสร้างโมเดลแล้ว จากองค์ความรู้ที่ได้จากการสร้างโมเดลด้วยวิธีการวิเคราะห์ส่วนประกอบหลักนั้นขั้นตอนต่อไปจึงนำตัววัดเข้าสู่กระบวนการวิเคราะห์ส่วนประกอบหลักซึ่งได้มีการทดลองแล้วว่าสามารถกำจัดตัวแปรหรือตัววัดที่ไม่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาสได้ จากนั้นจะต้องมีการปรับค่าสัมประสิทธิ์ของตัววัดเพื่อให้ช่วงค่าประมาณความเสถียรเชิงตรรกะของคลาสที่ได้มีค่าตั้งแต่ 0 ถึง 1 ซึ่งจะต้องวัดค่าตัววัดที่ปรากฏในโมเดลจากแผนภาพคลาสและแผนภาพลำดับของชุดโปรแกรมสำหรับสร้างโมเดลเป็นชุดโปรแกรมเดียวกันกับที่นำไปวัดค่าตัววัดแผนภาพในการสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อนและวิธีการวิเคราะห์ส่วนประกอบหลัก ซึ่งลักษณะข้อมูลที่ใช้วิเคราะห์ส่วนประกอบหลักเพื่อกำจัดตัววัดที่ไม่มีความสัมพันธ์กับความเสถียรเชิงตรรกะ และเพื่อใช้ในการปรับโมเดลประกอบด้วยค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัด แสดงตัวอย่างดังตารางที่ 22

ตารางที่ 22 ผลลัพธ์ที่ได้จากการวิเคราะห์ส่วนประกอบหลักของตัววัด 8 ตัววัด

Rotation: Varimax (Kaiser normalization)		
Rotation matrix:		
Component	% of Variance	Cumulative %
D1	**58.353**	58.353
D2	40.126	98.479
Rotated Component Matrix:		
Metrics	D1	D2
TA_x	0.064	**0.254**
TA_{xy}	**0.870**	0.051
TR_x	**0.834**	0.001
TR_{xy}	**0.012**	0.002
TM_x	0.058	**0.144**
TM_{xy}	**0.862**	0.009
TS_{xy}	0.039	**0.829**
TS'_{xy}	0.042	**0.804**

ตารางที่ 23 ค่าสัมประสิทธิ์การถดถอยของตัววัดในโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

Metrics	Value	Standard error	t	Pr > |t|	Lower bound (95%)	Upper bound (95%)
TA_{xy}	**0.012**	**0.004**	**3.006**	**0.003**	**0.004**	**0.020**
TR_x	0.000	0.000				
TR_{xy}	0.000	0.000				
TM_{xy}	**-0.002**	**0.001**	**-2.883**	**0.004**	**-0.003**	**-0.001**



รูปที่ 17 กราฟแสดงค่าสัมประสิทธิ์การถดถอยของตัววัด 4 ตัววัด

เมื่อนำข้อมูลของชุดโปรแกรมสำหรับสร้างโมเดลทั้งหมดที่มีตัวอย่างดังตารางที่ 21 ซึ่งนำข้อมูลมาเพียง 4 ตัววัดที่ได้จากการวิเคราะห์ส่วนประกอบหลักในตารางที่ 22 ได้แก่ TA_{xy}, TR_x, TR_{xy}

และ TM_{xy} นำมาวิเคราะห์ความถดถอยเชิงซ้อนจะได้ค่าสัมประสิทธิ์ความถดถอยของตัววัด ดังตารางที่ 23

$$DS_x = 1 / (1 + DLRE_x)$$

$$DLRE_x = a + b_0 TA_x + b_1 TA_{xy} + b_2 TR_x + b_3 TR_{xy} + b_4 TM_x + b_5 TM_{xy} + b_6 TS_{xy} + b_7 TS'_{xy} \quad (21)$$

$$DLRE_x = 0.070539196 + (0.009749294) TA_x + (0.002898403) TA_{xy} + (-0.018457601) TR_x + (0.038095932) TR_{xy}$$

$$+ (-0.003645325) TM_x + (-0.000432215) TM_{xy} + (0.007921985) TS_{xy} + (-0.006162119) TS'_{xy}$$

⇓ (PCA)

$$DLRE_x = a + b_i TA_{xy} + b_{i+1} TR_x + b_{i+2} TR_{xy} + b_{i+3} TM_{xy}$$

⇓ (MRA)

$$DLRE_x = 0.066721654 + (0.012248795) \sum_{y \in J_{xy}} TA_{xy} + (-0.001917440) \sum_{y \in J_{xy}} TM_{xy} \quad (22)$$

จากการวิเคราะห์ความถดถอยเชิงซ้อนจะได้โมเดลประมาณค่าความเสถียรเชิงตรรกะของการออกแบบซึ่งประกอบด้วย ตัววัด 2 ตัววัด ได้แก่ TA_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากแอททริบิวต์ที่ประกาศโดยคลาส *y* ถูกใช้โดยคลาส *x* และ TM_{xy} คือ จำนวนรวมของผลกระทบที่เกิดจากเมธอดของคลาส *y* โดยที่คลาส *x* สามารถเรียกใช้งาน ซึ่งเป็นตัววัดที่มีความสัมพันธ์กับค่าผลกระทบเชิงตรรกะของคลาส ค่าประมาณความเสถียรเชิงตรรกะของการออกแบบของคลาสใดๆ จะมีช่วงค่าเท่ากับ 0 ถึง 1 โดยรูปแบบสมการผลกระทบเชิงตรรกะของการออกแบบของคลาสใดๆ และค่าสัมประสิทธิ์ของตัววัด แสดงดังตารางที่ 20 และสมการที่ 22

เมื่อแบ่งชุดโปรแกรมตามเวอร์ชันเพื่อสร้างโมเดลด้วยการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ และใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดนำไปวิเคราะห์ความถดถอยเชิงซ้อน โดยชุดโปรแกรมสำหรับสร้างโมเดลประกอบด้วย 3 เวอร์ชัน ซึ่งขั้นตอนในการสร้างมีลักษณะเช่นเดียวกันกับการสร้างโมเดลในสมการที 22 โดยแสดงรูปแบบของโมเดลที่ได้ในแต่ละเวอร์ชันดังสมการที่ 23, 24 และ 25 ดังนี้

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (DLREA(PCA)) ในเวอร์ชันที่ 1

$$CLS = 0.921740629 - 0.003105296 * TA_x \quad (23)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (DLREA(PCA)) ในเวอร์ชันที่ 2

$$CLS = 0.950524609 - 0.006202401 * TR_x \quad (24)$$

โมเดลประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ (DLREA(PCA)) ในเวอร์ชันที่ 3

$$CLS = 1.01094724 - 0.001463411 * TA_x - 0.001703290 * \sum_{y \in J_{xy}} TM_{xy} \quad (25)$$

จากโมเดลที่ได้จะพบว่าหากชุดข้อมูลที่ใช้ในการสร้างโมเดลต่างกันก็จะได้ตัววัดที่ปรากฏในโมเดลต่างกัน เนื่องจากค่าของตัววัดในแต่ละคลาสของแต่ละเวอร์ชันมีค่าที่แตกต่างกัน ดังนั้นการสร้างโมเดลในการประมาณค่าความเสถียรจึงต้องสร้างเพื่อให้มีความเหมาะสมกับการนำไปใช้ในการ

ประมาณค่าความเสถียรเชิงตรรกะของคลาส โดยเป็นโมเดลที่มีรูปแบบเฉพาะเจาะจงกับโปรแกรมในแต่ละเวอร์ชัน

4.2 วัตถุประสงค์ของการประเมินผล

วัตถุประสงค์ของการประเมินผลโมเดลการประมาณค่าความเสถียรในงานวิจัยนี้คือ เพื่อให้ทราบว่าวิธีในการสร้างโมเดล และตัววัดที่ปรากฏในแต่ละโมเดลมีความสัมพันธ์อย่างไรกับค่าประมาณความเสถียร ซึ่งจะทำให้สามารถเปรียบเทียบได้ค่าประมาณที่ได้กับค่าจริงที่คำนวณได้จากซอร์ซโค้ด เพื่อนำไปทดสอบความถูกต้องและแม่นยำของค่าประมาณความเสถียรเชิงตรรกะของคลาสแต่ละคลาสในแต่ละอินคริเมนต์

4.3 การประเมินโมเดลการประมาณค่าความเสถียร

การประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสจะใช้ชุดโปรแกรมทดสอบนำมาวัดค่าความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ด และแปลงเป็นแผนภาพคลาสและแผนภาพลำดับ ซึ่งมีทั้งหมด 15 โปรแกรมแต่ละโปรแกรมจะมี 2 เวอร์ชัน ซึ่งเวอร์ชันแรกจะหมายถึงอินคริเมนต์ก่อนหน้า (Previous increment) และเวอร์ชันที่สองจะหมายถึงโครงสร้างการออกแบบใหม่หรือโมเดลการออกแบบที่เพิ่มอินคริเมนต์ปัจจุบัน (Current increment) เข้ากับอินคริเมนต์ก่อนหน้า จากนั้นจะหาค่าตัววัดหรือตัวแปรต่างๆที่ปรากฏในโมเดลแล้วนำมาคำนวณค่าประมาณตามโมเดลนั้นๆ แล้วนำมาค่าจริงที่คำนวณได้จากซอร์ซโค้ดมาเปรียบเทียบกับค่าประมาณที่คำนวณได้จากแผนภาพ เพื่อหาความคลาดเคลื่อนและประเมินผลโมเดล

เนื่องจากข้อมูลที่ใช้ประเมินผลโมเดลมีจำนวนมากจึงยกตัวอย่างมาเพียง 1 โปรแกรมเพื่อให้ทราบรายละเอียดของวิธีการประเมิลผลโมเดล โปรแกรมนี้เป็นโปรแกรมสำหรับการจัดการข้อความ โดยจำนวนคลาสของอินคริเมนต์แรกมีทั้งหมด 22 คลาส และจำนวนคลาสของโครงสร้างการออกแบบใหม่หรือจำนวนรวมของคลาสเมื่อเพิ่มอินคริเมนต์ที่สองมีทั้งหมด 67 คลาส โดยจะแสดงรายละเอียดผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของแต่ละโมเดลดังนี้

4.3.1 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน

โมเดลนี้สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน ซึ่งตัววัดในโมเดลประกอบด้วย 6 ตัววัด คือ EC_Par, EC_Attr, MsgSent, MsgRecv, NumAttr และ Nesting การประเมินจะต้องใช้ค่าจริงที่คำนวณได้จากซอร์ซโค้ดและค่าประมาณที่ได้จากการนำค่าตัววัดแผนภาพมาคำนวณตามโมเดลแล้วนำมาหาค่าความคลาดเคลื่อน ซึ่งแสดงตัวอย่างดังตารางที่ 24

ตารางที่ 24 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน

Class No.	Class Name	ค่าจริงจากซอร์ซโค้ด		ค่าประมาณจากแผนภาพ		MRE	
		Previous Increment	+ Current Increment	Previous Increment	+ Current Increment	Previous Increment	+ Current Increment
1	ExtraReturnsRemover	0.9708	0.9023	0.9628	0.9492	0.0082	0.0519
2	HTMLReplacer	0.9436	0.9023	0.9628	0.9628	0.0204	0.0670
3	LetterPulse	0.8879	0.9205	0.9519	0.9519	0.0721	0.0342
4	LetterPulseDialog	1.0000	0.9979	0.9437	0.9437	0.0563	0.0542
5	NonPrintingChars	1.0000	0.9979	0.9628	0.9628	0.0372	0.0351
6	Search	0.8149	0.9979	0.9546	0.9546	0.1715	0.0433
7	FindDialog	0.9297	0.9979	0.9110	0.9001	0.0201	0.0980
8	LibTTx	1.0000	1.0000	0.9628	0.9601	0.0372	0.0399
9	EndsWithFileFilter	1.0000	1.0000	0.9601	0.9601	0.0399	0.0399
10	PlugIn	1.0000	0.9837	0.9246	0.9246	0.0754	0.0591
11	PlugInAction	1.0000	0.9039	0.9628	0.9628	0.0372	0.0652
12	PlugInEvent	1.0000	0.9076	0.9546	0.9546	0.0454	0.0519
13	PlugInOutcome	1.0000	0.9979	0.9519	0.9464	0.0481	0.0515
14	PlugInWindow	0.8792	0.9979	0.9546	0.9546	0.0857	0.0433
15	Prefs	0.6218	0.9979	0.8074	0.8102	0.2986	0.1881
16	PrintPad	1.0000	0.9979	0.9546	0.9546	0.0454	0.0433
17	PrintPadPreview	1.0000	0.9979	0.9601	0.9601	0.0399	0.0379
18	StoppableThread	0.9923	0.9619	0.9601	0.9601	0.0325	0.0019
19	TextPad	0.8950	0.9545	0.9274	0.9192	0.0362	0.0370
20	TextTrix	0.9893	0.9942	0.8701	0.7993	0.1204	0.1961
21	ExtensionFileFilter	0.9979	0.9125	0.9573	0.9573	0.0406	0.0492
22	Tools	0.6800	1.0000	0.9628	0.9628	0.4159	0.0372
23	ExtraReturnsRemoverDialog	-	0.9923	-	0.9328	-	0.0600
24	SongSheet	-	0.9889	-	0.9328	-	0.0568
25	BrowseFiles	-	0.9966	-	0.9410	-	0.0558
26	BrowseFilesFromTextPad	-	0.9760	-	0.9628	-	0.0136
27	BrowseSingleFile	-	0.8790	-	0.9628	-	0.0953
28	DocumentSearchFilter	-	0.8682	-	0.9601	-	0.1058
29	LineDancePanel	-	1.0000	-	0.9464	-	0.0536
30	LineDanceTable	-	0.9850	-	0.9628	-	0.0225
...	...	...	...	...	...	...	...
67	GetoptDemo	-	1.0000	-	0.9628	-	0.0372

4.3.2 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ส่วนประกอบหลัก

ในโมเดลนี้ประกอบด้วยตัววัดแผนภาพ 3 ตัววัดที่ทำการวัดค่าจากแผนภาพ ได้แก่ EC_Par NumOps และ NumPubOps ซึ่งเมื่อนำค่าจริงที่คำนวณได้จากซอร์ซโค้ดมากับค่าประมาณที่คำนวณได้จากตัววัดแผนภาพมาหาค่าความคลาดเคลื่อน แสดงผลการประเมินความคลาดเคลื่อนตัวอย่างดังตารางที่ 25

ตารางที่ 25 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ส่วนประกอบหลัก

Class No.	Class Name	ค่าจริงจากซอร์ซโค้ด		ค่าประมาณจากแผนภาพ		MRE	
		Previous Increment	+ Current Increment	Previous Increment	+ Current Increment	Previous Increment	+ Current Increment
1	ExtraReturnsRemover	0.9708	0.9023	0.9595	0.9495	0.0116	0.0522
2	HTMLReplacer	0.9436	0.9023	0.9595	0.9595	0.0169	0.0634
3	LetterPulse	0.8879	0.9205	0.9586	0.9576	0.0797	0.0404
4	LetterPulseDialog	1.0000	0.9979	0.9605	0.9605	0.0395	0.0374
5	NonPrintingChars	1.0000	0.9979	0.9605	0.9605	0.0395	0.0374
6	Search	0.8149	0.9979	0.9500	0.9428	0.1659	0.0551
7	FindDialog	0.9297	0.9979	0.9538	0.9481	0.0260	0.0498
8	LibTTx	1.0000	1.0000	0.9396	0.9248	0.0604	0.0753
9	EndsWithFileFilter	1.0000	1.0000	0.9633	0.9633	0.0367	0.0367
10	PlugIn	1.0000	0.9837	0.9244	0.9329	0.0757	0.0517
11	PlugInAction	1.0000	0.9039	0.9643	0.9643	0.0357	0.0668
12	PlugInEvent	1.0000	0.9076	0.9614	0.9614	0.0386	0.0594
13	PlugInOutcome	1.0000	0.9979	0.9538	0.9491	0.0462	0.0489
14	PlugInWindow	0.8792	0.9979	0.9443	0.9529	0.0740	0.0451
15	Prefs	0.6218	0.9979	0.9143	0.9023	0.4705	0.0957
16	PrintPad	1.0000	0.9979	0.9576	0.9576	0.0424	0.0403
17	PrintPadPreview	1.0000	0.9979	0.9643	0.9643	0.0357	0.0336
18	StoppableThread	0.9923	0.9619	0.9624	0.9624	0.0302	0.0005
19	TextPad	0.8950	0.9545	0.8769	0.8479	0.0201	0.1118
20	TextTrix	0.9893	0.9942	0.8498	0.8250	0.1410	0.1701
21	ExtensionFileFilter	0.9979	0.9125	0.9614	0.9605	0.0365	0.0526
22	Tools	0.6800	1.0000	0.9605	0.9605	0.4125	0.0395
23	ExtraReturnsRemoverDialog	-	0.9923	-	0.9595	-	0.0330
24	SongSheet	-	0.9889	-	0.9309	-	0.0587
25	BrowseFiles	-	0.9966	-	0.9453	-	0.0515
26	BrowseFilesFromTextPad	-	0.9760	-	0.9605	-	0.0159
27	BrowseSingleFile	-	0.8790	-	0.9605	-	0.0927
28	DocumentSearchFilter	-	0.8682	-	0.9624	-	0.1084
29	LineDancePanel	-	1.0000	-	0.9586	-	0.0414
30	LineDanceTable	-	0.9850	-	0.9643	-	0.0210
...	...	...	...	...	...	...	...
67	GetoptDemo	-	1.0000	-	0.9643	-	0.0357

4.3.3 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

โมเดลนี้สร้างด้วยวิธีการวิเคราะห์เชิงตรรกะของการออกแบบ การประเมินจะต้องใช้ค่าจริงที่คำนวณได้จากซอร์ซโค้ดและค่าประมาณที่ได้จากการคำนวณหาค่าจำนวนรวมของผลกระทบจากองค์ประกอบต่างๆ ตามตัววัดในโมเดลทั้งหมด 8 ตัววัด แล้วนำมาหาค่าความคลาดเคลื่อน ซึ่งแสดงตัวอย่างดังตารางที่ 26

ตารางที่ 26 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

Class No.	Class Name	ค่าจริงจากซอร์ซโค้ด		ค่าประมาณจากแผนภาพ		MRE	
		Previous Increment	+ Current Increment	Previous Increment	+ Current Increment	Previous Increment	+ Current Increment
1	ExtraReturnsRemover	0.9708	0.9023	0.9652	0.9413	0.0058	0.0432
2	HTMLReplacer	0.9436	0.9023	0.9652	0.9576	0.0229	0.0613
3	LetterPulse	0.8879	0.9205	0.9564	0.9442	0.0772	0.0258
4	LetterPulseDialog	1.0000	0.9979	0.9542	0.9542	0.0458	0.0437
5	NonPrintingChars	1.0000	0.9979	0.9701	0.9625	0.0299	0.0354
6	Search	0.8149	0.9979	0.9144	0.9069	0.1221	0.0912
7	FindDialog	0.9297	0.9979	0.9438	0.9225	0.0152	0.0755
8	LibTTx	1.0000	1.0000	0.8808	0.8493	0.1192	0.1507
9	EndsWithFileFilter	1.0000	1.0000	0.9982	0.9982	0.0018	0.0018
10	PlugIn	1.0000	0.9837	0.8789	0.9174	0.1211	0.0675
11	PlugInAction	1.0000	0.9039	1.0000	1.0000	0.0000	0.1064
12	PlugInEvent	1.0000	0.9076	0.9160	0.9160	0.0840	0.0093
13	PlugInOutcome	1.0000	0.9979	0.9537	0.9338	0.0463	0.0642
14	PlugInWindow	0.8792	0.9979	0.8952	0.9280	0.0181	0.0700
15	Prefs	0.6218	0.9979	0.8553	0.8343	0.3756	0.1639
16	PrintPad	1.0000	0.9979	0.9706	0.9706	0.0294	0.0273
17	PrintPadPreview	1.0000	0.9979	1.0000	0.9654	0.0000	0.0325
18	StoppableThread	0.9923	0.9619	0.9564	0.9564	0.0362	0.0057
19	TextPad	0.8950	0.9545	0.8632	0.7011	0.0355	0.2655
20	TextTrix	0.9893	0.9942	0.7218	0.6781	0.2704	0.3179
21	ExtensionFileFilter	0.9979	0.9125	0.9520	0.9472	0.0460	0.0381
22	Tools	0.6800	1.0000	0.9815	0.9815	0.4434	0.0185
23	ExtraReturnsRemoverDialog	-	0.9923	-	0.9561	-	0.0365
24	SongSheet	-	0.9889	-	0.8973	-	0.0927
25	BrowseFiles	-	0.9966	-	0.8982	-	0.0988
26	BrowseFilesFromTextPad	-	0.9760	-	0.9562	-	0.0203
27	BrowseSingleFile	-	0.8790	-	0.9562	-	0.0879
28	DocumentSearchFilter	-	0.8682	-	0.9598	-	0.1054
29	LineDancePanel	-	1.0000	-	0.9443	-	0.0557
30	LineDanceTable	-	0.9850	-	1.0000	-	0.0153
..	...	...	...	...	...	...	...
67	GetoptDemo	-	1.0000	-	1.0000	-	0.0000

4.3.4 การประเมินผลโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด

โมเดลนี้สร้างด้วยวิธีการวิเคราะห์เชิงตรรกะของการออกแบบซึ่งคัดเลือกตัววัดที่จะนำมาสร้างโมเดลด้วยการวิเคราะห์ส่วนประกอบหลัก การประเมินจะต้องใช้ค่าจริงที่คำนวณได้จากซอร์ซโค้ดและค่าประมาณที่ได้จากการคำนวณหาค่าจำนวนรวมของผลกระทบจากองค์ประกอบต่างๆ ตามตัววัดในโมเดล แล้วนำมาหาค่าความคลาดเคลื่อน ซึ่งแสดงตัวอย่างดังตารางที่ 27

ตารางที่ 27 ตัวอย่างผลการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างโดยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบโดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด

Class No.	Class Name	ค่าจริงจากซอร์ซโค้ด		ค่าประมาณจากแผนภาพ		MRE	
		Previous Increment	+ Current Increment	Previous Increment	+ Current Increment	Previous Increment	+ Current Increment
1	ExtraReturnsRemover	0.9708	0.9023	0.8739	0.9194	0.0998	0.0189
2	HTMLReplacer	0.9436	0.9023	0.8739	0.8610	0.0738	0.0459
3	LetterPulse	0.8879	0.9205	0.9220	0.9091	0.0384	0.0123
4	LetterPulseDialog	1.0000	0.9979	0.8942	0.8942	0.1059	0.1039
5	NonPrintingChars	1.0000	0.9979	0.8725	0.8595	0.1275	0.1386
6	Search	0.8149	0.9979	0.9589	0.9416	0.1768	0.0564
7	FindDialog	0.9297	0.9979	0.9152	0.9145	0.0156	0.0835
8	LibTTx	1.0000	1.0000	0.9853	0.9864	0.0147	0.0136
9	EndsWithFileFilter	1.0000	1.0000	0.9301	0.9301	0.0699	0.0699
10	PlugIn	1.0000	0.9837	0.8423	0.8302	0.1577	0.1560
11	PlugInAction	1.0000	0.9039	0.9391	0.9391	0.0609	0.0390
12	PlugInEvent	1.0000	0.9076	0.7214	0.7214	0.2786	0.2052
13	PlugInOutcome	1.0000	0.9979	0.9470	0.9557	0.0530	0.0423
14	PlugInWindow	0.8792	0.9979	0.8693	0.8440	0.0113	0.1542
15	Prefs	0.6218	0.9979	0.8870	0.8931	0.4266	0.1050
16	PrintPad	1.0000	0.9979	0.9295	0.9295	0.0705	0.0685
17	PrintPadPreview	1.0000	0.9979	1.0000	0.9285	0.0000	0.0695
18	StoppableThread	0.9923	0.9619	0.9425	0.9425	0.0502	0.0202
19	TextPad	0.8950	0.9545	1.0000	0.9404	0.1174	0.0148
20	TextTrix	0.9893	0.9942	0.8582	0.8411	0.1325	0.1539
21	ExtensionFileFilter	0.9979	0.9125	0.9229	0.9245	0.0751	0.0132
22	Tools	0.6800	1.0000	0.9460	0.9460	0.3912	0.0541
23	ExtraReturnsRemoverDialog	-	0.9923	-	0.8860	-	0.1072
24	SongSheet	-	0.9889	-	0.9376	-	0.0520
25	BrowseFiles	-	0.9966	-	0.9420	-	0.0548
26	BrowseFilesFromTextPad	-	0.9760	-	0.8970	-	0.0810
27	BrowseSingleFile	-	0.8790	-	0.8970	-	0.0205
28	DocumentSearchFilter	-	0.8682	-	0.9494	-	0.0935
29	LineDancePanel	-	1.0000	-	0.9174	-	0.0826
30	LineDanceTable	-	0.9850	-	0.8814	-	0.1052
..	...	...	...	...	...	...	...
67	GetoptDemo	-	1.0000	-	0.9391	-	0.0609

4.4 การเปรียบเทียบการประเมินผลโมเดลการประมาณค่าความเสถียร

จากตัวอย่างผลการประเมินโมเดลทั้ง 4 โมเดลนั้น ในงานวิจัยนี้ได้ทำการประเมินผลโดยชุดโปรแกรมทดสอบซึ่งมีทั้งหมด 15 โปรแกรมรายละเอียดระบุดังตารางที่ 11 โดยนำค่าความคลาดเคลื่อนที่ได้ทั้งหมดมาคำนวณตามสมการการทำนายระดับที่ 0.25 ซึ่งกำหนดตามมาตรฐานระดับที่

0.25 จะต้องมากกว่าหรือเท่ากับ 75 เปอร์เซ็นต์ของค่าประมาณทั้งหมดจะต้องคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์

ตารางที่ 28 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียร

ตัววัดคุณภาพ	วิธีการในการสร้างโมเดล			
	MRA	PCA	DLREA	DLREA(PCA)
MMRE	0.0780	0.0575	0.1007	0.0858
PRED(0.25)	0.9594	0.9763	0.8571	0.9512

จากตารางที่ 28 พบว่าค่าประมาณความเสถียรเชิงตรรกะของคลาสทั้ง 3 โมเดลมีเกณฑ์ความความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์ผ่านมาตรฐานซึ่งให้ค่าที่สูงกว่า 75 เปอร์เซ็นต์ของค่าประมาณทั้งหมด โดยโมเดลที่ได้ค่าประมาณใกล้เคียงค่าจริงมากที่สุดคือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก รองลงมาคือโมเดลที่สร้างด้วยวิธการวิเคราะห์ความถดถอยเชิงซ้อน และอันดับสุดท้ายคือโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

การประมาณค่าความเสถียรเชิงตรรกะของคลาสด้วยโมเดลที่สร้างด้วยวิธีวิเคราะห์ความถดถอยเชิงซ้อน ได้ผลลัพธ์จากการประมาณค่ามีค่าเฉลี่ยขนาดความผิดพลาดสัมพัทธ์ เท่ากับ 0.0780 หมายความว่าค่าเฉลี่ยของการประมาณค่ามีขนาดความผิดพลาด 7.8% จากค่าจริง และการทำนายที่ระดับ 0.25 เท่ากับ 0.9594 หมายความว่า 95.94% ของค่าที่ได้จากการทำนายมีความผิดพลาดไม่เกิน 25% ของค่าจริง

การประมาณค่าความเสถียรเชิงตรรกะของคลาสด้วยโมเดลที่สร้างด้วยวิธีวิเคราะห์ส่วนประกอบหลัก ได้ผลลัพธ์จากการประมาณค่ามีค่าเฉลี่ยขนาดความผิดพลาดสัมพัทธ์ เท่ากับ 0.0575 หมายความว่าค่าเฉลี่ยของการประมาณค่ามีขนาดความผิดพลาด 5.75% จากค่าจริง และการทำนายที่ระดับ 0.25 เท่ากับ 0.9763 หมายความว่า 97.63% ของค่าที่ได้จากการทำนายมีความผิดพลาดไม่เกิน 25% ของค่าจริง

การประมาณค่าความเสถียรเชิงตรรกะของคลาสด้วยโมเดลที่สร้างด้วยวิธีวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการเลือกตัววัด ได้ผลลัพธ์จากการประมาณค่ามีค่าเฉลี่ยขนาดความผิดพลาดสัมพัทธ์ เท่ากับ 0.0858 หมายความว่าค่าเฉลี่ยของการประมาณค่ามีขนาดความผิดพลาด 8.58% จากค่าจริง และการทำนายที่ระดับ 0.25 เท่ากับ 0.9512 หมายความว่า 95.12% ของค่าที่ได้จากการทำนายมีความผิดพลาดไม่เกิน 25% ของค่าจริง ซึ่งผลการประเมินโมเดลดีกว่าการประมาณค่าความเสถียรเชิงตรรกะของคลาสด้วยโมเดลที่สร้างด้วยวิธีวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบโดยไม่ใช้การวิเคราะห์ส่วนประกอบหลักในการเลือกตัววัด ซึ่งได้ผลลัพธ์จากการประมาณค่ามีค่าเฉลี่ยขนาดความผิดพลาดสัมพัทธ์ เท่ากับ 0.1007 หมายความว่าค่าเฉลี่ยของการประมาณค่ามีขนาดความผิดพลาด 10.07% จากค่าจริง และการ

ทำนายที่ระดับ 0.25 เท่ากับ 0.8571 หมายความว่า 85.71% ของค่าที่ได้จากการทำนายมีความผิดพลาดไม่เกิน 25% ของค่าจริง

ตารางที่ 29 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียร 4 โมเดล ด้วยชุดโปรแกรมของแต่ละเวอร์ชัน

มาตรวัดประสิทธิภาพ	ผลการประเมินผลโมเดลประมาณค่าความเสถียร											
	Increment 1st				+ Increment 2nd				+ Increment 3rd			
	MRA	PCA	DLREA	DLREA (PCA)	MRA	PCA	DLREA	DLREA (PCA)	MRA	PCA	DLREA	DLREA (PCA)
MMRE	0.1251	0.1049	0.2812	0.1138	0.0590	0.0569	0.0722	0.0619	0.0978	0.0301	0.1133	0.1108
PRED (0.25)	0.9618	0.9709	0.8836	0.9309	0.9797	0.9882	0.9332	0.9765	0.9185	0.9806	0.9103	0.9174

จากตารางที่ 29 แสดงการเปรียบเทียบการประเมินโมเดลการประมาณค่าความเสถียรของ 4 โมเดล เบื้องต้นจากการนำไปประเมินผลด้วยชุดโปรแกรมสำหรับประเมินผลดังตารางที่ 28 และในตารางที่ 29 นำมาประเมินผลด้วยชุดโปรแกรมของแต่ละเวอร์ชันซึ่งมีทั้งหมด 3 เวอร์ชัน โดยโมเดลที่มีความแม่นยำในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล อันดับที่ 1 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) อันดับที่ 2 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) อันดับที่ 3 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) และอันดับที่ 4 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) โดยผลการประเมินที่ได้มีความสอดคล้องกันในทุกเวอร์ชัน ดังจะเห็นได้จากค่า MMRE ซึ่งหมายถึงค่าเฉลี่ยของการประมาณค่ามีขนาดความผิดพลาดที่คิดเป็นเปอร์เซ็นต์จากค่าจริง และ ค่า PRED(0.25) ซึ่งมีความหมายคือจำนวนเปอร์เซ็นต์ของค่าที่ได้จากการทำนายมีความผิดพลาดไม่เกิน 25% ของค่าจริง

ตารางที่ 30 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียรในแต่ละเวอร์ชัน

มาตรวัดประสิทธิภาพ	ผลการประเมินผลโมเดลประมาณค่าความเสถียร											
	Increment 1st				+ Increment 2nd				+ Increment 3rd			
	MRA	PCA	DLREA	DLREA (PCA)	MRA	PCA	DLREA	DLREA (PCA)	MRA	PCA	DLREA	DLREA (PCA)
MMRE	0.0913	0.0664	0.1484	0.1025	0.0730	0.0539	0.0911	0.0818	0.0801	0.0595	0.0922	0.0841
PRED (0.25)	0.9655	0.9727	0.7709	0.9600	0.9695	0.9813	0.8755	0.9653	0.9368	0.9684	0.8705	0.9195

จากตารางที่ 30 แสดงการเปรียบเทียบการประเมินโมเดลการประมาณค่าความเสถียรในแต่ละอินคริเมนต์ที่ได้ทำการศึกษาเพิ่มเติม เมื่อใช้ชุดโปรแกรมสำหรับสร้างโมเดลแบ่งออกตามเวอร์ชัน ซึ่งโมเดลที่มีความแม่นยำในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล อันดับที่ 1 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) ดังจะ

เห็นได้ว่า PRED(0.25) ของทั้ง 3 เวอร์ชันนั้นมีค่าสูงสุดจากวิธีการสร้างโมเดลทั้งหมด 4 วิธี อันดับที่ 2 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) อันดับที่ 3 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ โดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) และอันดับที่ 4 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) ซึ่งผลการประเมินมีความสอดคล้องกันในทุกเวอร์ชัน ดังจะเห็นได้จากค่า MMRE ซึ่งหมายถึงค่าเฉลี่ยของการประมาณค่ามีขนาดความผิดพลาดที่คิดเป็นเปอร์เซ็นต์จากค่าจริง และ ค่า PRED(0.25) ซึ่งมีความหมายคือจำนวนเปอร์เซ็นต์ของค่าที่ได้จากการทำนายมีความผิดพลาดไม่เกิน 25% ของค่าจริง

ตารางที่ 31 เปรียบเทียบผลการประเมินโมเดลการประมาณค่าความเสถียรที่ระดับ PRED(0.25) แบ่งตามวิธีการสร้างในแต่ละเวอร์ชัน

ผลการประเมินโมเดล	MRA PRED(0.25)	PCA PRED(0.25)	DLREA PRED(0.25)	DLREA(PCA) PRED(0.25)
เวอร์ชัน 1 (Increment 1st)	0.9655	**0.9727**	0.7709	0.9600
เวอร์ชัน 2 (+Increment 2nd)	0.9695	**0.9813**	0.8755	0.9653
เวอร์ชัน 3 (+Increment 3rd)	0.9368	**0.9684**	0.8705	0.9195
ชุดโปรแกรมสำหรับประเมินผลโมเดล	0.9594	**0.9763**	0.8571	0.9512

จากตารางที่ 31 เป็นตารางสรุปการเปรียบเทียบการประเมินโมเดลการประมาณค่าความเสถียรในแต่ละอินคริเมนต์จากตารางที่ 29 และตารางที่ 30 ซึ่งผลการประเมินด้วยการทำนายที่ระดับ 0.25 หรือ PRED(0.25) มีความสอดคล้องกันในทุกเวอร์ชัน โดยโมเดลที่มีความแม่นยำในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล อันดับที่ 1 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) อันดับที่ 2 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) อันดับที่ 3 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ โดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) และอันดับที่ 4 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA)

จากผลการประเมินโมเดลข้างต้นสามารถนำมาวิเคราะห์เหตุปัจจัยที่มาให้โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลักมีความแม่นยำสูงสุดนั้นมาจากการเลือกตัววัดแผนภาพจากทั้งหมด 27 ตัววัดโดยสกัดตัววัดที่มีความสัมพันธ์กันและตัดตัววัดที่ไม่มีความสัมพันธ์ออกไป จากงานวิจัยนี้จะได้ตัววัดเพียง 13 ตัววัดแล้วจึงนำมาสร้างโมเดลด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน ดังจะเห็นได้จากหากใช้วิธีการวิเคราะห์ความถดถอยเชิงซ้อนโดยใช้ตัววัดทั้ง 27 ตัวนั้นจะได้ผลของค่าประมาณที่ใกล้เคียงค่าจริงน้อยกว่า ส่วนโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบนั้น เป็นโมเดลที่กำหนดตัววัดขึ้นมาใหม่ ซึ่งทำการเปรียบเทียบระหว่างเมื่อใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดและไม่มีการคัดเลือกตัววัด ซึ่งหากเพียงแต่นำตัววัดทั้งหมด 8 ตัววัดที่กำหนดขึ้นมาหาค่าสัมประสิทธิ์ของแต่ละตัววัด จะได้ค่าประมาณความเสถียรของการออกแบบของคลาสใดๆ จากโมเดลประมาณค่าความเสถียรเชิงตรรกะของการออกกแบบของ

คลาสที่มีความแม่นยำน้อยกว่า การใช้วิธีการวิเคราะห์ส่วนประกอบหลัก เพื่อคัดเลือกตัววัดที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาสเพื่อนำมาสร้างโมเดล และตัดตัววัดที่ไม่มีความสัมพันธ์ออกไปจะทำให้ได้ค่าประมาณที่มีความแม่นยำมากขึ้น ซึ่งวิธีการดังกล่าวอยู่ในขั้นตอนของการสร้างโมเดลทั่วไปที่จะต้องมีการสร้างข้อกำหนดโมเดลและปรับโมเดล โดยการตัดตัววัดหรือเพิ่มตัววัดอื่นที่มีความสัมพันธ์กับข้อกำหนดของโมเดล ซึ่งหากมีการนำโมเดลทั้ง 4 โมเดลดังกล่าวเข้าสู่กระบวนการพัฒนาหรือปรับโมเดลต่อไป อาจเป็นไปได้ว่าค่าประมาณที่ได้จะมีค่าใกล้เคียงค่าจริงมากยิ่งขึ้น

จากงานวิจัยนี้สามารถเลือกโมเดลที่ดีที่สุดในการนำไปประมาณค่าความเสถียรเชิงตรรกะของคลาสในขั้นตอนการออกแบบของซอฟต์แวร์ที่พัฒนาแบบอินคริเมนทัล ซึ่งก็คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โดยการคำนวณค่าจะใช้ค่าจากตัววัดแผนภาพ แต่หากต้องการใช้โมเดลที่มีข้อจำกัดของตัววัดที่สอดคล้องกับนิยามของความเสถียร โมเดลที่เหมาะสมก็คือโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ โดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) ซึ่งจากวิธีการในการสร้างโมเดลทั้งสองนี้ทำให้สามารถสรุปได้ว่า การใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัดที่จะนำมาสร้างโมเดลนั้นจะสามารถทำให้ได้โมเดลที่มีประสิทธิภาพในการนำไปประมาณค่าความเสถียรมากกว่าโมเดลที่ไม่ได้มีการคัดเลือกตัววัดด้วยการวิเคราะห์ส่วนประกอบหลัก

การเลือกวิธีการในการสร้างโมเดลและตัววัดเป็นปัจจัยสำคัญที่จะทำให้โมเดลที่ได้มีคุณภาพ งานวิจัยนี้ได้นำเสนอโมเดลทั้ง 4 เพื่อศึกษาการสร้างโมเดลด้วยวิธีการที่แตกต่างกันและทำการเปรียบเทียบโมเดลด้วยผลการประเมินโมเดล ซึ่งงานวิจัยนี้ได้พัฒนาเครื่องมือที่ผู้ใช้สามารถเลือกใช้โมเดลที่ต้องการซึ่งผลของค่าประมาณที่ได้จะแตกต่างกัน ดังตารางที่ 32 เป็นตัวอย่างของโปรแกรมเดียวกันกับที่แสดงผลการประเมินโมเดลทั้ง 4 โมเดลข้างต้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 32 เปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาสของโมเดลทั้ง 4 โมเดล

No.	Class Name	Previous Increment				+ Current Increment				Change of CLS			
		MRA	PCA	DLREA	DLREA (PCA)	MRA	PCA	DLREA	DLREA (PCA)	MRA	PCA	DLREA	DLREA (PCA)
1	ExtraReturnsRemover	0.9628	0.9595	0.9652	0.8739	0.9492	0.9495	0.9194	0.9194	-0.0136	-0.0101	0.0454	0.0454
2	HTMLReplacer	0.9628	0.9595	0.9652	0.8739	0.9628	0.9595	0.8610	0.8610	0.0000	0.0000	-0.0130	0.0454
3	LetterPulse	0.9519	0.9586	0.9564	0.9220	0.9519	0.9576	0.9091	0.9091	0.0000	-0.0010	-0.0129	-0.0130
4	LetterPulseDialog	0.9437	0.9605	0.9542	0.8942	0.9437	0.9605	0.8942	0.8942	0.0000	0.0000	0.0000	-0.0129
5	NonPrintingChars	0.9628	0.9605	0.9701	0.8725	0.9628	0.9605	0.8595	0.8595	0.0000	0.0000	-0.0129	0.0000
6	Search	0.9546	0.9500	0.9144	0.9589	0.9546	0.9428	0.9416	0.9416	0.0000	-0.0072	-0.0173	-0.0129
7	FindDialog	0.9110	0.9538	0.9438	0.9152	0.9001	0.9481	0.9145	0.9145	-0.0109	-0.0057	-0.0006	-0.0173
8	LibTTx	0.9628	0.9396	0.8808	0.9853	0.9601	0.9248	0.9864	0.9864	-0.0027	-0.0148	0.0011	-0.0006
9	EndsWithFileFilter	0.9601	0.9633	0.9982	0.9301	0.9601	0.9633	0.9301	0.9301	0.0000	0.0000	0.0000	0.0011
10	PlugIn	0.9246	0.9244	0.8789	0.8423	0.9246	0.9329	0.8302	0.8302	0.0000	0.0086	-0.0121	0.0000
11	PlugInAction	0.9628	0.9643	1.0000	0.9391	0.9628	0.9643	0.9391	0.9391	0.0000	0.0000	0.0000	-0.0121
12	PlugInEvent	0.9546	0.9614	0.9160	0.7214	0.9546	0.9614	0.7214	0.7214	0.0000	0.0000	0.0000	0.0000
13	PlugInOutcome	0.9519	0.9538	0.9537	0.9470	0.9464	0.9491	0.9557	0.9557	-0.0055	-0.0048	0.0087	0.0000
14	PlugInWindow	0.9546	0.9443	0.8952	0.8693	0.9546	0.9529	0.8440	0.8440	0.0000	0.0086	-0.0253	0.0087
15	Prefs	0.8074	0.9143	0.8553	0.8870	0.8102	0.9023	0.8931	0.8931	0.0027	-0.0120	0.0061	-0.0253
16	PrintPad	0.9546	0.9576	0.9706	0.9295	0.9546	0.9576	0.9295	0.9295	0.0000	0.0000	0.0000	0.0061
17	PrintPadPreview	0.9601	0.9643	1.0000	1.0000	0.9601	0.9643	0.9285	0.9285	0.0000	0.0000	-0.0715	0.0000
18	StoppableThread	0.9601	0.9624	0.9564	0.9425	0.9601	0.9624	0.9425	0.9425	0.0000	0.0000	0.0000	-0.0715
19	TextPad	0.9274	0.8769	0.8632	1.0000	0.9192	0.8479	0.9404	0.9404	-0.0082	-0.0291	-0.0596	0.0000
20	TextTrix	0.8701	0.8498	0.7218	0.8582	0.7993	0.8250	0.8411	0.8411	-0.0709	-0.0247	-0.0171	-0.0596
21	ExtensionFileFilter	0.9573	0.9614	0.9520	0.9229	0.9573	0.9605	0.9245	0.9245	0.0000	-0.0010	0.0016	-0.0171
22	Tools	0.9628	0.9605	0.9815	0.9460	0.9628	0.9605	0.9460	0.9460	0.0000	0.0000	0.0000	0.0016
23	ExtraReturnsRe...	-	-	-		0.9328	0.9595	0.8860	0.8860	-	-	-	-
24	SongSheet	-	-	-		0.9328	0.9309	0.9376	0.9376	-	-	-	-
25	BrowseFiles	-	-	-		0.9410	0.9453	0.9420	0.9420	-	-	-	-
26	BrowseFilesFromTextPad	-	-	-		0.9628	0.9605	0.8970	0.8970	-	-	-	-
27	BrowseSingleFile	-	-	-		0.9628	0.9605	0.8970	0.8970	-	-	-	-
28	DocumentSearchFilter	-	-	-		0.9601	0.9624	0.9494	0.9494	-	-	-	-
29	LineDancePanel	-	-	-		0.9464	0.9586	0.9174	0.9174	-	-	-	-
30	LineDanceTable	-	-	-		0.9628	0.9643	0.8814	0.8814	-	-	-	-
31	MotherTabbedPane	-	-	-		0.9573	0.9538	0.9581	0.9581	-	-	-	-
32	PlugInWindowTemplate	-	-	-		0.9492	0.9595	0.9161	0.9161	-	-	-	-
33	PrintPadText	-	-	-		0.9573	0.9624	0.9318	0.9318	-	-	-	-
34	CompileException	-	-	-		0.9628	0.9633	0.9621	0.9621	-	-	-	-
35	CLexer	-	-	-		0.8701	0.8704	0.9395	0.9395	-	-	-	-
36	CToken	-	-	-		0.5567	0.9462	0.9578	0.9578	-	-	-	-
37	HTMLLexer	-	-	-		0.8674	0.8704	0.9395	0.9395	-	-	-	-
38	HTMLLexer1	-	-	-		0.8156	0.8704	0.9395	0.9395	-	-	-	-
39	HTMLToken	-	-	-		0.9246	0.9491	0.9526	0.9526	-	-	-	-
40	HTMLToken1	-	-	-		0.9083	0.9453	0.9596	0.9596	-	-	-	-
41	JavaLexer	-	-	-		0.8756	0.8704	0.9395	0.9395	-	-	-	-
42	JavaScriptLexer	-	-	-		0.8756	0.8704	0.9395	0.9395	-	-	-	-
43	JavaScriptToken	-	-	-		0.5839	0.9472	0.9561	0.9561	-	-	-	-
44	JavaToken	-	-	-		0.5975	0.9472	0.9561	0.9561	-	-	-	-
45	LatexLexer	-	-	-		0.8756	0.8704	0.9395	0.9395	-	-	-	-
46	LatexToken	-	-	-		0.9301	0.9510	0.9491	0.9491	-	-	-	-
47	PlainLexer	-	-	-		0.8756	0.8704	0.9395	0.9395	-	-	-	-
48	PlainToken	-	-	-		0.9410	0.9510	0.9491	0.9491	-	-	-	-
49	PropertiesLexer	-	-	-		0.8510	0.8623	0.9395	0.9395	-	-	-	-
50	PropertiesToken	-	-	-		0.9246	0.9491	0.9526	0.9526	-	-	-	-
51	SQLLexer	-	-	-		0.8619	0.8704	0.9288	0.9288	-	-	-	-
52	SQLToken	-	-	-		0.9028	0.9472	0.9561	0.9561	-	-	-	-
53	Token	-	-	-		0.9573	0.9548	0.9564	0.9564	-	-	-	-
54	Colorer	-	-	-		0.9437	0.9542	0.8911	0.8911	-	-	-	-

ตารางที่ 32 เปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาสของโมเดลทั้ง 4 โมเดล (ต่อ)

55	DocPosition	-	-	-		0.9601	0.9533	0.9442	0.9442	-	-	-	-
56	DocPositionComparator	-	-	-		0.9601	0.9552	0.9301	0.9301	-	-	-	-
57	DocumentReader	-	-	-		0.9546	0.9538	0.9470	0.9470	-	-	-	-
58	HighlightedDocument	-	-	-		0.9219	0.9260	0.9352	0.9352	-	-	-	-
59	ProgrammerEditorDemo	-	-	-		0.9573	0.9470	0.9196	0.9196	-	-	-	-
60	ToHTML	-	-	-		0.9083	0.8385	0.8973	0.8973	-	-	-	-
61	TokenStyles	-	-	-		0.9601	0.9480	0.9285	0.9285	-	-	-	-
62	BufferOverflowException	-	-	-		0.9628	0.9633	0.9578	0.9578	-	-	-	-
63	CircularCharBuffer	-	-	-		0.9274	0.9150	0.9328	0.9328	-	-	-	-
64	StringHelper	-	-	-		0.9601	0.9257	0.9846	0.9846	-	-	-	-
65	Getopt	-	-	-		0.9083	0.9375	0.9255	0.9255	-	-	-	-
66	LongOpt	-	-	-		0.9410	0.9605	0.9142	0.9142	-	-	-	-
67	GetoptDemo	-	-	-		0.9628	0.9643	0.9391	0.9391	-	-	-	-

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 5
การออกแบบและพัฒนาเครื่องมือ
สำหรับการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์
ในการพัฒนาแบบอินคริเมนทัล

ในบทนี้อธิบายวิธีการพัฒนาเครื่องมือ ซึ่งมีรายละเอียดคือการอธิบายความต้องการที่เป็นหน้าที่หลัก การออกแบบการทำงาน ส่วนต่อประสานผู้ใช้งาน และการทดสอบเครื่องมือที่พัฒนา ซึ่งจะออกแบบด้วยภาษายูเอ็มแอลเพราะเป็นภาษาที่ใช้เพื่ออธิบายโมเดลต่างๆ ในการนำเสนอภาพรวมของระบบ ประกอบด้วย แผนภาพยูสเคสแสดงการเกิดปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน แผนภาพคลาสแสดงความสัมพันธ์ขององค์ประกอบในระบบ แผนภาพลำดับแสดงการเกิดปฏิสัมพันธ์ของวัตถุในระบบ และแผนภาพกิจกรรมแสดงการเกิดกิจกรรมภายในระบบ ซึ่งเครื่องมือสำหรับการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัลมีรายละเอียด ดังนี้

5.1 ความต้องการที่เป็นหน้าที่หลัก

ความต้องการที่เป็นหน้าที่หลักของเครื่องมือที่พัฒนา จะแสดงให้เห็นว่าเครื่องมือมีหน้าที่ในการทำงานอะไรบ้าง ซึ่งประกอบด้วยส่วนการทำงานหลักๆ 3 ส่วน ดังนี้

5.1.1 การนำเข้าไฟล์

เครื่องมือที่พัฒนานี้ต้องมีการนำเข้าไฟล์ซึ่งประกอบด้วยไฟล์ 2 ประเภทคือ ไฟล์แผนภาพคลาสและแผนภาพลำดับซึ่งอยู่ในรูปแบบของภาษาเอ็กซ์เอ็มแอล (xml) และไฟล์ค่าตัววัดแผนภาพซึ่งอยู่ในรูปแบบของตาราง (csv) โดยจะมีทั้งหมด 4 ไฟล์เนื่องจากแต่ละประเภทนั้นจะประกอบด้วยส่วนที่เป็นการออกแบบของอินคริเมนต์ปัจจุบัน และส่วนที่เพิ่มการออกแบบของอินคริเมนต์ถัดไปเข้าด้วยกัน

5.1.2 การเตรียมข้อมูล

เครื่องมือจะทำการอ่านไฟล์และเก็บข้อมูลของทุกๆคลาสในแต่ละอินคริเมนต์โดยข้อมูลดังกล่าวนั้นซึ่งก็คือ ค่าของตัววัดแต่ละตัวที่ได้จากโมเดลทั้งสามโมเดลที่งานวิจัยนี้ได้นำเสนอไปข้างต้น ข้อมูลจะมาจากไฟล์ 2 ประเภท คือ ไฟล์แผนภาพซึ่งอยู่ในรูปแบบของภาษาเอ็กซ์เอ็มแอล(xml) และไฟล์ค่าตัววัดแผนภาพซึ่งอยู่ในรูปแบบของตาราง (csv)

เมื่อทำการเก็บค่าต่างๆที่ระบุไว้ตามโมเดลทั้งสามแล้วก็จะทำการคำนวณค่าตามข้อกำหนดในโมเดลนั้นๆ ซึ่งโมเดลในการประมาณค่าจะมีทั้งหมด 4 โมเดลดังที่งานวิจัยได้นำเสนอไว้ คือโมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) โมเดลที่ 2 สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โมเดลที่ 3 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) และโมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) เพื่อให้ได้ค่าประมาณความเสถียรเชิง

ตรรกะของคลาสในแต่ละคลาส จากนั้นเครื่องมือจะค้นหาคลาสที่มีการเปลี่ยนแปลงของค่าประมาณที่ได้จากอินคริเมนต์ปัจจุบันกับค่าประมาณที่ได้จากการเพิ่มส่วนของอินคริเมนต์ถัดไป แล้วจึงเก็บข้อมูลไว้เพื่อการแสดงผลในขั้นตอนถัดไป

5.1.3 การแสดงผล

การแสดงผลของเครื่องมือนี้จะแบ่งออกเป็น 2 รูปแบบ คือ แสดงผลในรูปแบบของตารางซึ่งจะประกอบไปด้วยคอลัมน์ต่างๆที่จะนำไปใช้ในการเปรียบเทียบค่าประมาณ และแสดงผลในรูปแบบของกราฟเส้นซึ่งจะทำให้เห็นถึงความแตกต่างของค่าประมาณความเสถียรของอินคริเมนต์ปัจจุบันและเมื่อมีการเพิ่มส่วนของอินคริเมนต์ถัดไป ผู้ใช้สามารถบันทึกไฟล์เพื่อการนำข้อมูลไปใช้งานในภายหลังได้ซึ่งเครื่องมือจะบันทึกตาราง (csv) และบันทึกกราฟเส้นเป็นไฟล์รูปภาพ (png)

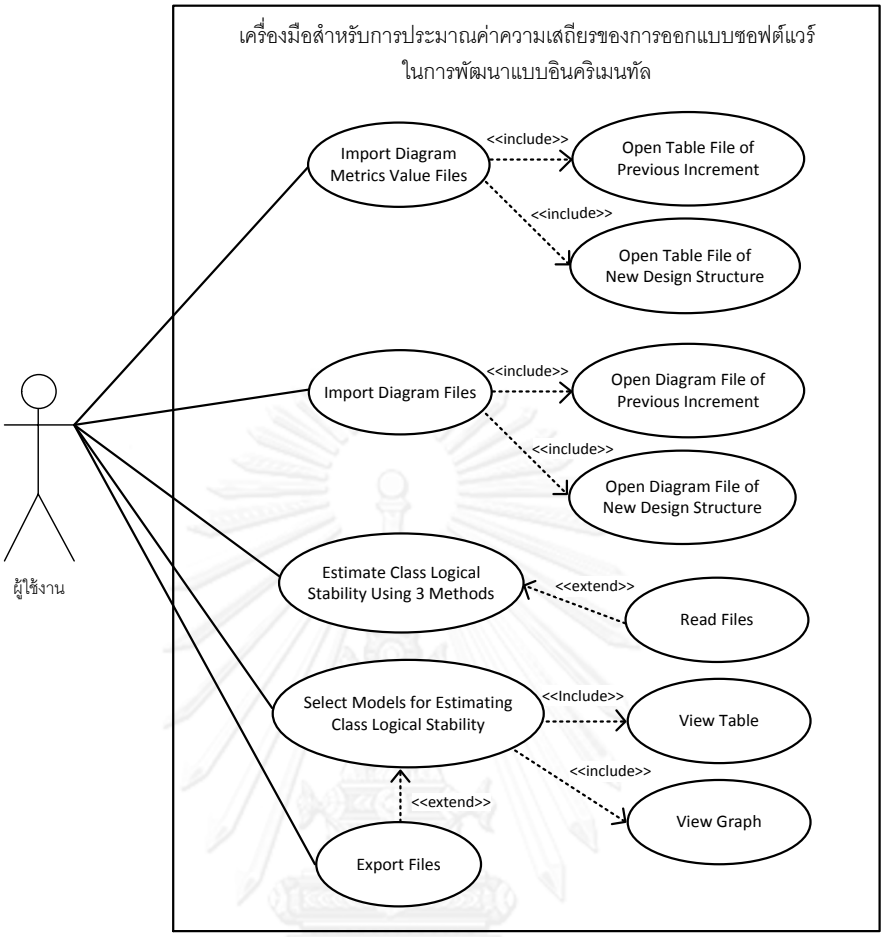
5.2 การออกแบบเครื่องมือ

เครื่องมือนี้พัฒนาขึ้นเพื่อช่วยอำนวยความสะดวกในการประมาณค่าความเสถียรเชิงตรรกะของคลาส จากขั้นตอนการออกแบบในการพัฒนาแบบอินคริเมนทัล ซึ่งสามารถอธิบายรายละเอียดต่างๆ โดยแผนภาพการออกแบบดังต่อไปนี้

5.2.1 แผนภาพยูสเคส

จากการวิเคราะห์การเกิดปฏิสัมพันธ์ระหว่างระบบงานกับผู้ใช้งาน สามารถเขียนแผนภาพได้ดังรูปที่ 18 โดยจะอธิบายเป็นลำดับเหตุการณ์ที่เกิดขึ้น เพื่อแสดงให้เห็นว่าผู้ใช้งานจะสามารถใช้งานระบบทำอะไรบ้าง ซึ่งมีรายละเอียดดังนี้

1) ผู้ใช้งานสามารถกำหนดชุดข้อมูลค่าตัววัดนำเข้าที่ต้องการนำมาประมาณค่าความเสถียรเชิงตรรกะของคลาสได้
2) ผู้ใช้งานสามารถกำหนดชุดข้อมูลแผนภาพนำเข้าที่ต้องการนำมาประมาณค่าความเสถียรเชิงตรรกะของคลาสได้
3) ผู้ใช้งานสามารถประมาณค่าความเสถียรเชิงตรรกะของคลาสที่ต้องการ ทั้งในอินคริเมนต์ก่อนหน้าและในโครงสร้างการออกแบบใหม่ที่ได้มีการเพิ่มส่วนการออกแบบของอินคริเมนต์ปัจจุบัน
4) ผู้ใช้งานสามารถเลือกการแสดงผลค่าประมาณความเสถียรได้ มีให้เลือกทั้งหมด 4 โมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสที่ได้นำเสนอไว้ในงานวิจับข้างต้น และการแสดงผลจะประกอบด้วย ตาราง และกราฟ แสดงการเปรียบเทียบค่าประมาณความเสถียรของคลาสในแต่ละอินคริเมนต์
5) ผู้ใช้งานสามารถบันทึกไฟล์ค่าประมาณความเสถียรเชิงตรรกะของคลาสเก็บไว้ได้



รูปที่ 18 แผนภาพยูสเคสของเครื่องมือ

สามารถอธิบายความสัมพันธ์ของยูสเคสที่เกิดขึ้นภายในระบบงานได้ดังต่อไปนี้

1) ยูสเคส Import Diagram Metrics Value Files

จากตารางที่ 33 เป็นยูสเคสที่แสดงลำดับเหตุการณ์เมื่อผู้ใช้นำเข้าไฟล์ชุดข้อมูลค่าตัววัดเข้าสู่ระบบ เพื่อนำไปประมาณค่าความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์

ตารางที่ 33 ขั้นตอนการทำงานของยูสเคส Import Diagram Metrics Value Files

Use case name	Import Diagram Metrics Value Files
Entry condition	1. ระบุชื่อแฟ้มงานที่จะบันทึกเมื่อเสร็จสิ้นการประมาณค่าความเสถียร และเลือกแหล่งที่เก็บ 2. มีไฟล์ชุดข้อมูลค่าตัววัดแผนภาพเพื่อใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาสของทั้งอินคริเมนต์ก่อนหน้า (Previous increment) และเมื่อมีการเพิ่มอินคริเมนต์ปัจจุบัน (Current increment) จนเกิดเป็นโครงสร้างการออกแบบใหม่ (New design structure) 3. ผู้ใช้งานเลือกนำเข้าไฟล์ค่าตัววัดแผนภาพ
Flow of events	4. ผู้ใช้งานเลือกนำเข้าไฟล์ข้อมูลค่าตัววัดของอินคริเมนต์ก่อนหน้าจากแหล่งที่เก็บ 5. ผู้ใช้งานเลือกนำเข้าไฟล์ข้อมูลค่าตัววัดของโครงสร้างการออกแบบใหม่จากแหล่งที่เก็บ
Exit condition	6. ผู้ใช้งานเลือกทำรายการอื่น

2) ยูสเคส Import Diagram Files

จากตารางที่ 34 เป็นยูสเคสที่แสดงลำดับเหตุการณ์เมื่อผู้ใช้นำเข้าไฟล์ชุดแผนภาพ ซึ่งจะประกอบด้วยแผนภาพคลาสและแผนภาพลำดับเข้าสู่ระบบ เพื่อนำไปประมาณค่าความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์

ตารางที่ 34 ขั้นตอนการทำงานของยูสเคส Import Diagram Files

Use case name	Import Diagram Files
Entry condition	1. มีไฟล์ชุดแผนภาพเพื่อใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาสของทั้งอินคริเมนต์ก่อนหน้า (Previous increment) และเมื่อมีการเพิ่มอินคริเมนต์ปัจจุบัน (Current increment) จนเกิดเป็นโครงสร้างการออกแบบใหม่ (New design structure) 2. ผู้ใช้งานเลือกนำเข้าไฟล์แผนภาพ
Flow of events	3. ผู้ใช้งานเลือกนำเข้าไฟล์แผนภาพของอินคริเมนต์ก่อนหน้าจากแหล่งที่เก็บ 4. ผู้ใช้งานเลือกนำเข้าไฟล์แผนภาพของโครงสร้างการออกแบบใหม่จากแหล่งที่เก็บ
Exit condition	5. ผู้ใช้งานเลือกทำรายการขั้นตอนถัดไป

ตารางที่ 35 ขั้นตอนการทำงานของยูสเคส Estimate Class Logical Stability Using 3 Methods

Use case name	Estimate Class Logical Stability Using 3 Methods
Entry condition	1. มีชุดข้อมูลค่าตัววัดแผนภาพและชุดแผนภาพที่มีการนำเข้าอย่างความถูกต้อง และระบบสามารถเปิดเพื่ออ่านไฟล์ได้ 2. ผู้ใช้งานเลือกเมนูในขั้นตอนต่อไป (Compare files)
Flow of events	3. ระบบนำไฟล์มาตรวจความสอดคล้องของการออกแบบในอินคริเมนต์ก่อนหน้า กับโครงสร้างการออกแบบใหม่ที่มีการเพิ่มส่วนของอินคริเมนต์ปัจจุบัน 4. ระบบอ่านค่าตัววัดตามที่กำหนดไว้ในแต่ละโมเดล และเก็บข้อมูลระบุตามหมายเลขคลาส ชื่อคลาส ลำดับของอินคริเมนต์ 5. ระบบหาจำนวนของข้อกำหนดจากโมเดล โดยทำการวัดจากแผนภาพคลาสและแผนภาพลำดับ และเก็บข้อมูลระบุตามหมายเลขคลาส ชื่อคลาส ลำดับของอินคริเมนต์ 6. ระบบทำการประมาณค่าความเสถียรเชิงตรรกะของคลาส โดยโมเดลการประมาณค่าความเสถียรทั้ง 4 โมเดลที่งานวิจัยได้นำเสนอไว้ข้างต้น 7. ระบบค้นหาคลาสที่มีการเปลี่ยนแปลงของค่าประมาณความเสถียร และคำนวณค่าที่เพิ่มขึ้นหรือลดลง แล้วเก็บค่าไว้ 8. ระบบแสดงจำนวนคลาสของอินคริเมนต์ก่อนหน้า จำนวนคลาสของโครงสร้างการออกแบบใหม่ และจำนวนคลาสของอินคริเมนต์ปัจจุบัน
Exit condition	9. ผู้ใช้งานเลือกทำรายการขั้นตอนถัดไป

3) ยูสเคส Estimate Class Logical Stability Using 3 Methods

จากตารางที่ 35 เป็นยูสเคสที่แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้นำเข้าไฟล์ทั้งหมดเสร็จเรียบร้อยแล้ว ระบบจะทำการตรวจสอบไฟล์ อ่านไฟล์และเก็บค่าของตัววัดต่างๆ แล้วนำไปประมาณค่าความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์ด้วยโมเดลทั้ง 3 โมเดลที่ได้นำเสนอไว้

4) ยูสเคส Select Models for Estimating Class Logical Stability

จากตารางที่ 36 เป็นยูสเคสที่แสดงลำดับเหตุการณ์การเลือกโมเดลที่ใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาสในแต่อินคริเมนต์ เพื่อดูค่าประมาณที่ได้จากโมเดลที่เลือก

ตารางที่ 36 ขั้นตอนการทำงานของยูสเคส Select Models for Estimating Class Logical Stability

Use case name	Select Models for Estimating Class Logical Stability
Entry condition	1. ชุดข้อมูลที่ผู้ใช้นำเข้าสู่ระบบมีความถูกต้องสอดคล้องกันเป็นซอฟต์แวร์แบบอินคริเมนทัล หรือตรวจสอบแล้วว่าเป็นการออกแบบของโปรแกรมเดียวกัน 2. ระบบสามารถอ่านไฟล์และเก็บค่าข้อมูลที่ใช้ในการประมาณค่า 3. ผู้ใช้งานเลือกเมนูในขั้นตอนต่อไป (Result)
Flow of events	4. ผู้ใช้งานเลือกโมเดลที่ใช้ในการประมาณค่าความเสถียรเชิงตรรกะของคลาส 5. ระบบแสดงค่าประมาณในรูปแบบของตารางและกราฟแสดงการเปรียบเทียบค่าประมาณของอนคริเมนต์ก่อนหน้าและเมื่อมีการเพิ่มส่วนการออกแบบของอินคริเมนต์ปัจจุบัน
Exit condition	6. ผู้ใช้งานเลือกใช้งานเมนูอื่น

5) ยูสเคส Export Files

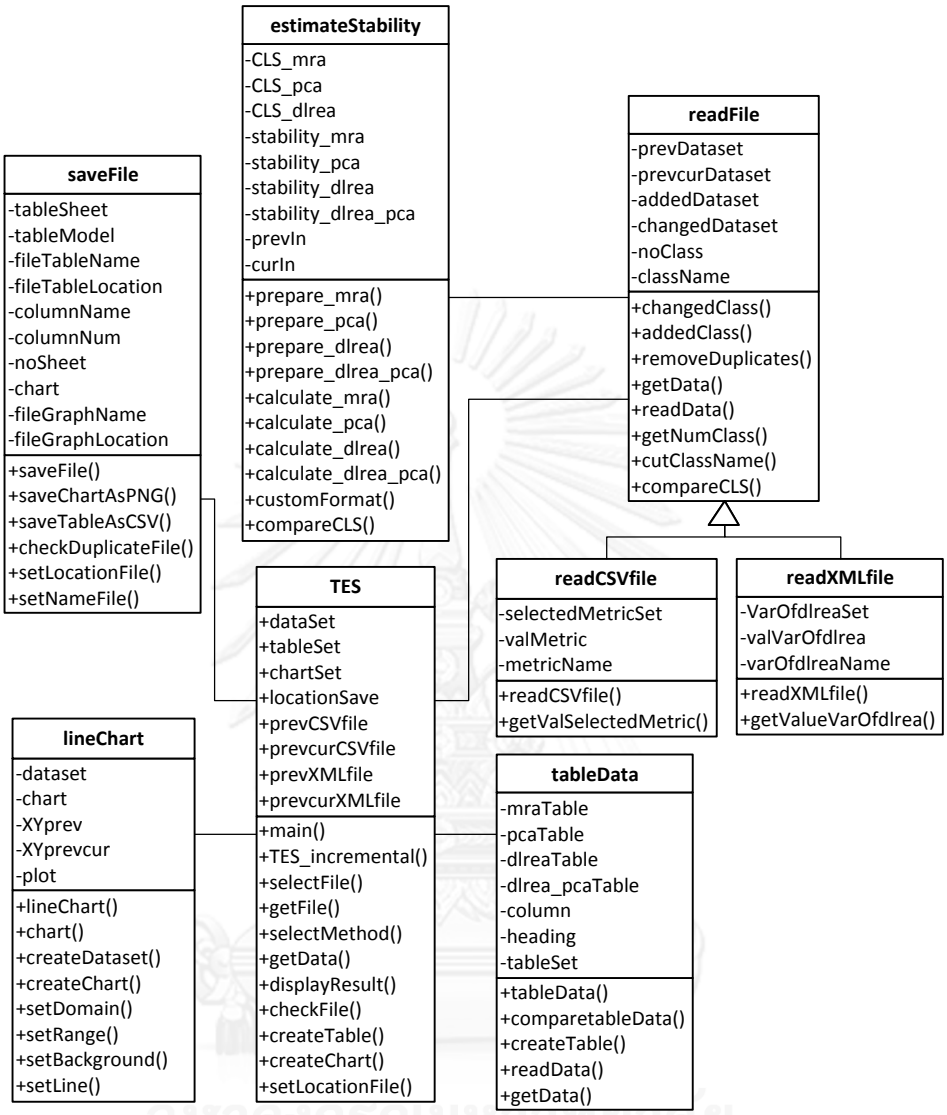
จากตารางที่ 37 เป็นยูสเคสที่แสดงลำดับเหตุการณ์ที่เกิดขึ้นเมื่อผู้ใช่เลือกเมนูคำสั่ง Export files จากระบบ เพื่อการบันทึกไฟล์ของการเปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาสทั้งในรูปการแสดงแบบตารางและกราฟเส้น

ตารางที่ 37 ขั้นตอนการทำงานของยูสเคส Export Files

Use case name	Export Files
Entry condition	1. มีชุดข้อมูลของค่าประมาณความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์ 2. ผู้ใช้เลือกเมนู Export files
Flow of events	3. ผู้ใช้งานเลือกที่เก็บของไฟล์ตารางและกราฟเปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาส 4. ผู้ใช้งานตั้งชื่อแฟ้มงาน หรือใช้ชื่อแฟ้มงานเดิมที่ปรากฏ
Exit condition	5. ผู้ใช้งานเลือกบันทึก หรือยกเลิกการบันทึก

5.2.2 แผนภาพคลาส

แผนภาพคลาสแสดงความสัมพันธ์ขององค์ประกอบของระบบงานภายในเครื่องมือดังรูปที่ 19 ซึ่งสามารถอธิบายได้ดังนี้



รูปที่ 19 แผนภาพคลาสแสดงองค์ประกอบของเครื่องมือ

1) แผนภาพคลาส TES

คลาสแสดงหน้าจอการติดต่อของเครื่องมือ TES เป็นคลาสหลักของระบบ ซึ่งคำว่า TES ในที่นี้เป็นคำย่อมาจาก Tool for Estimating Stability โครงสร้างของคลาสแบ่งตามการทำงานของหน้าจอ ซึ่งแบ่งเป็นขั้นตอน 3 ขั้นตอนตามลำดับ เริ่มจากการนำเข้าไฟล์ เตรียมข้อมูล และการแสดงผล โดยมีการเรียกใช้งานไปยังคลาสอื่นๆ คือ เรียกใช้งานคลาส readFile เพื่อเปิดไฟล์เพื่ออ่านและเก็บค่าชุดข้อมูล เรียกใช้งานคลาส tableData เพื่อนำผลการประมาณค่าที่ได้มาแสดงผลในรูปแบบของตาราง เรียกใช้งานคลาส lineChart เพื่อนำผลการประมาณค่าที่ได้มาแสดงผลในรูปแบบของกราฟเส้น และเรียกใช้งานคลาส saveFile เพื่อการบันทึกผลค่าประมาณความเสถียรเชิงตรรกะที่คำนวณได้จากระบบ

2) แผนภาพคลาส readFile

คลาสอ่านไฟล์ readFile เป็นคลาสที่ถูกเรียกใช้งานจากคลาส TES โดยการอ่านไฟล์แบ่งออกเป็นลักษณะเฉพาะ 2 ชนิด คือ ไฟล์แผนภาพ (xml) และไฟล์ค่าตัววัด (csv) โดยจะมีการเรียกใช้งานจากคลาส estimateStability เมื่อสามารถอ่านไฟล์และเก็บค่าตัววัดจากไฟล์ทั้ง 2 ชนิดแล้ว เพื่อประมาณค่าความเสถียรเชิงตรรกะของคลาส

3) แผนภาพคลาส readXMLfile

คลาสอ่านไฟล์แผนภาพ readXMLfile เป็นคลาสที่สามารถสืบทอดแอททริบิวต์และเมธอดจากคลาส readFile ซึ่งภายในไฟล์แผนภาพนั้นจะประกอบด้วยแผนภาพคลาส และแผนภาพลำดับ เป็นคลาสที่จะต้องอ่านไฟล์และทำการเก็บค่าตามตัววัดและข้อจำกัดของโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ

4) แผนภาพคลาส readCVSfile

คลาสอ่านไฟล์ค่าตัววัด readCVSfile เป็นคลาสที่สามารถสืบทอดแอททริบิวต์และเมธอดจากคลาส readFileภายในไฟล์ค่าตัววัดนั้นจะประกอบด้วยตัววัดต่างๆและค่าของตัววัดนั้นๆ ซึ่งจะระบุไว้ตามคลาส เป็นคลาสที่จะต้องอ่านไฟล์และทำการเก็บค่าตัววัดตามตัวแปรของโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาส

5) แผนภาพคลาส estimateStability

คลาสประมาณค่าความเสถียรเชิงตรรกะของคลาส estimateStability เป็นคลาสที่คำนวณค่าประมาณความเสถียรเชิงตรรกะของคลาสนั้นๆในแต่ละอินคริเมนต์ โดยใช้โมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสทั้ง 3 โมเดล ซึ่งจะได้ค่าที่แตกต่างกันไปในแต่ละโมเดล พร้อมทั้งระบุคลาสที่มีการเปลี่ยนแปลงของค่าความเสถียรและคำนวณค่าที่เปลี่ยนแปลงไป โดยจะถูกเรียกใช้งานโดยคลาส readFile

6) แผนภาพคลาส lineChart

คลาสสร้างกราฟเส้นเปรียบเทียบค่าความเสถียรเชิงตรรกะของคลาส lineChart เป็นคลาสที่นำชุดค่าประมาณความเสถียรเชิงตรรกะของคลาสในแต่ะอินคริเมต์ที่ได้มาสร้างเป็นกราฟเส้น เพื่อนำไปแสดงผลซึ่งจะถูกเรียกใช้งานโดยคลาส TES

7) แผนภาพคลาส tableData

คลาสสร้างตารางเปรียบเทียบค่าความเสถียรเชิงตรรกะของคลาส tableData เป็นคลาสที่นำชุดค่าประมาณความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์ที่ได้มาสร้างเป็นตาราง เพื่อนำไปแสดงผลซึ่งจะถูกเรียกใช้งานโดยคลาส TES

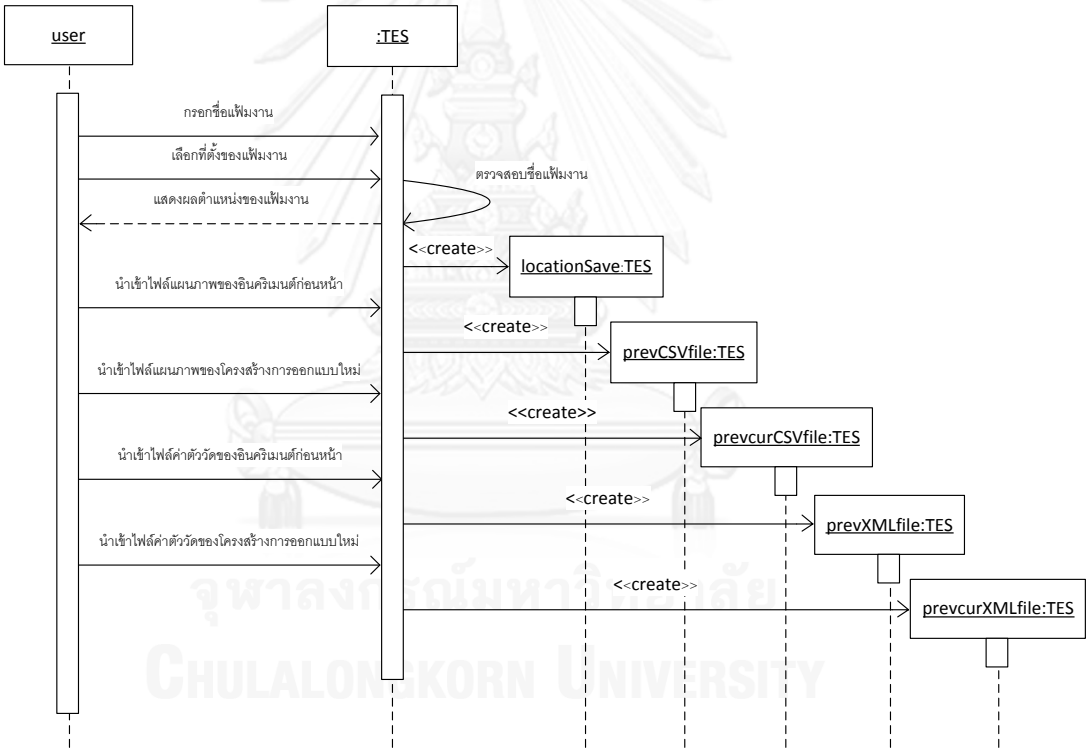
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

8) แผนภาพคลาส saveFile

คลาสบันทึกไฟล์ saveFile เป็นคลาสที่จะบันทึกค่าประมาณที่แสดงผลทางหน้าจอ ประกอบด้วย ไฟล์ตารางเปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาส และไฟล์กราฟเส้นแสดงการเปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาส ซึ่งจะถูกเรียกใช้งานโดยคลาส TES เมื่อมีคำสั่งบันทึกไฟล์

5.2.3 แผนภาพลำดับ

แผนภาพลำดับแสดงการเกิดปฏิสัมพันธ์ภายในระบบงานของเครื่องมือ ซึ่งสามารถอธิบายได้ดังนี้



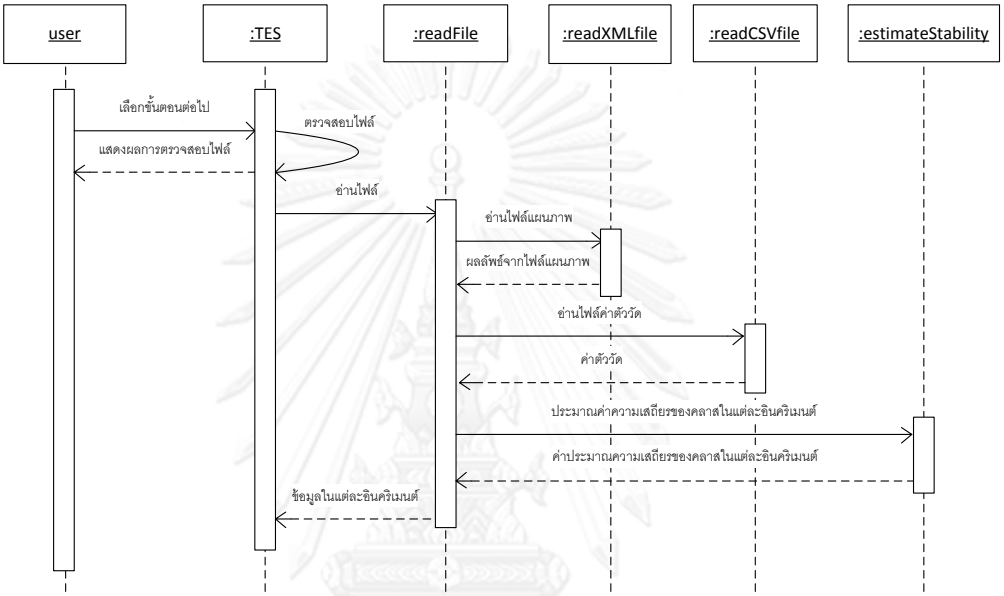
รูปที่ 20 แผนภาพลำดับแสดงการนำเข้าไฟล์สู่ระบบงาน

1) แผนภาพลำดับแสดงการนำเข้าไฟล์สู่ระบบงาน

จากรูปที่ 20 เป็นแผนภาพลำดับที่แสดงการนำเข้าไฟล์สู่ระบบงาน โดยผู้ใช้ทำการกรอกชื่อแฟ้มงานและเลือกที่อยู่แฟ้มงาน จากนั้นระบบจะทำการตรวจสอบชื่อแฟ้มงานแล้วแสดงผลตำแหน่งของแฟ้มงานและสร้างวัตถุ locationSave ซึ่งระบุที่ตั้งของแฟ้มงานเมื่อมีการบันทึกไฟล์ เมื่อผู้ใช้เลือกไฟล์จากที่เก็บทั้งไฟล์แผนภาพและไฟล์ค่าตัววัด ระบบก็จะสร้างวัตถุซึ่งมีทั้งหมด 4 วัตถุ ได้แก่

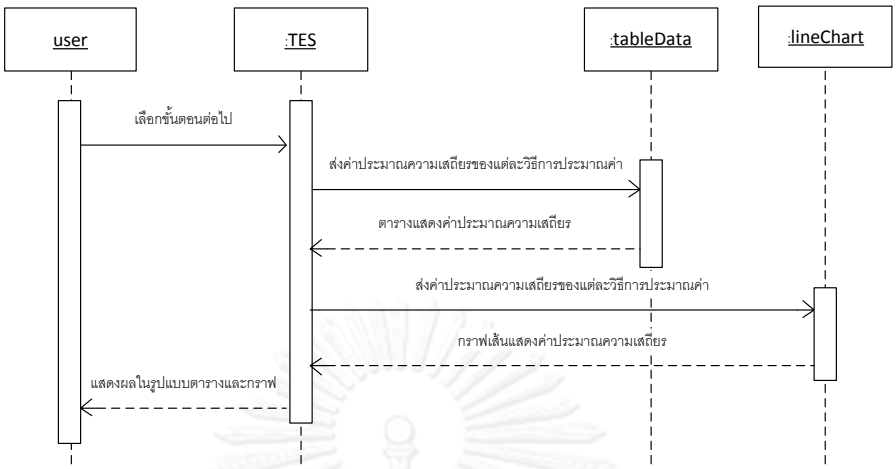
prevCSVfile prevcurCSVfile prevXMLfile และ prevcurXMLfile เนื่องจากประกอบด้วยไฟล์ 2 ประเภท และในแต่ละประเภทประกอบด้วย ไฟล์ของอินคริเมนต์ก่อนหน้า (Previous increment) และไฟล์ของโครงสร้างการออกแบบใหม่ (New design structure) เมื่อมีการเพิ่มส่วนของอินคริเมนต์ปัจจุบัน (Current increment)

2) แผนภาพลำดับแสดงการเตรียมข้อมูล



รูปที่ 21 แผนภาพลำดับแสดงการเตรียมข้อมูล

จากรูปที่ 21 แผนภาพลำดับแสดงลำดับขั้นตอนในการเตรียมข้อมูล หรือการคำนวณค่าประมาณความเสถียรเชิงตรรกะของคลาส โดยผู้ใช้จะทำการเลือกขั้นตอนถัดไปเมื่อทำการนำเข้าไฟล์จนครบถ้วนเรียบร้อยแล้ว ระบบจะทำการตรวจสอบความถูกต้องครบถ้วนของไฟล์แล้วแสดงผลการตรวจสอบ จากนั้นในกรณีที่ไฟล์ถูกต้องครบถ้วน ระบบจะทำการอ่านไฟล์เพื่อเก็บค่าจากไฟล์แผนภาพและไฟล์ค่าตัววัด แล้วทำการคำนวณค่าประมาณความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์ แล้วทำการรวบรวมข้อมูล เมื่อเสร็จสิ้นการคำนวณค่าจากทั้ง 3 โมเดลแล้ว ระบบจะแสดงผลข้อมูลในแต่ละอินคริเมนต์ ประกอบด้วย จำนวนคลาสของอินคริเมนต์ก่อนหน้า จำนวนคลาสของอินคริเมนต์ปัจจุบัน และจำนวนคลาสของโครงสร้างการออกแบบใหม่ ส่วนค่าประมาณความเสถียรนั้นจะทำการแสดงผลในขั้นตอนถัดไป



รูปที่ 22 แผนภาพลำดับการแสดงผลค่าประมาณความเสถียรเชิงตรรกะของคลาส

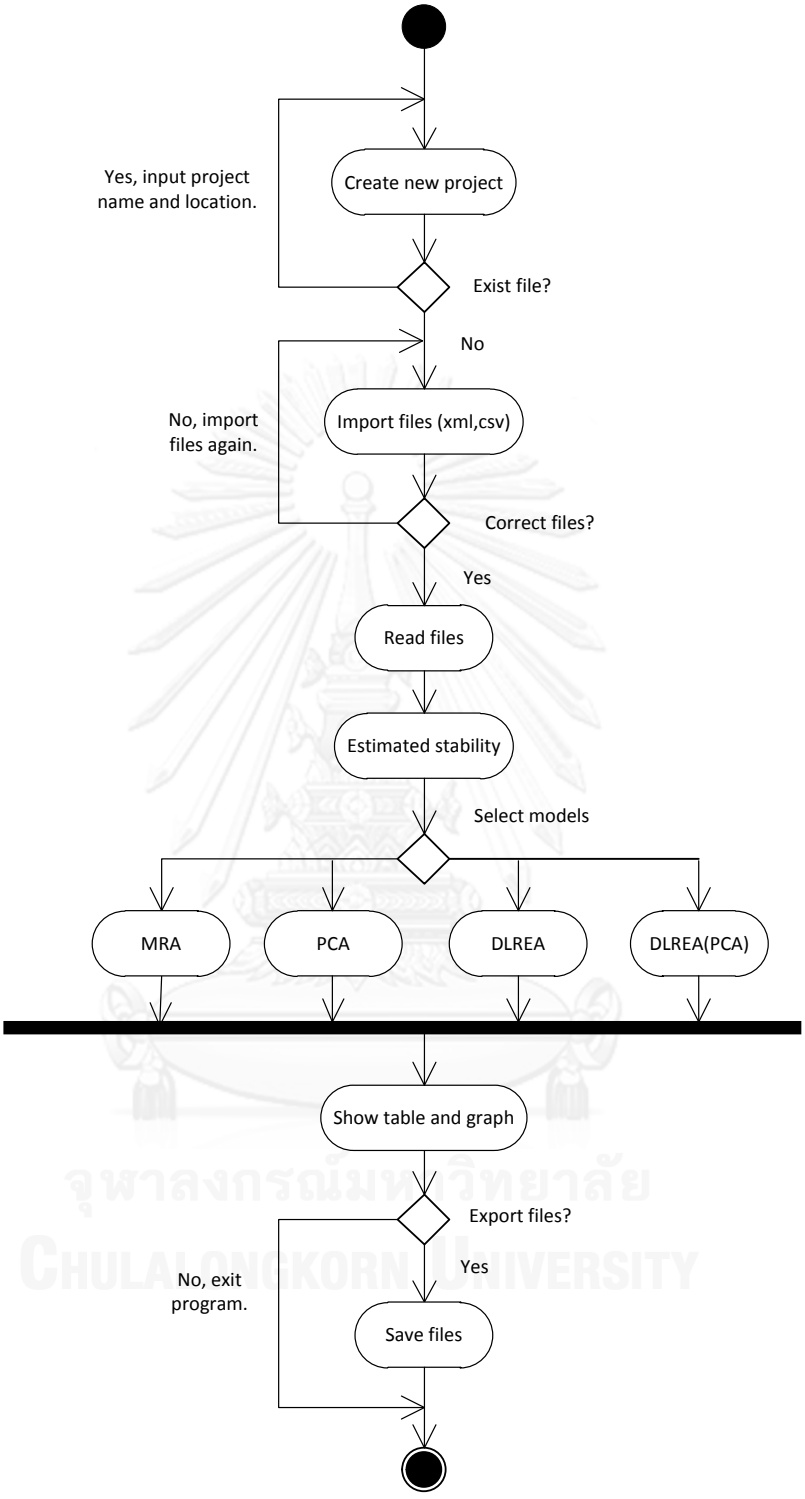
3) แผนภาพลำดับแสดงการแสดงผลค่าประมาณความเสถียรเชิงตรรกะของคลาส

จากรูปที่ 22 เป็นแผนภาพแสดงลำดับขั้นตอนการแสดงผลค่าประมาณความเสถียรเชิงตรรกะของคลาส เริ่มต้นจากการที่ผู้ใช้เลือกขั้นตอนต่อไปคือการแสดงผล ระบบจะส่งค่าประมาณความเสถียรของแต่ละโมเดลหรือวิธีการประมาณค่าซึ่งมีทั้งหมด 4 โมเดล เพื่อทำการประมวลผลแล้วแสดงผลในรูปแบบของตารางและกราฟเส้นเปรียบเทียบค่าประมาณความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์

5.2.4 แผนภาพกิจกรรม

แผนภาพกิจกรรมแสดงการเกิดกิจกรรมการทำงานต่างๆ เป็นลำดับภายในระบบงานของเครื่องมือดังรูปที่ 23 เครื่องมือนี้ใช้สำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาส จากการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล โดยใช้แผนภาพคลาสและแผนภาพลำดับ

Chulalongkorn University



รูปที่ 23 แผนภาพกิจกรรมของเครื่องมือการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล

กิจกรรมเริ่มต้นจากผู้ใช้สร้างโครงการใหม่ (Create new project) โดยกรอกชื่อโครงการและเลือกที่ตั้งแฟ้มงานของโครงการ จากนั้นระบบจะตรวจสอบไฟล์ว่าซ้ำกับที่มีอยู่ในที่ตั้งแฟ้มหรือไม่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ถ้าซ้ำก็จะต้องกลับไปกรอกชื่อโครงการและเลือกที่ตั้งแฟ้มงานใหม่อีกรอบเพื่อสร้างโครงการใหม่ ถ้าไม่ซ้ำก็จะเข้าสู่กิจกรรมถัดไปคือนำเข้าไฟล์ (Import files) สู่ระบบประกอบด้วยไฟล์ 2 ประเภท คือ ไฟล์ xml เป็นประเภทของไฟล์แผนภาพ และไฟล์ csv เป็นประเภทของไฟล์ค่าตัววัด จากนั้นระบบของเครื่องมือจะทำการตรวจสอบความถูกต้อง ครบถ้วนของไฟล์ที่นำเข้าสู่ระบบ ถ้าไม่ถูกต้องและครบถ้วนระบบจะให้ผู้ใช้กลับไปเลือกไฟล์นำเข้าสู่ระบบใหม่อีกครั้ง แต่หากถูกต้องครบถ้วนระบบก็จะทำการอ่านไฟล์ (Read files) และประมาณค่าความเสถียรเชิงตรรกะของคลาส (Estimated Stability) ซึ่งโมเดลในการประมาณค่าจะประกอบไปด้วย 4 โมเดล ได้แก่ โมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) โมเดลที่ 2 สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โมเดลที่ 3 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) และโมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) ซึ่งเมื่อผู้ใช้งานเลือกโมเดล ระบบก็จะการแสดงผลในรูปแบบของตารางและกราฟเส้น (Show table and graph) ที่แสดงค่าประมาณความเสถียรเชิงตรรกะของคลาสในแต่ละอินคริเมนต์ เมื่อระบบแสดงผลทางหน้าจอแล้ว หากผู้ใช้เลือกบันทึกไฟล์ (Export files) ระบบจะทำการบันทึกไฟล์ตารางรางและรูปภาพของกราฟเส้นไปยังที่ตั้งและชื่อโครงการที่ผู้ใช้สร้างไว้ แต่หากผู้ใช้เลือกปิดหน้าจอการทำงาน ระบบปิดและหยุดการทำงานทั้งหมด

5.2.5 เครื่องมือสนับสนุนที่ใช้

ฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ที่ใช้ในการพัฒนาเครื่องมือของงานวิจัยนี้ มีรายละเอียดดังนี้

1) ด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ส่วนบุคคล (Intel core i5 2.50 GHz)
- หน่วยความจำ (Memory) 4 กิกะไบต์
- จานบันทึกแบบแข็ง (Hard disk) ความจุ 512 กิกะไบต์

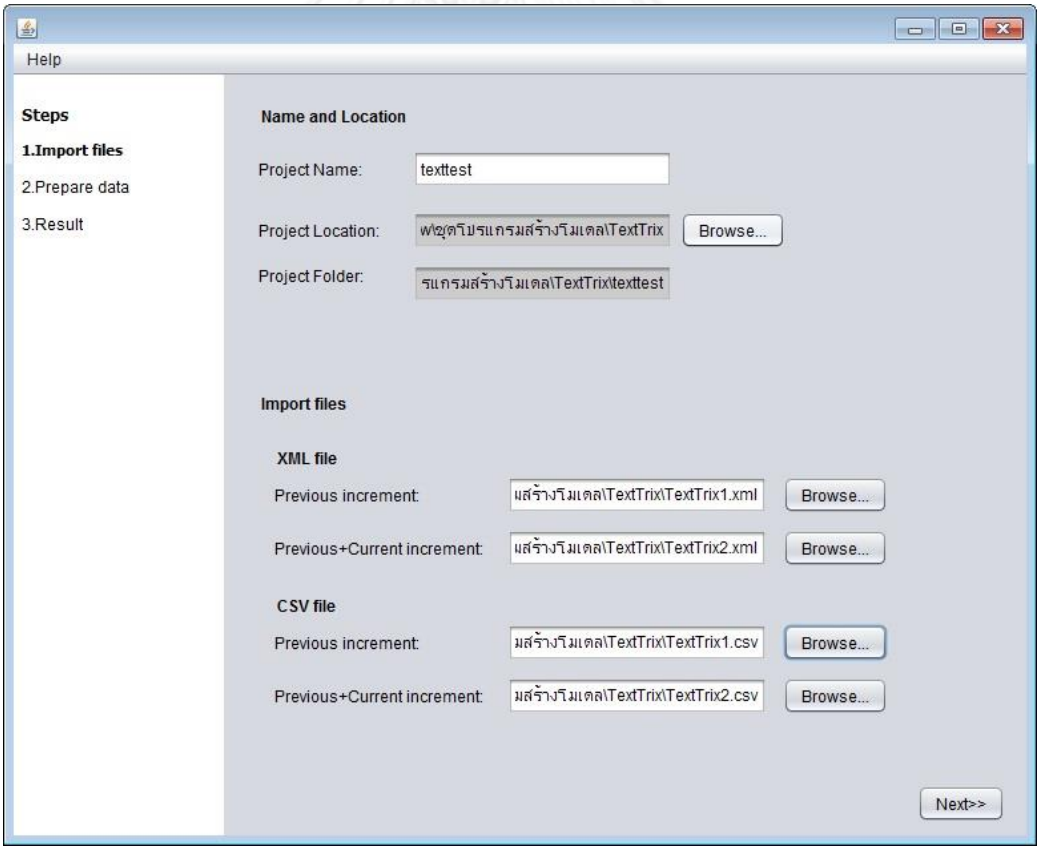
2) ด้านซอฟต์แวร์

- เครื่องมือพัฒนาโปรแกรมภาษาจาวาเน็ตบีนส์ (NetBeans)
- เครื่องมือคำนวณค่าความเสถียรเชิงตรรกะของคลาส จากซอร์ซโค้ด (ChangeImpact)
- เครื่องมือแปลงซอร์ซโค้ดเป็นแผนภาพคลาสและแผนภาพลำดับ (MagicDraw UML)
- เครื่องมือที่ใช้ในการวัดค่าตัววัดแผนภาพ (SDMetrics)
- ไมโครซอฟท์ออฟฟิส 2010 (Microsoft office)

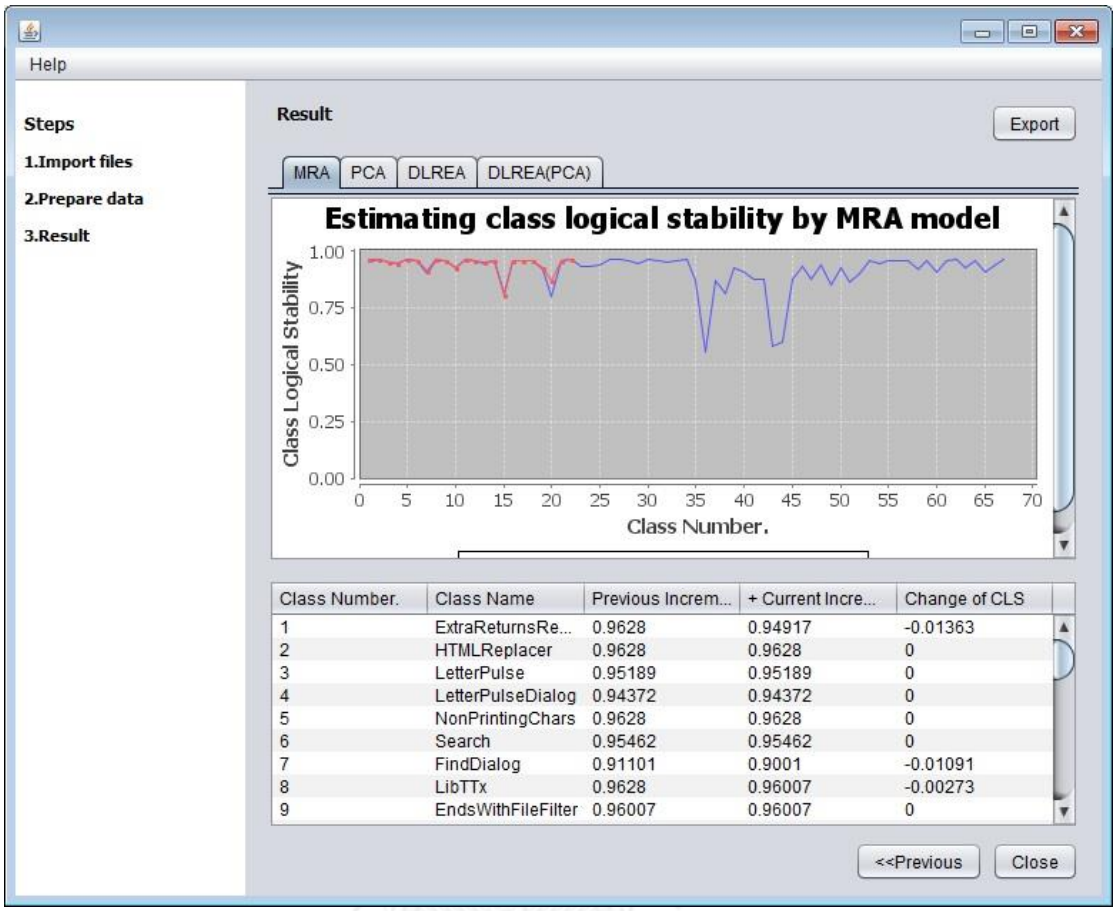
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.3 การใช้งานเครื่องมือ

เครื่องมือประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัลนี้เป็นเครื่องมือที่พัฒนาจากงานวิจัยที่ได้นำเสนอโมเดลในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัลโดยใช้แผนภาพคลาสและแผนภาพลำดับ ซึ่งงานวิจัยนี้ได้ทำการเปรียบเทียบผลการประมาณค่าความเสถียรเชิงตรรกะของคลาสในแต่ละคลาสเมื่อมีการเพิ่มการออกแบบของทุกๆ อินคริเมนต์ โดยเครื่องมือนี้จะมีการนำเข้าไฟล์ 2 ประเภท คือ xml และ csv ในแต่ละประเภทจะต้องนำเข้า 2 ไฟล์ คือ ไฟล์ของอินคริเมนต์ก่อนหน้า และไฟล์ที่มีการเพิ่มอินคริเมนต์ปัจจุบัน ซึ่งจะแบ่งการทำงานของโปรแกรมออกเป็น 3 ขั้นตอน คือ ขั้นตอนการนำเข้าไฟล์ (แสดงส่วนต่อประสานกับผู้ใช้ ดังรูปที่ 24) ขั้นตอนการเตรียมข้อมูลหรือการคำนวณค่าประมาณความเสถียรเชิงตรรกะของคลาสด้วยโมเดลที่นำเสนอในงานวิจัยนี้ทั้ง 3 โมเดล และขั้นตอนการแสดงผลซึ่งจะแสดงในรูปแบบของตารางและกราฟเส้น (แสดงส่วนต่อประสานกับผู้ใช้ ดังรูปที่ 25) เพื่อเปรียบเทียบค่าประมาณความเสถียรของคลาสในแต่ละอินตคริเมนต์ ซึ่งรายละเอียดของวิธีการใช้งานเครื่องมือและส่วนต่อประสานผู้ใช้จะนำเสนอในภาคผนวก ง



รูปที่ 24 ส่วนต่อประสานกับผู้ใช้ของขั้นตอนการนำเข้าไฟล์



รูปที่ 25 ส่วนต่อประสานกับผู้ใช้ของขั้นตอนการแสดงผล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6
สรุปผลการวิจัย

ในบทนี้ซึ่งเป็นบทสุดท้ายของวิทยานิพนธ์ฉบับนี้ จะกล่าวถึงการสรุปผลการวิจัย งานวิจัยในอนาคต และบทความวิชาการที่ได้รับการตีพิมพ์ โดยมีรายละเอียดดังนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอโมเดลในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ในการพัฒนาแบบอินคริเมนทัล โดยใช้แผนภาพคลาสและแผนภาพลำดับในการประมาณค่าความเสถียรเชิงตรรกะของคลาส เพื่อสามารถทำการประมาณค่าความเสถียรตั้งแต่ขั้นตอนของการออกแบบซอฟต์แวร์ เพราะหากต้องแก้ไขก็จะสามารถแก้ไขในขั้นตอนของการออกแบบซึ่งจะง่ายและประหยัดกว่าการแก้ไขที่ซอร์ซโค้ดหรือในขั้นตอนของการบำรุงรักษา เนื่องจากค่าประมาณความเสถียรเชิงตรรกะของคลาสนั้นบ่งบอกถึงความสามารถในการต้านทานผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้นกับซอฟต์แวร์ ซึ่งเป็นปัจจัยที่ต้องคำนึงถึงเมื่อต้องการซอฟต์แวร์ที่มีคุณภาพ

การพัฒนาซอฟต์แวร์แบบอินคริเมนทัลเป็นกระบวนการที่มีการเก็บความต้องการทั้งหมดของซอฟต์แวร์ที่จะพัฒนาก่อนที่จะนำไปวิเคราะห์ ออกแบบ และพัฒนาในส่วนของความต้องการหลักเป็นอินคริเมนต์แรก จากนั้นจึงพัฒนาตามความต้องการที่เหลือในอินคริเมนต์ถัดไป เมื่ออินคริเมนต์ปัจจุบันได้ถูกเพิ่มไปในส่วนของอินคริเมนต์ก่อนหน้า จะได้โครงสร้างการออกแบบใหม่ ซึ่งอาจจะได้รับผลกระทบจากการเปลี่ยนแปลง ด้วยเหตุนี้ความเสถียรเชิงตรรกะอาจจะมีค่าที่เปลี่ยนไปด้วย ซึ่งถ้ามีค่าลดลงจากเดิมก็อาจเป็นไปได้ว่าจะต้องมีการแก้ไขการออกแบบ ดังนั้นหากสามารถคาดการณ์หรือการประมาณค่าความเสถียรเชิงตรรกะของคลาสเมื่อมีการเพิ่มการพัฒนาในแต่ละอินคริเมนต์ในขั้นตอนของการออกแบบ ก็จะสามารถนำไปพิจารณาแก้ไขโมเดลการออกแบบก่อนจะนำไปพัฒนา

งานวิจัยนี้นำเสนอโมเดลสำหรับการประมาณค่าความเสถียรเชิงตรรกะของคลาสทั้งหมด 4 โมเดลที่สร้างด้วยวิธีแตกต่างกัน คือ โมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) โมเดลที่ 2 สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โมเดลที่ 3 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของคลาส (DLREA) และโมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของคลาสซึ่งใช้วิธีการวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด จากนั้นทำการเปรียบเทียบโมเดลทั้งหมดจากการประเมินผลโมเดล ในการประเมินผลโมเดลสำหรับประมาณค่าความเสถียรทั้ง 4 โมเดลนั้นใช้มาตรฐานทั่วไป คือ ยอมรับโมเดลเพื่อนำมาใช้ในการประมาณค่า โดยค่าประมาณที่ได้ต้องมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25% เป็นจำนวนมากกว่าหรือเท่ากับ 75 % จากค่าประมาณทั้งหมด

ขั้นตอนในการสร้างโมเดลสำหรับการประมาณค่าความเสถียรเชิงตรรกะของคลาสประกอบด้วย การเตรียมโปรแกรมโอเพ่นซอร์ซที่จะนำมาใช้ในการสร้างโมเดลและทดสอบโมเดล การวัดค่าความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ด การแปลงแผนภาพจากซอร์ซโค้ด การวัดตัววัดแผนภาพ การสร้างโมเดลและประเมินผลโมเดล แล้วจึงนำโมเดลในการประมาณค่าความเสถียรเชิง

ตรรกะของคลาสที่ได้ไปพัฒนาเครื่องมือสำหรับการประมาณค่าความเสถียรของการออกแบบในการพัฒนาแบบอินคริเมนทัล

การวัดค่าความเสถียรเชิงตรรกะของคลาสจากซอร์ซโค้ดในงานวิจัยนี้ใช้เครื่องมือที่ผู้พัฒนาได้ทำการกำหนดการเปลี่ยนแปลงที่สามารถเกิดขึ้นกับซอฟต์แวร์ซึ่งแบ่งออกเป็น 2 ประเภทคือ การเปลี่ยนแปลงในระดับของระบบซึ่งเป็นการเปลี่ยนแปลงที่เกี่ยวข้องกับความสัมพันธ์ระหว่างคลาส การเพิ่มและลบคลาส และการเปลี่ยนแปลงในระดับของคลาสซึ่งเป็นการเปลี่ยนแปลงที่เกี่ยวข้องกับองค์ประกอบภายในคลาส ประกอบด้วย แอททริบิวต์ และเมธอด

การแปลงแผนภาพจากซอร์ซโค้ด และการวัดค่าตัววัดจากแผนภาพนั้น อาศัยเครื่องมือในการแปลงจากซอร์ซโค้ดเป็นแผนภาพคลาสและแผนภาพลำดับซึ่งบันทึกเป็นไฟล์ภาษาเอ็กซ์เอ็มแอล (xml) แล้วจึงใช้เครื่องมือในการคำนวณค่าตัววัดแต่ละตัว โดยตัววัดแผนภาพที่ใช้ในงานวิจัยนี้ได้รวบรวมมาทั้งหมด 27 ตัววัดซึ่งนำมาใช้ในการสร้างโมเดลที่ 1 และโมเดลที่ 2 ส่วนโมเดลที่ 3 และโมเดลที่ 4 นั้นได้กำหนดตัววัดขึ้นมาใหม่จากข้อกำหนดของโมเดลคือวิเคราะห์องค์ประกอบของการออกแบบซึ่งงานวิจัยนี้ใช้แผนภาพคลาสและแผนภาพลำดับ

โมเดลการประมาณค่าความเสถียรที่นำเสนอในงานวิจัยนี้ประกอบด้วยตัววัดที่แตกต่างกัน คือ โมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน ตัววัดแผนภาพที่ปรากฏในโมเดลมี 6 ตัววัดจากทั้งหมด 27 ตัววัด ประกอบด้วย EC_Par, EC_Attr, MsgSent, MsgRecv, NumAttr และ Nesting โมเดลที่ 2 สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก ตัววัดแผนภาพที่สกัดได้มี 13 ตัววัดจากทั้งหมด 27 ตัววัด ได้แก่ NOC, NOP, OpsInh, AttrInh, NumDesc, NumAnc, DIT, CLD, IFImpl, EC_Par, IC_Attr, NumOps และ NumPubOps แต่ตัววัดที่ปรากฏในโมเดลประกอบด้วย EC_Par, NumOps และ NumPubOps ส่วนโมเดลที่ 3 และโมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ สร้างข้อกำหนดของตัววัดที่มาจากองค์ประกอบของแผนภาพคลาสและแผนภาพลำดับ ประกอบด้วย แอททริบิวต์ (Attribute) เมธอด (Method) ความสัมพันธ์ (Relation) และการรับส่งสาร (Message)

จากแนวคิดของการสร้างโมเดลด้วยวิธีการที่แตกต่างกันใน 2 วิธีแรกคือการวิเคราะห์ความถดถอยเชิงซ้อน และเมื่อมีการวิเคราะห์ส่วนประกอบหลักก่อนที่จะใช้การวิเคราะห์ความถดถอยเชิงซ้อนในการสร้างโมเดล ทำให้ทราบว่าการเลือกตัววัดก่อนการสร้างโมเดลก็เป็นปัจจัยสำคัญที่จะทำให้โมเดลให้ค่าประมาณมีความผิดพลาดลดลงและใกล้เคียงค่าจริงมากขึ้น แต่ตัววัดที่ใช้ในการสร้างโมเดลทั้ง 2 โมเดลดังกล่าว เป็นตัววัดที่มีผู้วิจัยในอดีตเป็นผู้นำเสนอและข้อกำหนดของตัววัดไม่มีความเชื่อมโยงกับความเสถียร ทำให้ในโมเดลที่ 3 และโมเดลที่ 4 มีแนวคิดที่จะสร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะ ซึ่งเกิดจากนิยามของความเสถียรคือค่าความต้านทานต่อผลกระทบที่เกิดจากการเปลี่ยนแปลงที่เกิดขึ้นซอฟต์แวร์ จึงนำนิยามนี้มาเป็นข้อกำหนดเพื่อการสร้างตัววัดที่มีความสอดคล้องกันสอดคล้อง ซึ่งก็คือจำนวนผลกระทบที่ได้รับเมื่อมีการเปลี่ยนแปลง

หลังจากทำการสร้างโมเดลทั้ง 4 โมเดลแล้วได้ทำการประเมินผลด้วยชุดโปรแกรมทดสอบโมเดล แล้วนำมาเปรียบเทียบผลการประเมินผลโมเดลพบว่า โมเดลที่ 1 สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน ผลการประเมินการทำนายที่ระดับ 0.25 เท่ากับ 95.94 เปอร์เซ็นต์ของ

ค่าประมาณที่ได้จะคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์ โมเดลที่ 2 สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก มีผลการประเมินการทำนายที่ระดับ 0.25 เท่ากับ 97.63 เปอร์เซ็นต์ของค่าประมาณที่ได้จะคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์ โมเดลที่ 3 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะ มีผลการประเมินการทำนายที่ระดับ 0.25 เท่ากับ 85.71 เปอร์เซ็นต์ของค่าประมาณที่ได้จะคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์ และ โมเดลที่ 4 สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะซึ่งใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด มีผลการประเมินการทำนายที่ระดับ 0.25 เท่ากับ 95.12 เปอร์เซ็นต์ของค่าประมาณที่ได้จะคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์ ซึ่งผลการประเมินโมเดลทั้ง 4 โมเดลนั้นผ่านมาตรฐานที่จะสามารถยอมรับเพื่อการการนำไปคำนวณค่าประมาณความเสถียรเชิงตรรกะของคลาส ในการพัฒนาแบบอินคริเมนทัลได้

เมื่อทำการเปรียบเทียบการประเมินโมเดลการประมาณค่าความเสถียรที่สร้างขึ้นในแต่เวอร์ชันเพิ่มเติมจาก 4 โมเดลที่กล่าวข้างต้นนั้น ผลการประเมินด้วยการทำนายที่ระดับ0.25 หรือ PRED(0.25) มีความสอดคล้องกันในทุกเวอร์ชัน โดยโมเดลที่มีความแม่นยำในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล อันดับที่ 1 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) อันดับที่ 2 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ความถดถอยเชิงซ้อน (MRA) อันดับที่ 3 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ โดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) และอันดับที่ 4 คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ (DLREA) ซึ่งได้ผลสอดคล้องกับผลการประเมินโมเดลทั้ง 4 โมเดลดังกล่าว

จากงานวิจัยนี้สามารถเลือกโมเดลที่ดีที่สุดในการนำไปประมาณค่าความเสถียรเชิงตรรกะของคลาสในขั้นตอนการออกแบบของซอฟต์แวร์ที่พัฒนาแบบอินคริเมนทัล ซึ่งก็คือ โมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลัก (PCA) โดยการคำนวณค่าจะใช้ค่าจากตัววัดแผนภาพ แต่หากต้องการใช้โมเดลที่มีข้อจำกัดของตัววัดที่สอดคล้องกับนิยามของความเสถียร โมเดลที่เหมาะสมก็คือโมเดลที่สร้างด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ โดยใช้การวิเคราะห์ส่วนประกอบหลักในการคัดเลือกตัววัด (DLREA(PCA)) เมื่อเปรียบเทียบการประเมินผลโมเดลที่สร้างด้วยวิธีที่แตกต่างกัน 4 วิธีนั้น สามารถสรุปได้ว่าโมเดลที่สร้างด้วยวิธีการวิเคราะห์ส่วนประกอบหลักจะทำให้โมเดลในการประมาณค่าความเสถียรเชิงตรรกะของคลาสให้ค่าประมาณที่ใกล้เคียงค่าจริงหรือค่าความเสถียรเชิงตรรกะของคลาสที่คำนวณจากซอร์ซโค้ดมากที่สุด ซึ่งแตกต่างจากโมเดลที่สร้างด้วยการวิเคราะห์ความถดถอยเชิงซ้อนเพียงอย่างเดียว เนื่องจากการวิเคราะห์ส่วนประกอบหลักเป็นการสกัดตัววัดก่อนจะนำมาวิเคราะห์ความถดถอยเชิงซ้อนเพื่อสร้างโมเดล แต่อย่างไรก็ตามผลการประเมินโมเดลที่ได้คือการสร้างโมเดลด้วยวิธีการวิเคราะห์ผลกระทบเชิงตรรกะของการออกแบบ ซึ่งเป็นการสร้างข้อกำหนดของตัววัดขึ้นมาใหม่ เพื่อให้สอดคล้องกับนิยามของความเสถียรเชิงตรรกะนั้นมีความแม่นยำในการทำนายค่าต่ำกว่าการสร้างโมเดลโดยใช้ตัววัดแผนภาพที่มีข้อกำหนดของตัววัดไม่สอดคล้องกับความเสถียรเชิงตรรกะ ซึ่งก็อาจเป็นไปได้ว่ามีสาเหตุมาจากความผิดพลาดในการกำหนดตัววัดที่ใช้ในการสร้างโมเดล ข้อมูลที่ใช้ในการสร้างและประเมินผลโมเดล หรืออาจมีปัจจัยอื่นนอกเหนือจากข้อกำหนดและนิยามที่ยังไม่ได้ศึกษา

6.2 ข้อจำกัด

ข้อจำกัดของงานวิจัยนี้ประกอบด้วย

1) การสร้างโมเดลการประมาณค่าความเสถียรของการออกแบบในการพัฒนาแบบอินคริเมนทัล ใช้ข้อมูลจากซอฟต์แวร์จากโอเพ่นซอร์ซ ซึ่งอาจพัฒนาด้วยกระบวนการพัฒนาซอฟต์แวร์อื่นๆ ที่ไม่ใช่การพัฒนาแบบอินคริเมนทัล โดยในงานวิจัยนี้ใช้วิธีการเลือกซอฟต์แวร์จากหมายเลขเวอร์ชันในตำแหน่ง major ที่เปลี่ยนไป ซึ่งแสดงถึงการเพิ่มของฟีเจอร์หรือฟังก์ชันต่างๆ ในเวอร์ชันถัดไปเพื่อให้มีลักษณะใกล้เคียงกับซอฟต์แวร์ที่พัฒนาแบบอินคริเมนทัล

6.3 งานวิจัยในอนาคต

สามารถแบ่งการทำงานวิจัยในอนาคตได้ดังนี้

1) ปรับปรุงโมเดลโดยออกแบบตัววัดอื่นๆ ที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาส เพื่อให้ค่าประมาณมีความใกล้เคียงกับค่าจริงหรือค่าความเสถียรเชิงตรรกะของคลาสที่คำนวณได้จากซอร์ซโค้ด
2) ทำการทดลองเพื่อหาช่วงการเปลี่ยนแปลงของค่าประมาณความเสถียรของคลาสที่ลดลงในอินคริเมนต์ถัดไป เพื่อนำไปใช้ในการตัดสินใจปรับปรุงการออกแบบคลาสนั้นๆ
3) เพิ่มตัววัดแผนภาพอื่นๆ หรือศึกษาการออกแบบโดยใช้แผนภาพอื่นๆ นอกเหนือจากแผนภาพคลาสและแผนภาพลำดับ เพื่อหาตัววัดที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาส

6.4 ผลงานตีพิมพ์จากวิทยานิพนธ์

1) หัวเรื่องงานวิจัยที่ตีพิมพ์ชื่อ “Estimating Stability of Software Design in Incremental Development” ในการประชุมวิชาการระดับนานาชาติ “The 7th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2013)” ซึ่งจัดขึ้นที่จังหวัดเชียงใหม่ ประเทศไทย ระหว่างวันที่ 18-20 ธันวาคม พ.ศ. 2556
2) หัวเรื่องงานวิจัยที่ตีพิมพ์ชื่อ “Comparison of Stability Models in Incremental Development” ในการประชุมวิชาการระดับนานาชาติ “The 6th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2014)” ซึ่งจัดขึ้นที่จังหวัดกรุงเทพมหานคร ประเทศไทย ระหว่างวันที่ 7-9 เมษายน พ.ศ. 2557

รายการอ้างอิง

[1] Yau S.S., Collofello, J.S. Some Stability Measures for Software Maintenance. IEEE Transactions on Software Engineering, 1980.

[2] Yau S.S., Collofello, J.S. Design Stability Measures for Software Maintenance. IEEE Transactions on Software Engineering, 1985.

[3] Elish M.O., Rine, D. Investigation of metrics for object-oriented design logical stability. Proceeding of the Seventh European Conference on Software Maintenance and Reengineering, 2003.

[4] Rangsiyawath S., Muenchaisri, P. Estimating Software Logical Stability from Class Diagram and Sequence Diagram. International Joint Conference on Computer Science & Software Engineering, 2007.

[5] Cheewaviriyanon C., Muenchaisri, P. Estimating Software Logical Stability Using Analogy from Class Diagram and Sequence Diagram. The 2009 Joint Conference on Computer Science and Software Engineering, Phuket, Thailand, 2009.

[6] Nimol D., Muenchaisri, P. Estimating Software Logical Stability using ANN from Class diagram and Sequence diagram. The Information Technology Journal of IT faculty of KMUTNB, 2011.

[7] Alshayeb M., Naji M., Elish M.O., Al-Ghamdi J. Towards measuring object-oriented class stability. IET Software, 2011.

[8] Yau S.S., Collofello J.S. and Macgregor T. Ripple effect analysis of software Maintenance. Computer Software and Applications Conference, 1978.

[9] David Chenho Kung, Jerry Gao, Pei Hsia, F. Wen, Yasufumi Toyoshima, and Cris Chen. Change Impact Identification in Object Oriented Software Maintenance. Proceedings of the International Conference on Software Maintenance (ICSM '94), pp.202-211, 1994.

[10] Li L. Change Impact Analysis of Object-Oriented Software. PhD thesis, George Mason University, Virginia, USA, 1998.

[11] Shaochun Xu, Zendi Cui, Yufeng Gui. Cognitive Process during Incremental Software Development. IEEE/ACIS International Conference on Computer and Information Science, 2007.

[12] Larman C. and V. R. Basili. Iterative and Incremental Development: A Brief History. IEEE Transaction on Computer. 36 (June 2003): 47-56.

[13] Abrahamsson P., et al. Effort Prediction in Iterative Software Development Processes -- Incremental Versus Global Prediction Models, Proceedings of ESEM, 2007.

[14] Pressman R. Software Engineering: A Practitioner's Approach, McGraw-Hill, 2005.

[15] Toshiya Fujii, Tadashi Dohi, and Takaji Fujiwara. Towards quantitative software Reliability assessment in incremental development processes. Proceedings of the 33rd International Conference on Software Engineering, 2011.

[16] Grady Booch, James Rumbaugh and Ivar Jacobson. The unified modeling language user guide, Addison Wesley Longman, Inc., 1999.

[17] Dongxiao N., et al. Research on Chinese cities comprehensive competitiveness based on principal component analysis and factor analysis in SPSS. IEEE 2nd International Conference on Software Engineering and Service Science, 2011.

[18] Norman E.F., Shari L.P. Software Metrics A Rigorous and Practical Approach, Second Edition ed, PWS Publishing Company, 1997.

[19] Cont, S. Dunsnore H. and Shen V. Software Engineering Metrics and Models, California:Benjamin/Cummings, 1986.

[20] CodePlex Project Hosting for Open Source Software. Available from : http://www.codeplex.com [2013, May 5]

[21] SourceForge Open Source Applications and Software Directory. Available from : http://sourceforge.net [2013, May 5]

[22] MagicDraw Software Package. Available from : http://www.nomagic.com/products/magicdraw.html [2013, April 24]

[23] SDMetrics Software Package. Available from : http://www.sdmetrics.com [2013, April 24]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

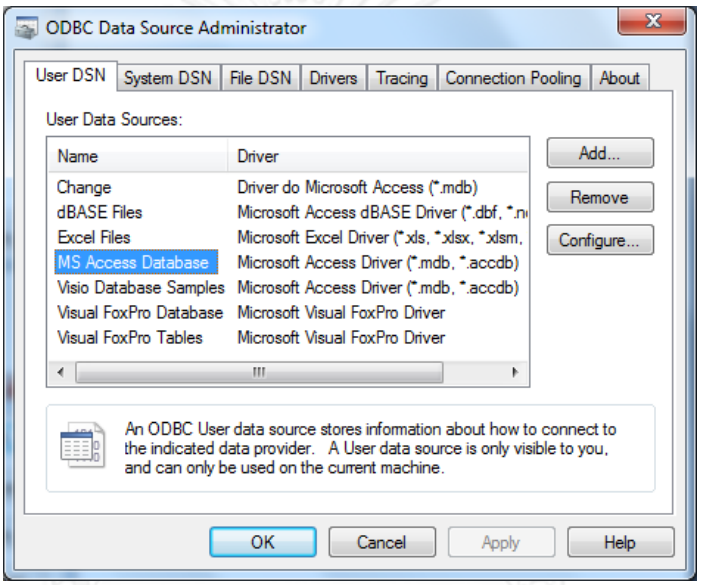
ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

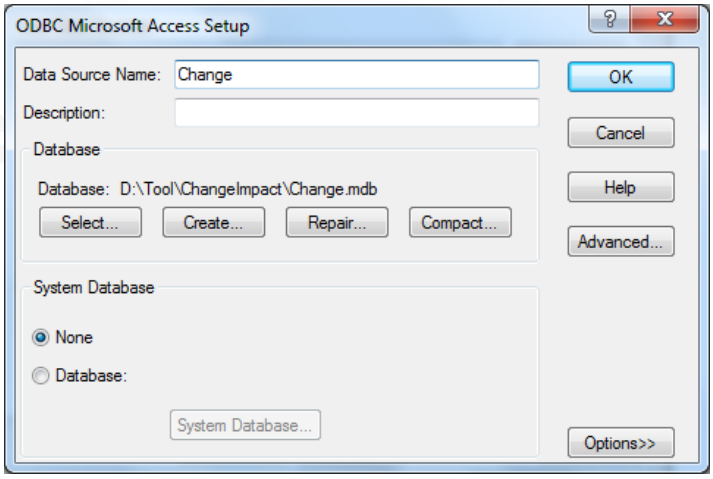
ภาคผนวก ก

การคำนวณความเสถียรเชิงตรรกะของคลาสโดยโปรแกรม

ในภาคผนวก ก จะนำเสนอการคำนวณความเสถียรเชิงตรรกะของคลาสจะใช้โปรแกรม ChangeImpact ซึ่งพัฒนาโดยนายศุภวัฒน์ รังสิยวัฒน์ ซึ่งคำนวณค่าโดยการจำลองการเปลี่ยนแปลงตามที่ได้กำหนดไว้ไปยังซอร์ซโค้ดภาษาจาวา จากนั้นจะได้ค่าความเสถียรเชิงตรรกะของคลาสทั้งหมดในโปรแกรม ซึ่งต้องใช้โอดีบีซีในการกำหนดที่จัดเก็บข้อมูล เลือก MS Access Database แล้วเลือก Add... แสดงได้ดังรูปที่ 26 ทำการกำหนด Data Source Name ชื่อ Change และไฟล์ Database ชื่อ Change.mdb แสดงได้ดังรูปที่ 27

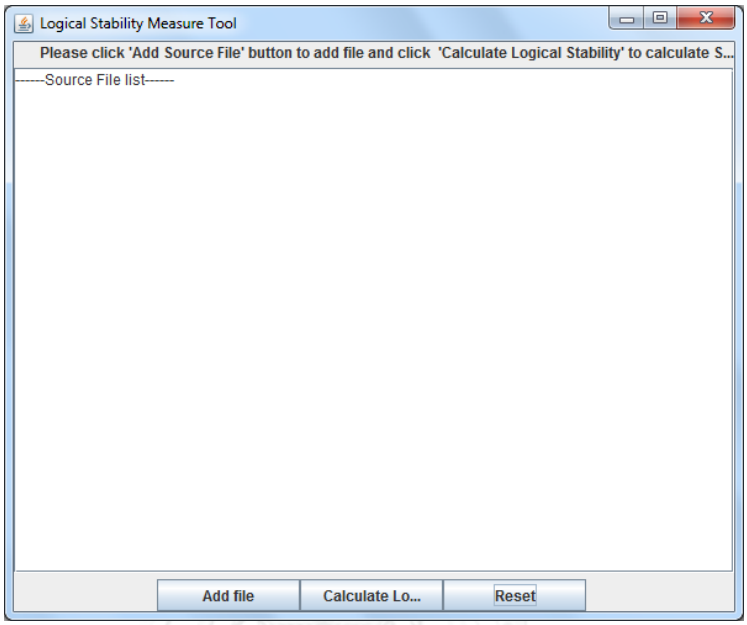


รูปที่ 26 การตั้งค่าโอดีบีซีในหน้าต่าง ODBC Data Source Administrator

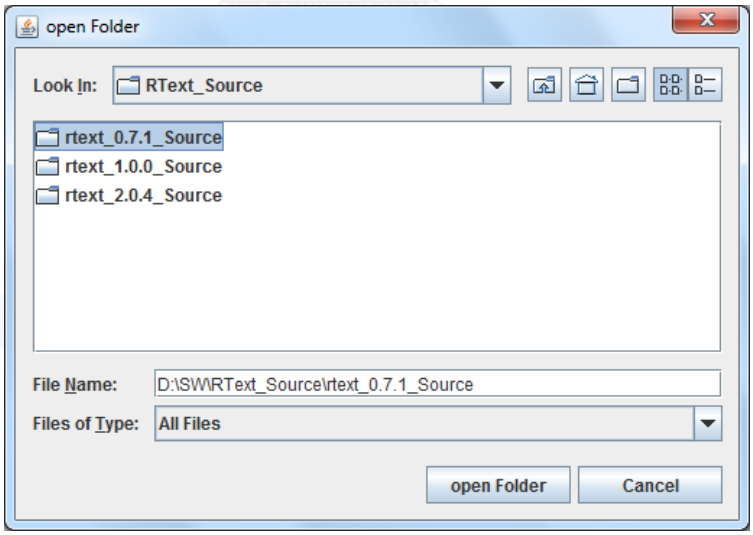


รูปที่ 27 การตั้งค่าโอดีบีซีในหน้าต่าง ODBC Microsoft Access Setup

เมื่อเปิดโปรแกรมจะแสดงส่วนต่อประสานของโปรแกรมคำนวณค่าความเสถียรเชิงตรรกะของคลาส แสดงได้ดังรูปที่ 28 จากนั้นทำการเลือกไฟล์ซอร์ซโค้ดที่ต้องการคำนวณค่าความเสถียรเชิงตรรกะของคลาส เลือกเมนู Add file ดังแสดงในรูปที่ 28 จากนั้นจะปรากฏหน้าต่าง Open folder เพื่อทำการเลือกไฟล์ซอร์ซโค้ดภาษาจาวาตามต้องการ ดังแสดงในรูปที่ 29



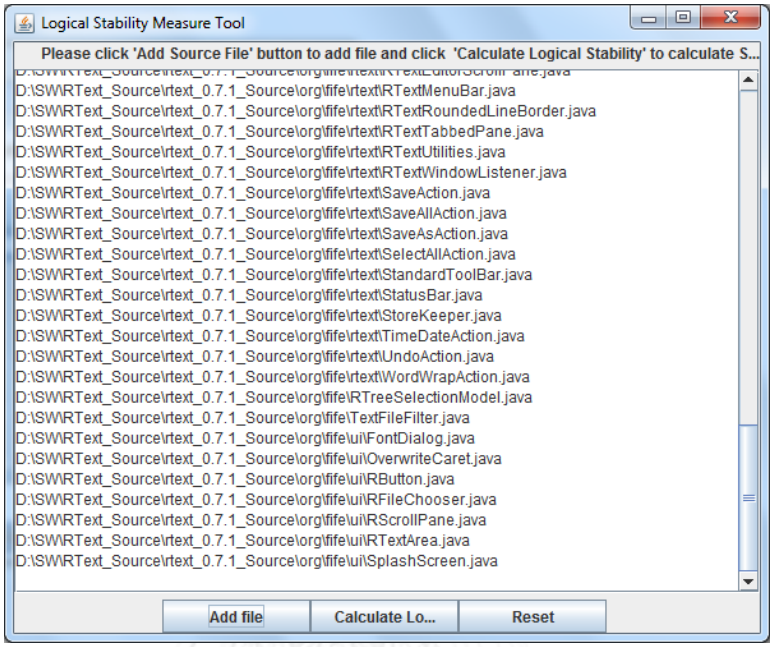
รูปที่ 28 หน้าต่างของโปรแกรม Logical Stability Measure Tool



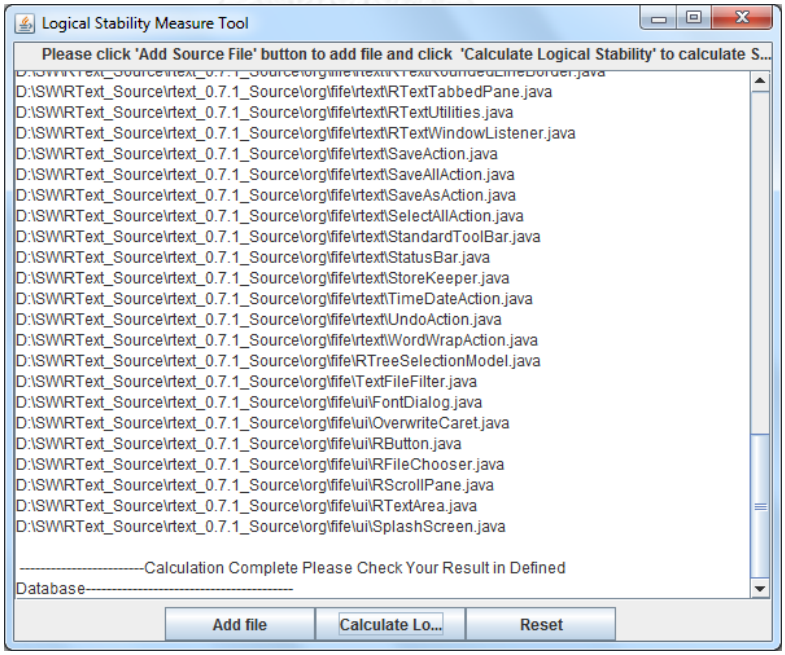
รูปที่ 29 หน้าต่างเลือกไฟล์ซอร์ซโค้ดที่ต้องการคำนวณค่าความเสถียรเชิงตรรกะของคลาส

เมื่อทำการเลือกไฟล์ซอร์ซโค้ดเรียบร้อยแล้ว โปรแกรมจะแสดงรายการไฟล์ทั้งหมดในหน้าต่าง Logical stability measure tool ดังแสดงในรูปที่ 30 จากนั้นทำการเลือกเมนู Calculate Logical Stability of Selected file โปรแกรมจะทำการคำนวณค่าความเสถียรเชิงตรรกะของคลาส

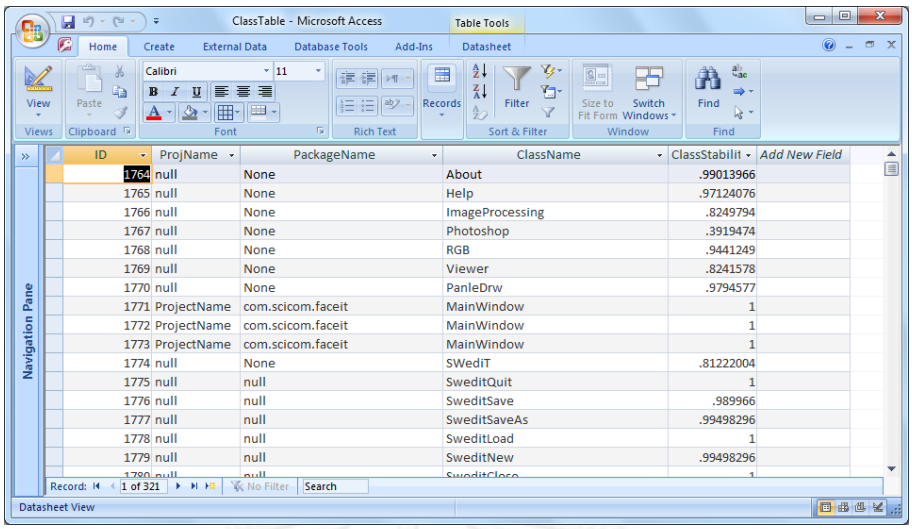
แล้วทำการบันทึกอัตโนมัติลงในฐานข้อมูลที่ได้ทำการกำหนดไว้เบื้องต้น เมื่อเสร็จสิ้นโปรแกรมจะแสดงข้อความให้ทราบดังแสดงในรูปที่ 31 ผลลัพธ์ค่าความเสถียรเชิงตรรกะของคลาสจะแสดงในฐานข้อมูลดังแสดงในรูปที่ 32



รูปที่ 30 หน้าต่างแสดงไฟล์ที่นำมาประมวลผลเพื่อคำนวณค่าความเสถียรเชิงตรรกะของคลาส



รูปที่ 31 หน้าต่างแสดงข้อความเมื่อมีการคำนวณและบันทึกลงในฐานข้อมูลเรียบร้อยแล้ว



ID	ProjName	PackageName	ClassName	ClassStabilit	Add New Field
1764	null	None	About	.99013966	
1765	null	None	Help	.97124076	
1766	null	None	ImageProcessing	.8249794	
1767	null	None	Photoshop	.3919474	
1768	null	None	RGB	.9441249	
1769	null	None	Viewer	.8241578	
1770	null	None	PanleDrw	.9794577	
1771	ProjectName	com.scicom.faceit	MainWindow	1	
1772	ProjectName	com.scicom.faceit	MainWindow	1	
1773	ProjectName	com.scicom.faceit	MainWindow	1	
1774	null	None	SWediT	.81222004	
1775	null	null	SweditQuit	1	
1776	null	null	SweditSave	.989966	
1777	null	null	SweditSaveAs	.99498296	
1778	null	null	SweditLoad	1	
1779	null	null	SweditNew	.99498296	

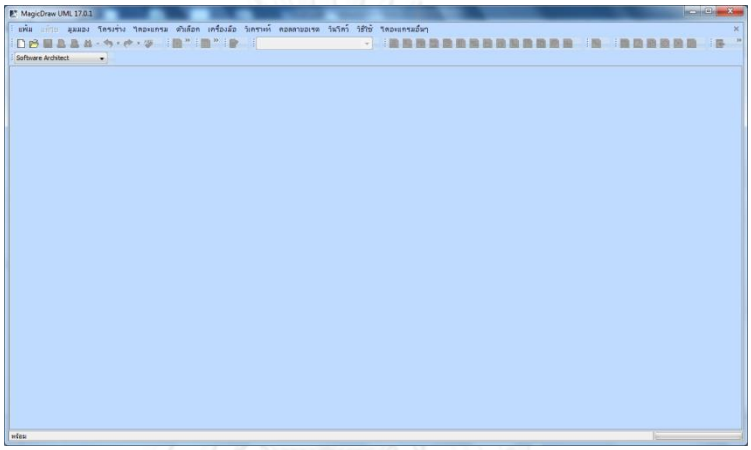
รูปที่ 32 หน้าต่างแสดงตัวอย่างผลการคำนวณค่าความเสถียรเชิงตรรกะของคลาส

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

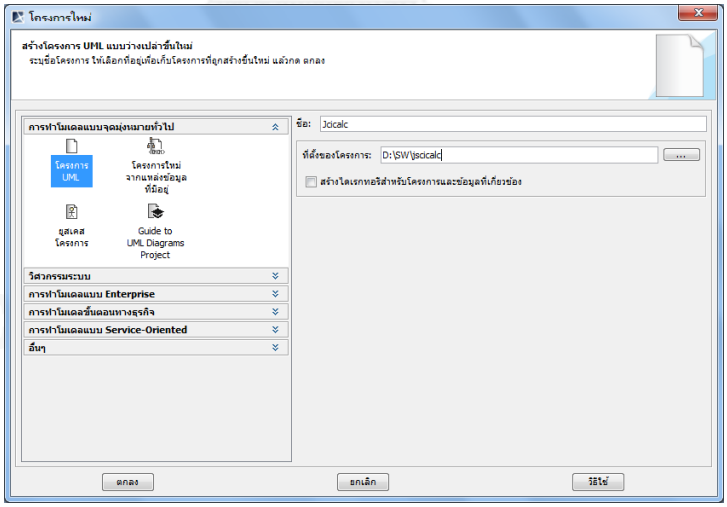
ภาคผนวก ข

การแปลงซอร์ซโค้ดเป็นแผนภาพคลาสและแผนภาพลำดับ

การแปลงกลับจากซอร์ซโค้ดเป็นแผนภาพคลาสและแผนภาพลำดับ ในงานวิจัยนี้ใช้โปรแกรม MagicDraw UML เวอร์ชัน 17.0.1 เมื่อเปิดโปรแกรมจะแสดงหน้าต่างหลักของโปรแกรมดังรูปที่ 33 จากนั้นทำการสร้างโครงการใหม่ โดยเลือกจากเมนูแฟ้ม และเลือกสร้างโครงการใหม่ โปรแกรมจะแสดงหน้าต่างสร้างโครงการใหม่ดังรูปที่ 34 เลือกโครงการ UML กรอกชื่อโครงการ และที่ตั้งของโครงการหรือที่อยู่ของไฟล์ซอร์ซโค้ดที่จะทำการแปลงกลับเป็นแผนภาพคลาส



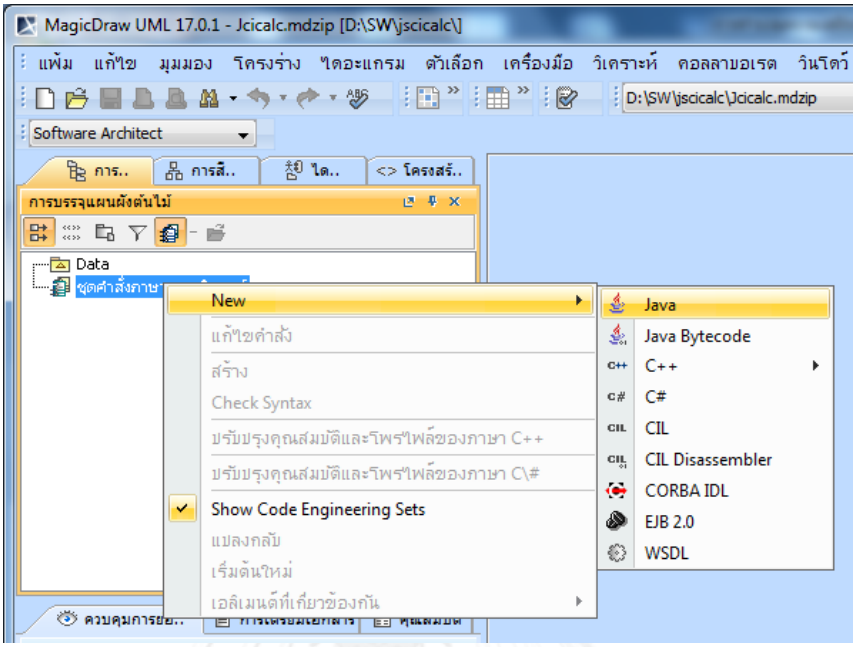
รูปที่ 33 หน้าต่างหลักของโปรแกรม MagicDraw UML



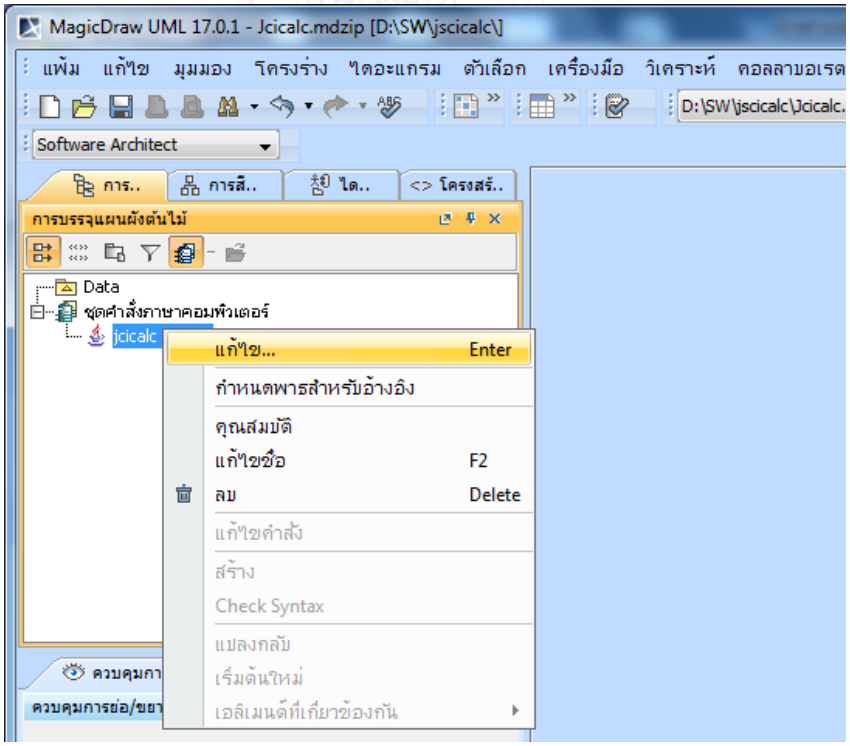
รูปที่ 34 หน้าต่างสร้างโครงการใหม่ของโปรแกรม MagicDraw UML

เมื่อสร้างโครงการใหม่เสร็จเรียบร้อยคลิกขวาที่ชุดคำสั่งภาษาคอมพิวเตอร์แล้วเลือกสร้างชุดคำสั่งภาษาเป็นภาษาจาวา ดังรูปที่ 35 และทำการเพิ่มไฟล์ซอร์ซโค้ดโดยเลือกที่เมนูแก้ไขดังรูปที่ 36 และจัดการเพิ่มไฟล์ซอร์ซโค้ดที่ต้องการแปลงกลับเป็นแผนภาพคลาสดังรูปที่ 37 เมื่อดำเนินการ

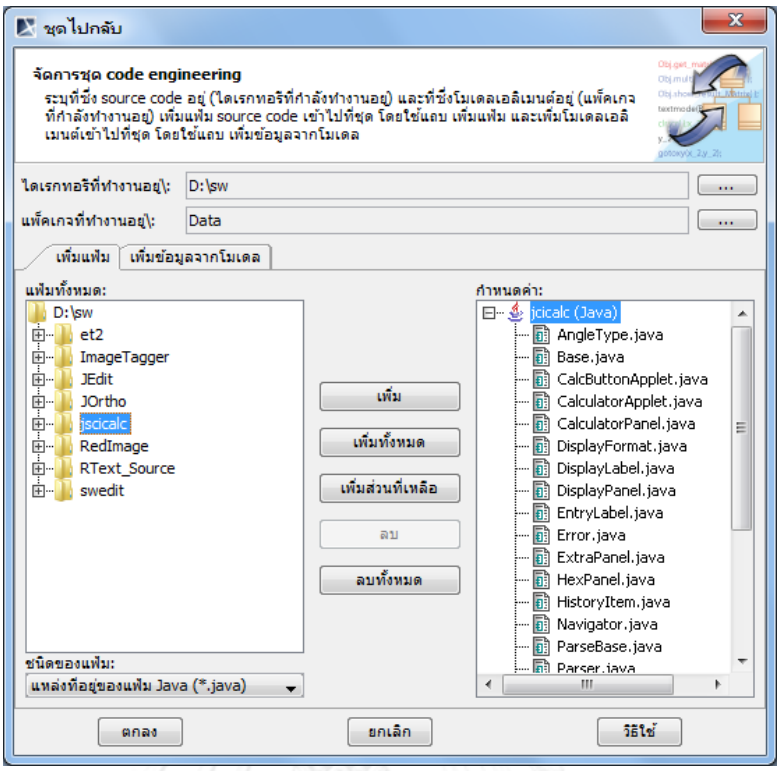
เสร็จเรียบร้อยก็จะปรากฏแสดงรายการไฟล์ซอร์ซโค้ดภาษาจาวาที่ต้องการแปลงกลับเป็นแผนภาพคลาสดังรูปที่ 38



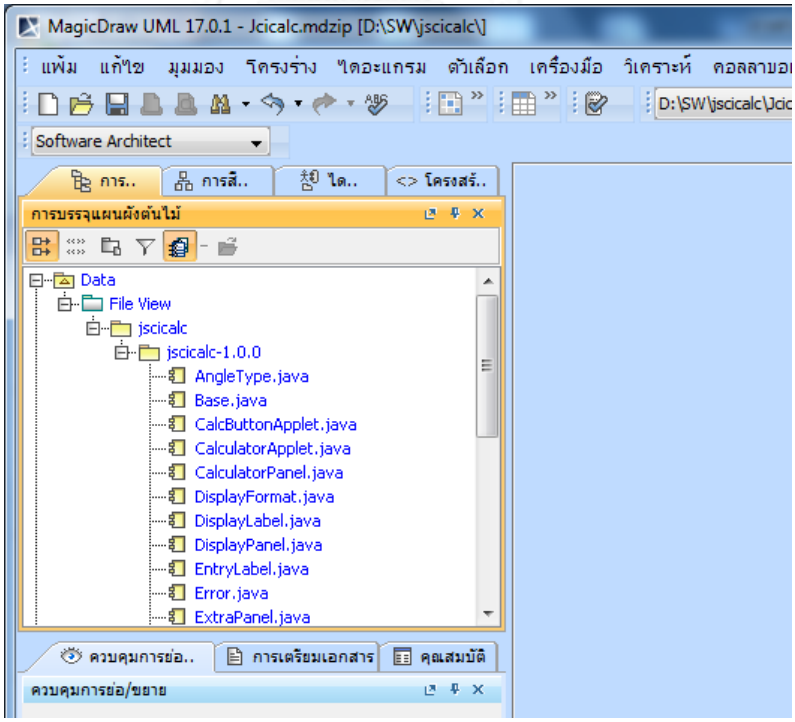
รูปที่ 35 หน้าต่างแสดงการสร้างชุดคำสั่งภาษาคอมพิวเตอร์



รูปที่ 36 หน้าต่างแสดงการแก้ไขชุดคำสั่งภาษาคอมพิวเตอร์

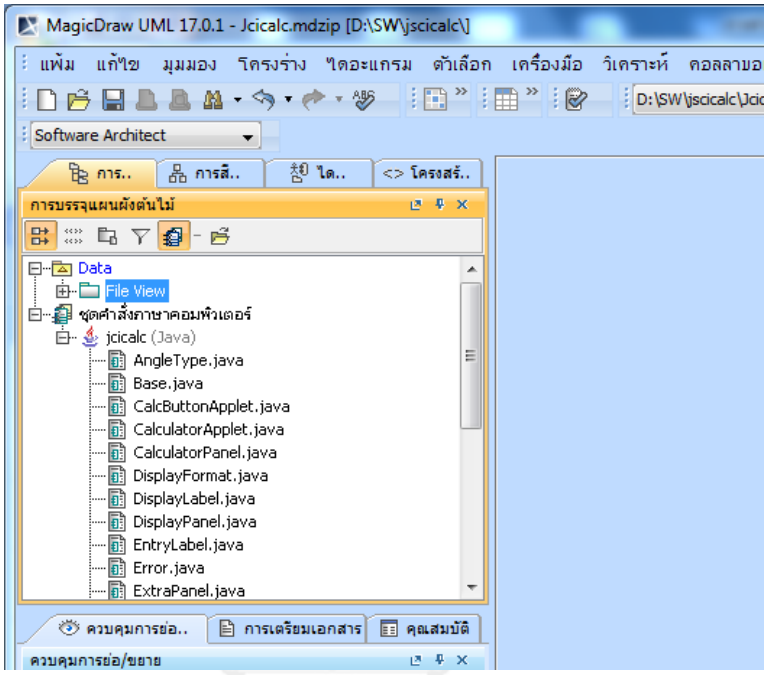


รูปที่ 37 หน้าต่างแสดงการจัดการซอร์ซโค้ดที่ต้องการแปลงกลับเป็นแผนภาพคลาส

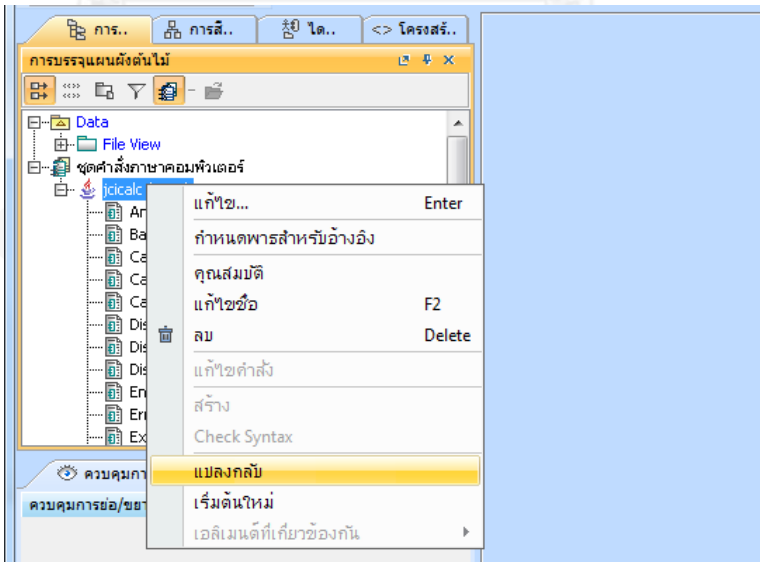


รูปที่ 38 หน้าต่างแสดงรายการไฟล์ซอร์ซโค้ดภาษาจาวาที่ต้องการแปลงกลับเป็นแผนภาพคลาส

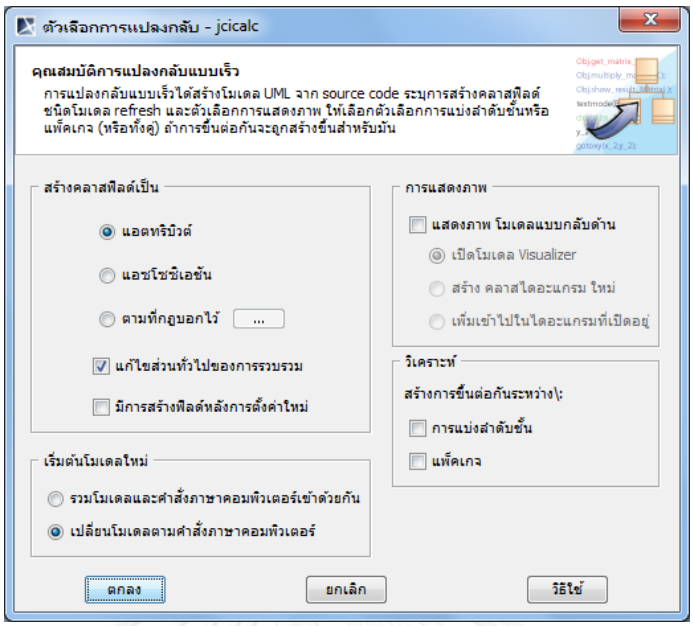
จากนั้นทำการแปลงกลับเป็นแผนภาพคลาสโดยไปที่รายการไฟล์ซอร์ซโค้ดภาษาจาวาในชุดคำสั่งภาษาคอมพิวเตอร์ในรูปที่ 39 และเลือกเมนูแปลงกลับดังรูปที่ 40 แล้วโปรแกรมจะปรากฏหน้าต่างตัวเลือกการแปลงกลับดังรูปที่ 41 จากนั้นทำการตั้งค่าของตัวเลือกการแปลงกลับดังรูปที่ 42 เมื่อเสร็จสิ้นขั้นตอนดังกล่าว โปรแกรมจะปรากฏหน้าต่างแสดงแผนภาพคลาสที่ได้จากการแปลงกลับดังรูปที่ 43



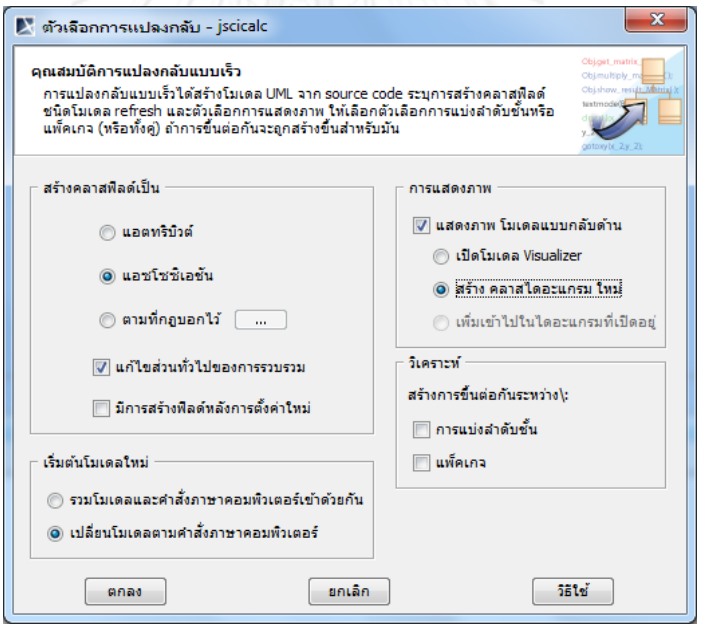
รูปที่ 39 หน้าต่างแสดงรายการไฟล์ซอร์ซโค้ดภาษาจาวาในชุดคำสั่งภาษาคอมพิวเตอร์



รูปที่ 40 หน้าต่างแสดงการแปลงกลับจากซอร์ซโค้ดเป็นแผนภาพคลาส



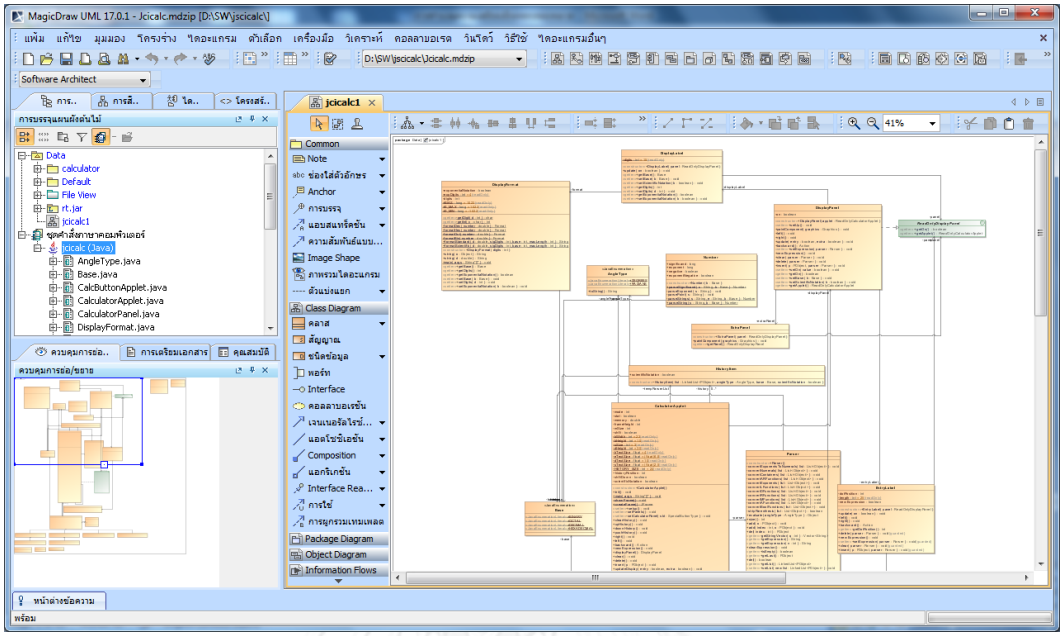
รูปที่ 41 หน้าต่างตัวเลือกการแปลงกลับ



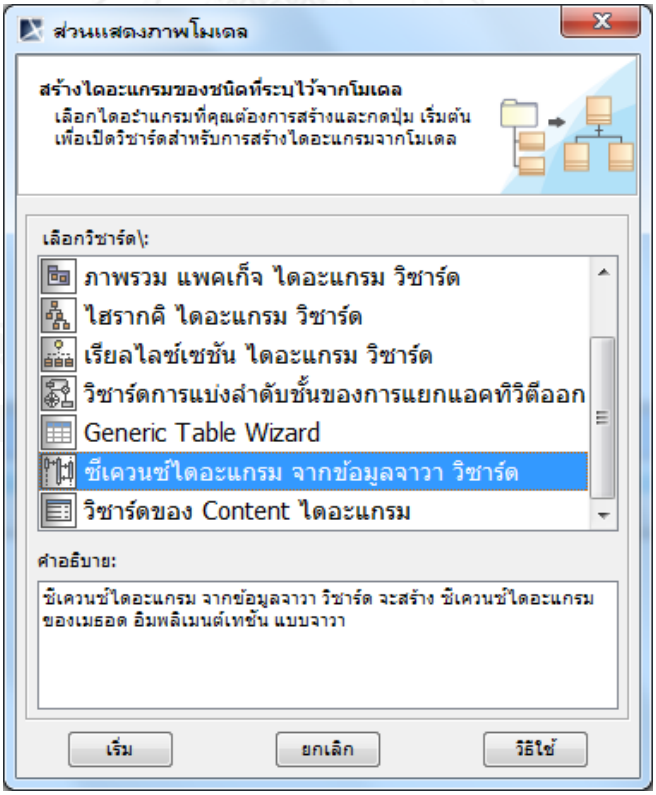
รูปที่ 42 หน้าต่างแสดงการตั้งค่าของตัวเลือกการแปลงกลับ

การแปลงกลับเป็นแผนภาพลำดับทำได้โดยการเลือกที่เมธอดที่ต้องการแล้วเลือกเมนูแปลงกลับ เช่นเดียวกับการแปลงกลับเป็นแผนภาพคลาส จากนั้นโปรแกรมจะแสดงหน้าต่างตัวเลือกการแปลงกลับ แล้วเลือกการแสดงภาพโดยเปลี่ยนจากสร้างคลาสไดอะแกรมใหม่เป็น เปิดโมเดล Visualizes ซึ่งโปรแกรมจะปรากฏหน้าต่างดังรูปที่ 44 โดยเลือกที่เมนูซีเคว้นไดอะแกรม เมื่อทำตามขั้นตอนดังกล่าวแล้วจากนั้นโปรแกรมจะแสดงแผนภาพลำดับที่ได้จากการแปลงกลับดังรูปที่ 45

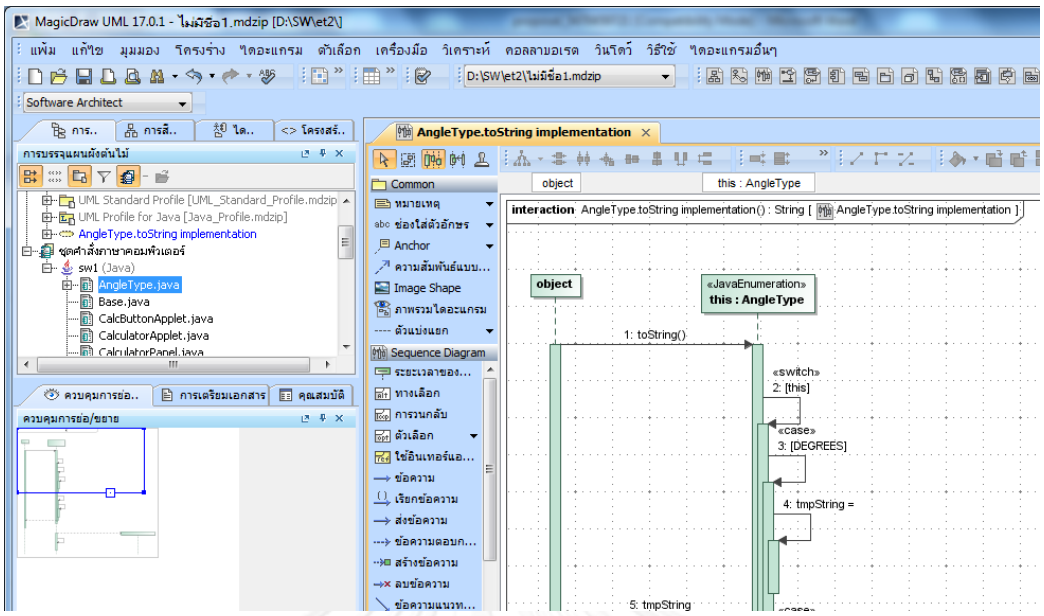
จะต้องทำการแปลงกลับทุกๆเมธอดในทุกๆคลาส เพื่อให้ได้รายละเอียดมากที่สุด ซึ่งแผนภาพคลาสและแผนภาพลำดับที่ได้จะต้องบันทึกไว้ในไฟล์ xml ดังรูปที่ 46



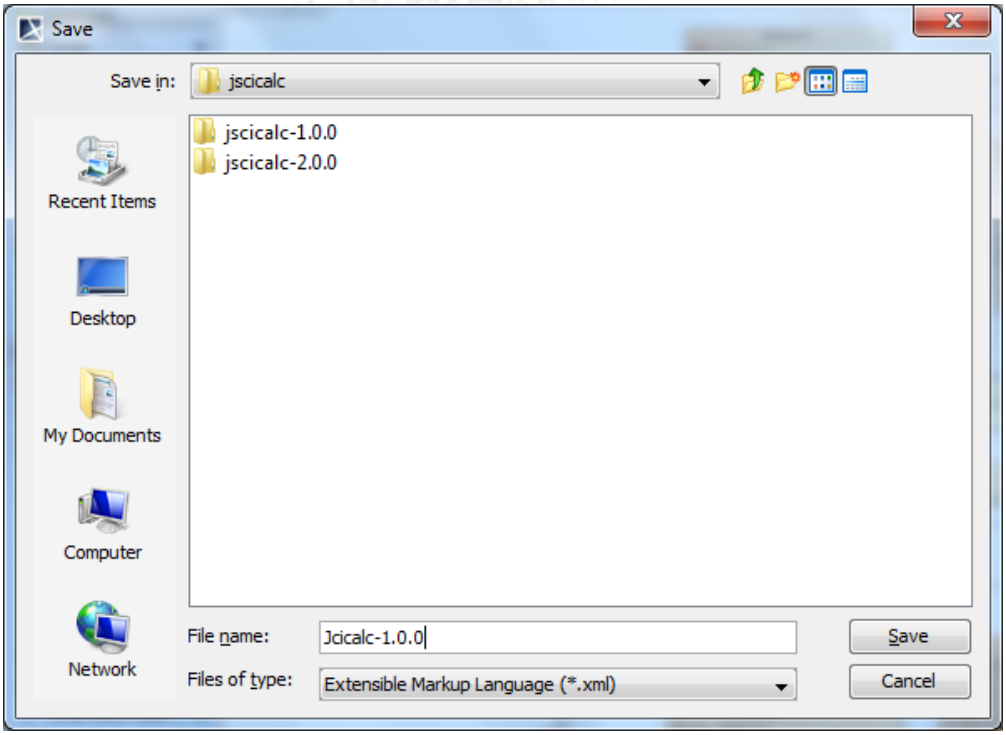
รูปที่ 43 หน้าต่างแสดงแผนภาพคลาสที่ได้จากการแปลงกลับ



รูปที่ 44 หน้าต่างแสดงการแปลงกลับแผนภาพลำดับ



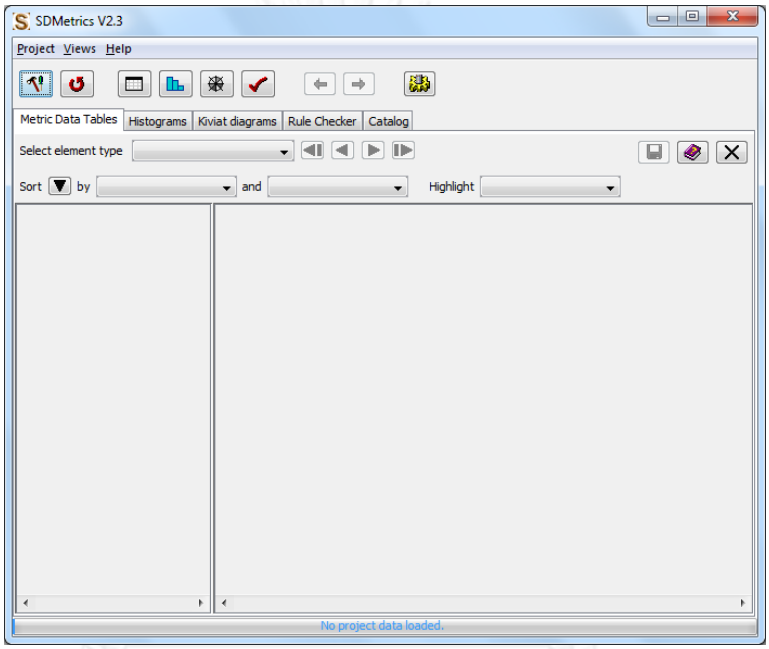
รูปที่ 45 หน้าต่างแสดงแผนภาพลำดับที่ได้จากการแปลงกลับ



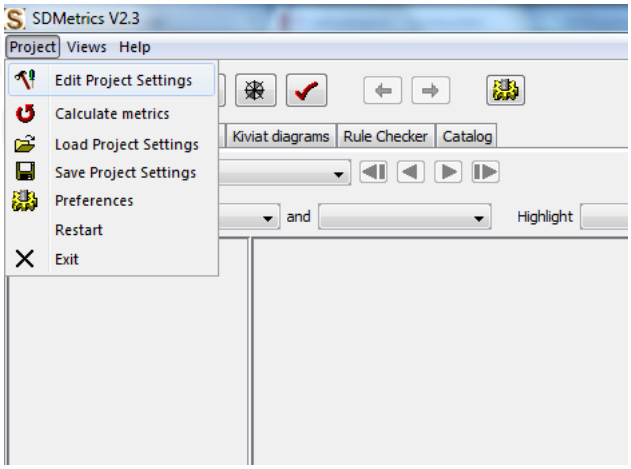
รูปที่ 46 หน้าต่างบันทึกโครงการเป็นไฟล์ xml

ภาคผนวก ค
การวัดค่าตัววัดแผนภาพ

การวัดค่าตัววัดแผนภาพใช้โปรแกรม SDMetrics เวอร์ชัน 2.3 ซึ่งโปรแกรมนี้จะต้องนำเข้าไฟล์ xml ซึ่งจากงานวิจัยนี้ได้ทำการบันทึกมาจากโปรแกรม MagicDraw UML เรียบร้อยแล้ว เมื่อเปิดโปรแกรม SDMetrics จะปรากฏหน้าต่างหลักของโปรแกรมดังรูปที่ 47 จากนั้นเลือกที่ Project และเลือก Edit Project Settings ดังรูปที่ 48 เพื่อตั้งค่าโครงการที่จะนำเข้าไฟล์เพื่อคำนวณค่าตัววัดคุณลักษณะ

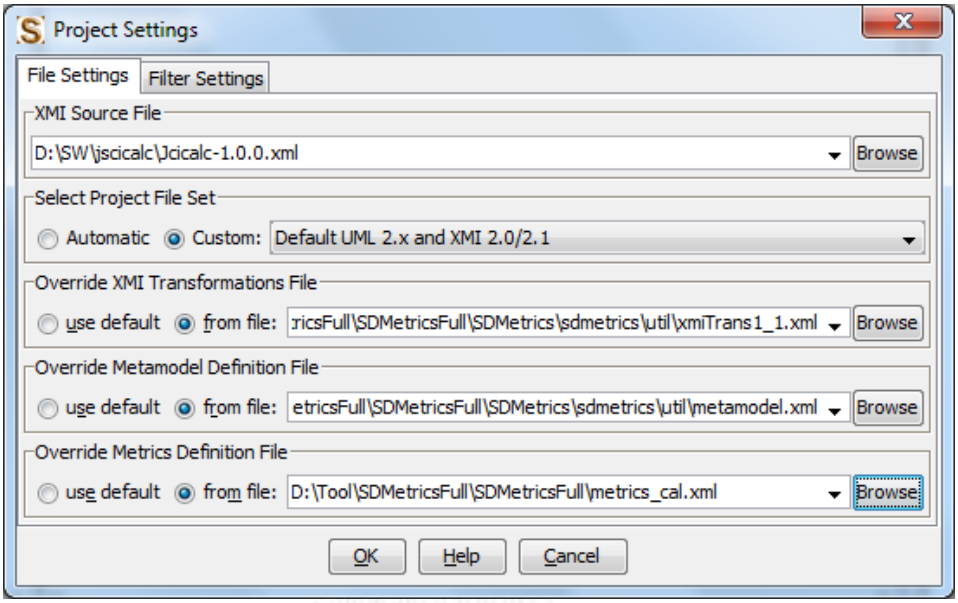


รูปที่ 47 หน้าต่างหลักของโปรแกรม SDMetrics เวอร์ชัน 2.3

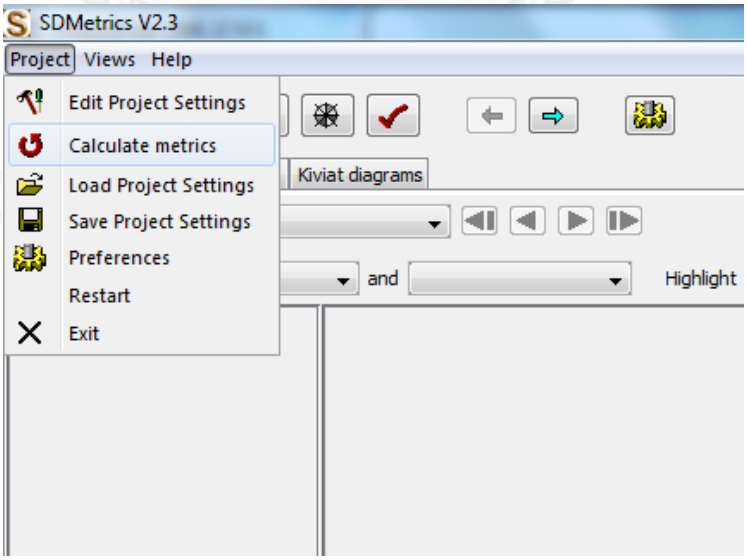


รูปที่ 48 หน้าต่างแสดงการตั้งค่าโครงการ

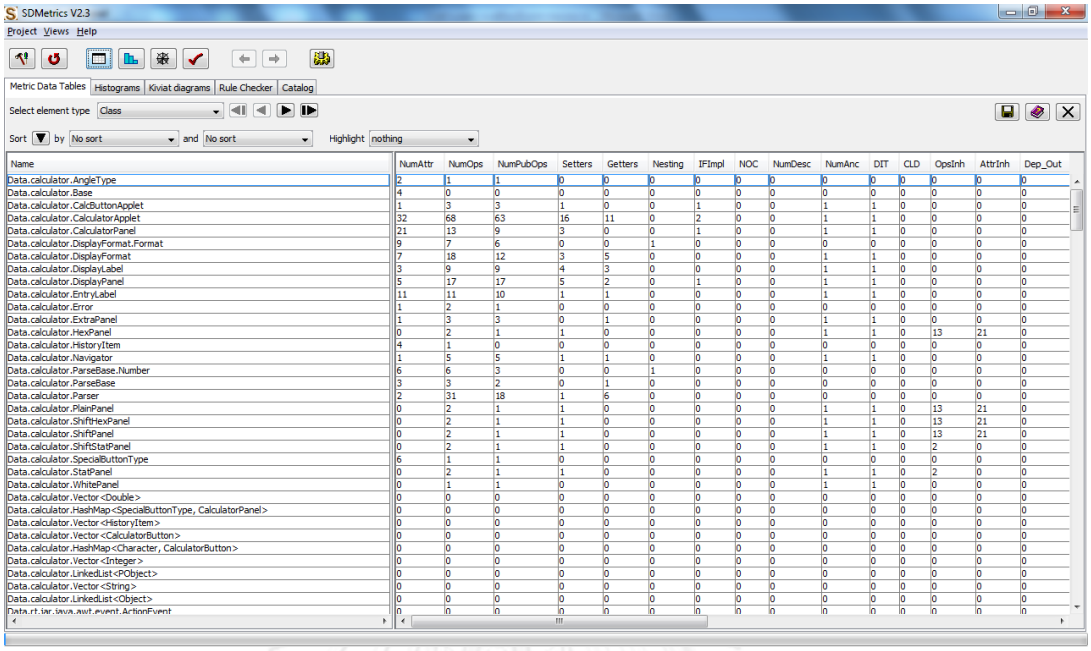
จากนั้นทำการตั้งค่าที่หน้าต่าง Project Settings โดยเลือกไฟล์ xml ที่ต้องการ ทำการตั้งค่า Select Project File Set เป็น Default UML 2.x and XMI 2.0/2.1 ในส่วนของ Override XMI Transformations File, Override Metamodel Definitions File, Override Metrics Definition File ตั้งค่าโดยเลือกไฟล์ที่มีคุณสมบัติที่ต้องการนำมาใช้ดังแสดงในรูปที่ 49 เมื่อเสร็จสิ้นการตั้งค่าแล้ว เลือก OK จากนั้นเลือกที่เมนู Project และเลือก Calculate metrics ดังแสดงในรูปที่ 50 โปรแกรมจะแสดงผลลัพธ์เป็นค่าของตัววัดต่างๆดังรูปที่ 51 ซึ่งจะสามารถบันทึกได้ซึ่งในที่นี้เลือกบันทึกเป็นไฟล์ csv ซึ่งแสดงดังรูปที่ 52



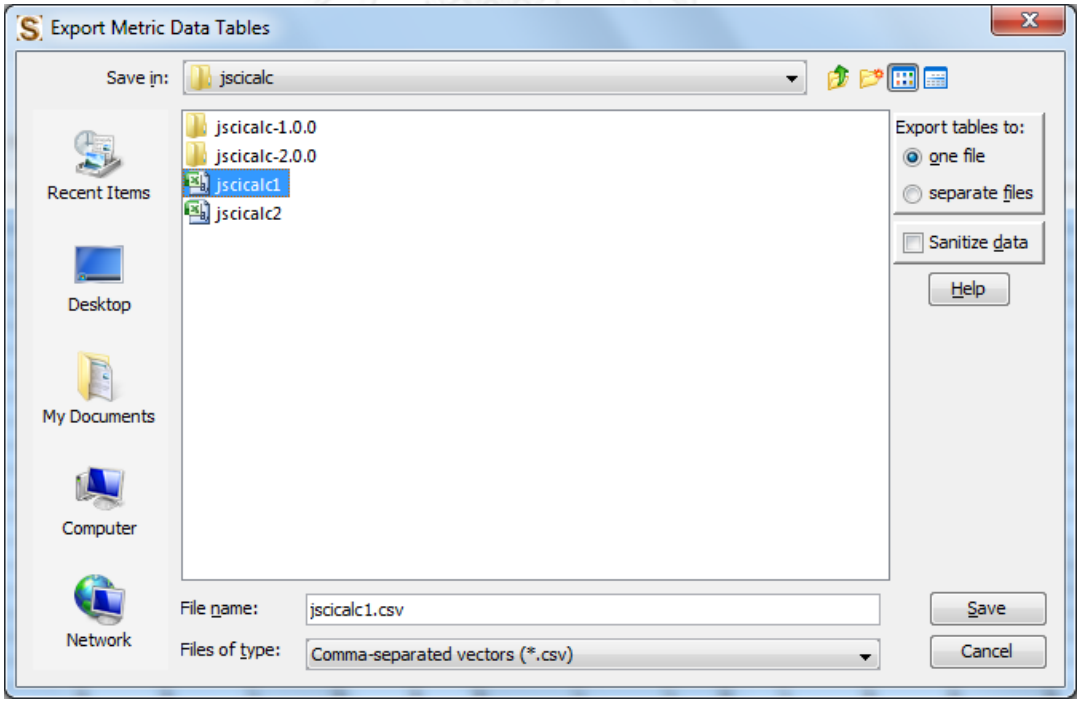
รูปที่ 49 หน้าต่าง Project Settings



รูปที่ 50 หน้าต่างแสดงเมนูการคำนวณค่าตัววัดแผนภาพ



รูปที่ 51 หน้าต่างแสดงค่าตัววัดแผนภาพ



รูปที่ 52 หน้าต่างแสดงการบันทึกค่าตัววัดแผนภาพเป็นไฟล์ csv

ภาคผนวก ง
การใช้งานเครื่องมือประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัล

เครื่องมือนี้เป็นเครื่องมือที่พัฒนาจากงานวิจัยที่ได้นำเสนอโมเดลในการประมาณค่าความเสถียรของการออกแบบซอฟต์แวร์ ในการพัฒนาแบบอินคริเมนทัลโดยใช้แผนภาพคลาสและแผนภาพลำดับ ซึ่งงานวิจัยนี้ได้ทำการเปรียบเทียบผลการประมาณค่าความเสถียรเชิงตรรกะของคลาสในแต่ละคลาสเมื่อมีการเพิ่มการออกแบบของทุกๆอินคริเมนต์ โดยเครื่องมือนี้จะมีการนำเข้าไฟล์ 2 ประเภท คือ xml และ cvs ในแต่ละประเภทจะต้องนำเข้า 2 ไฟล์ คือ ไฟล์ของอินคริเมนต์ปัจจุบัน และไฟล์ที่มีการเพิ่มอินคริเมนต์ถัดไป ซึ่งจะแบ่งการทำงานของโปรแกรมออกเป็น 3 ขั้นตอน คือ

1) ขั้นตอนการนำเข้าไฟล์

เมื่อเปิดโปรแกรมจะแสดงหน้าต่างหลักดังรูปที่ 53 จากนั้นผู้ใช้ใส่ข้อมูลตามหมายเลขในรูปที่ 54 ดังนี้

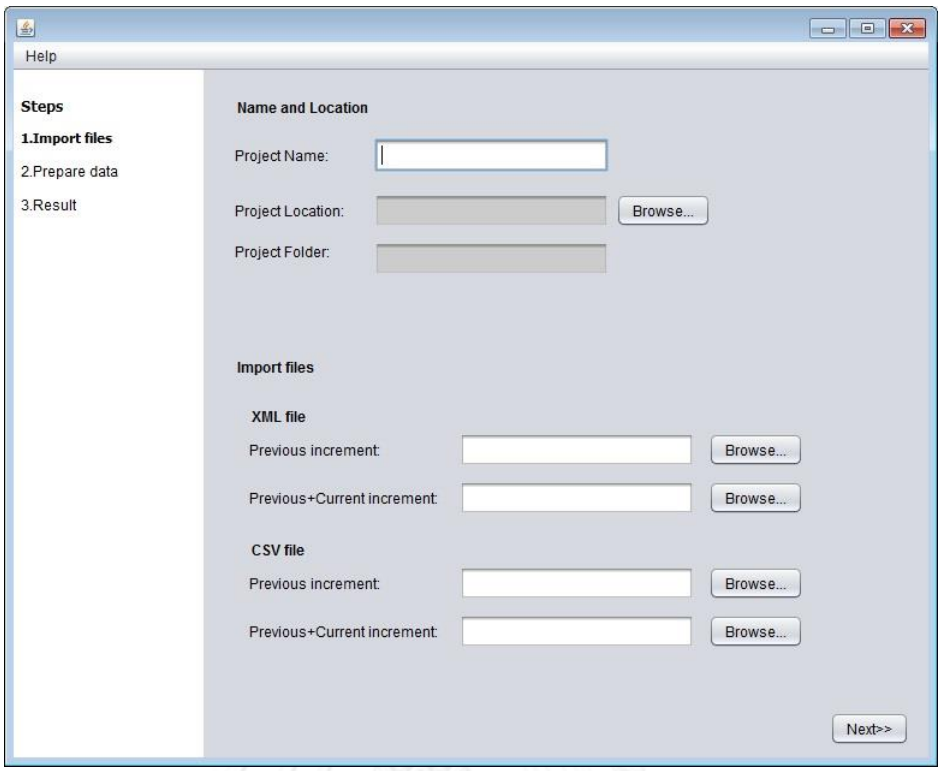
หมายเลข 1 Project Name ใส่ชื่อแฟ้มงานที่ผู้ใช้ต้องการ

หมายเลข 2 Project Location คลิกที่ปุ่ม Browse เพื่อเลือกที่ตั้งของแฟ้มงาน จากนั้นในช่อง Project Folder จะแสดงตำแหน่งของแฟ้มงานที่ถูกบันทึกผลในขั้นตอนสุดท้าย

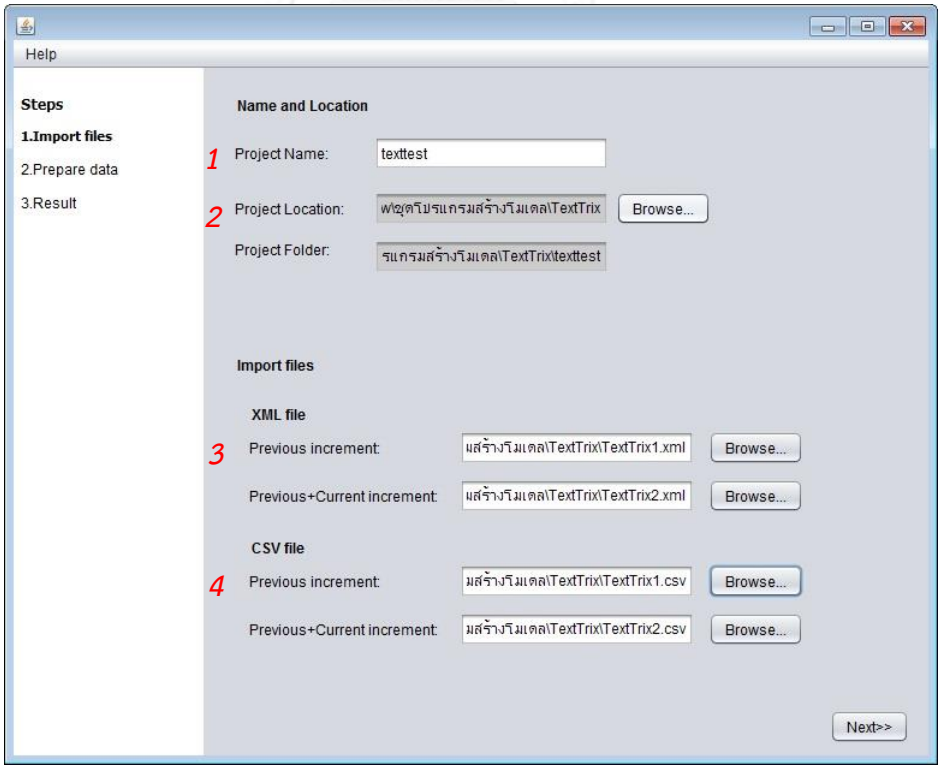
หมายเลข 3 Import XML file ซึ่งเป็นไฟล์ของแผนภาพคลาสและแผนภาพลำดับ คลิกที่ปุ่ม Browse เพื่อเลือกที่ตั้งของไฟล์โดยทำการเลือกไฟล์อินคริเมนต์ก่อนหน้า (previous increment) และเมื่อเพิ่มอินคริเมนต์ปัจจุบัน (previous + current increment)

หมายเลข 4 Import CSV file ซึ่งเป็นไฟล์ของค่าตัววัดแผนภาพ คลิกที่ปุ่ม Browse เพื่อเลือกที่ตั้งของไฟล์โดยทำการเลือกไฟล์อินคริเมนต์ก่อนหน้า (previous increment) และเมื่อเพิ่มอินคริเมนต์ปัจจุบัน (previous + current increment)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 53 หน้าต่างหลักของเครื่องมือประมาณค่าความเสถียร



รูปที่ 54 ตัวอย่างขั้นตอนการนำเข้าไฟล์

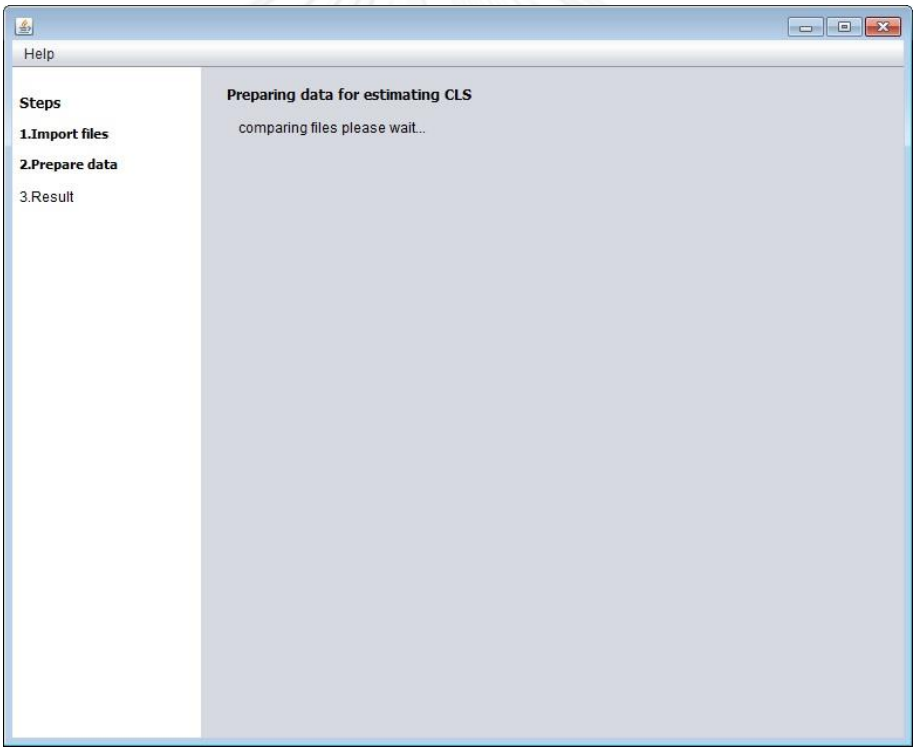
1) ขั้นตอนการเตรียมข้อมูล

ในขั้นตอนนี้โปรแกรมจะทำการตรวจสอบไฟล์และเตรียมข้อมูล เพื่อการนำไปคำนวณค่าประมาณความเสถียร หากไฟล์ที่นำเข้าไม่สามารถนำไปคำนวณได้โปรแกรมจะแสดงข้อความให้นำเข้าไฟล์ใหม่อีกครั้ง เมื่อนำเข้าไฟล์แล้วระหว่างการประมวลผลโปรแกรมจะแสดงข้อความดังรูปที่ 55 เมื่อเสร็จสิ้นขั้นตอนนี้โปรแกรมจะแสดงข้อความดังรูปที่ 56 ซึ่งจะประกอบด้วย

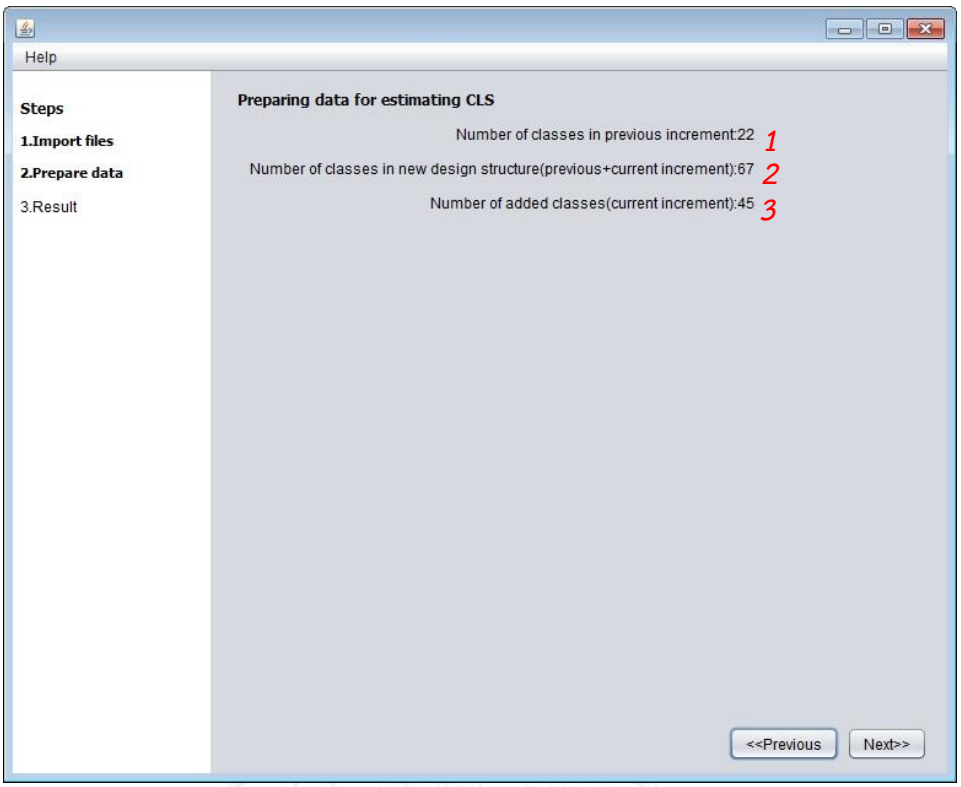
หมายเลข 1 Number of classes in previous increment แสดงจำนวนของคลาสในอินคริเมนต์ปัจจุบัน

หมายเลข 2 Number of classes in new design structure (previous + current increment) แสดงจำนวนของคลาสในอินคริเมนต์ปัจจุบันรวมกับอินคริเมนต์ก่อนหน้า

หมายเลข 3 Number of added classes (current increment) แสดงจำนวนของคลาสในอินคริเมต์ปัจจุบัน ที่ถูกเพิ่มเข้าไปในอินคริเมนต์ก่อนหน้า จากนั้นคลิกปุ่ม Next



รูปที่ 55 ตัวอย่างขั้นตอนการตรวจสอบไฟล์และเตรียมข้อมูล



รูปที่ 56 ตัวอย่างเมื่อเสร็จสิ้นขั้นตอนการตรวจสอบไฟล์และเตรียมข้อมูล

2) ขั้นตอนการแสดงผล

ขั้นตอนที่ 3 เป็นส่วนของการแสดงผล โปรแกรมจะแสดงผลค่าประมาณความเสถียรอยู่ในรูปของกราฟและตารางดังรูปที่ 57 ซึ่งประกอบด้วย

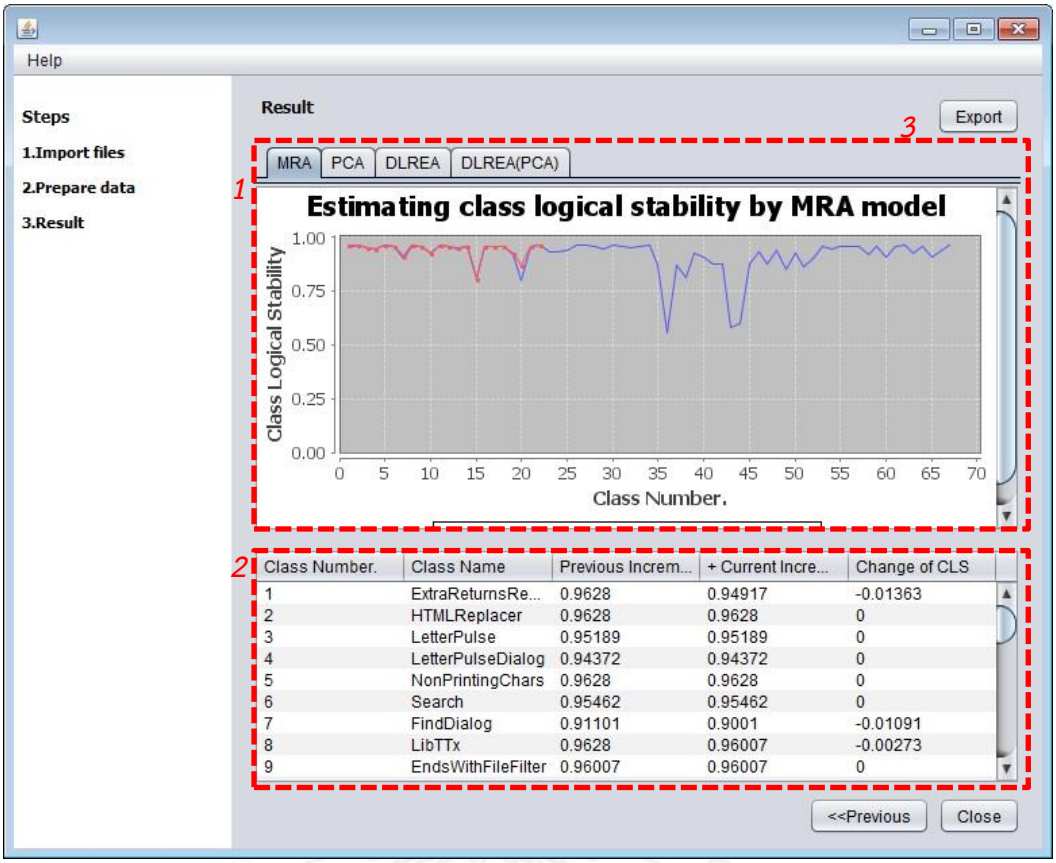
หมายเลข 1 เป็นส่วนแสดงกราฟเส้นเปรียบเทียบค่าประมาณความเสถียรของอินคริเมนต์ก่อนหน้า (Previous increment) และเมื่อเพิ่มอินคริเมนต์ปัจจุบัน (previous + current increment) ซึ่งมีรูปแบบวิธีในการประมาณค่า 4 แบบ คือ MRA, PCA, DLREA, DLREA(PCA)

หมายเลข 2 เป็นส่วนของตารางแสดงค่าประมาณความเสถียรของอินคริเมนต์ก่อนหน้า (previous increment) และเมื่อเพิ่มอินคริเมนต์ปัจจุบัน (previous + current increment) ซึ่งแสดงผลตามการคลิกเลือกรูปแบบวิธีในการประมาณค่า 4 แบบ คือ MRA, PCA, DLREA, DLREA(PCA)

หมายเลข 3 ปุ่ม Export เป็นการส่งออกไฟล์หรือบันทึกไฟล์ค่าประมาณความเสถียรเชิงตรรกะของคลาส ซึ่งจะบันทึกตามชื่อแฟ้มงาน และที่ตั้งของแฟ้มงานที่ผู้ใช้เลือกไว้ในขั้นตอนที่ 1 ดังรูปที่ 58 ภายแฟ้มงานจะประกอบด้วยไฟล์ดังนี้

1) ไฟล์ตารางค่าประมาณความเสถียรของรูปแบบวิธีในการประมาณค่าทั้ง 4 แบบ (จำนวน 1 ไฟล์)

2) ไฟล์ภาพกราฟเส้นเปรียบเทียบค่าประมาณความเสถียรของรูปแบบวิธีในการประมาณค่าทั้ง 4 แบบ (จำนวน 4 ไฟล์)



รูปที่ 57 ตัวอย่างของการแสดงผลกราฟและตารางค่าประมาณความเสถียร

รูปที่ 58 ไฟล์ทั้งหมดที่ได้จากการบันทึกไฟล์ค่าประมาณความเสถียรเชิงตรรกะของคลาส

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวอลิสา แสงภู่วงษ์ เกิดเมื่อวันที่ 23 พฤษภาคม พ.ศ. 2532 ณ จังหวัดสุพรรณบุรี วุฒิการศึกษาระดับปริญญาตรี หลักสูตรวิทยาศาสตร์บัณฑิต สาขาวิชาการพัฒนาซอฟต์แวร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สำเร็จการศึกษาในปี พ.ศ. 2553 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2554

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University