รีแฟคทอริงแบบเอ็กซ์แทรกต์แพ็กเกจโดยใช้การตรวจหาคอมมิวนิตีสำหรับแพ็กเกจขนาดใหญ่เกิน

นายบุณยไชย จันทร์เทียน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

EXTRACT-PACKAGE REFACTORING USING COMMUNITY DETECTION FOR TOO LARGE PACKAGES

Mr. Bunyachai Chantian

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2019

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	รีแฟคทอริงแบบเอ็กซ์แทรกต์แพ็กเกจโดยใช้การตรวจหาคอมมิวนิตีสำหรับแพ็กเกจขนาดใหญ่เกิน
โดย	นายบุณยไชย จันทร์เทียน
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

..... กรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.บัณฑิต ฐานะโสภณ)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บุณยไชย จันทร์เทียน : รีแฟคทอริงแบบเอ็กซ์แทรกต์แพ็กเกจโดยใช้การตรวจหาคอมมิวนิตีสำหรับแพ็กเกจขนาดใหญ่เกิน. (EXTRACT-PACKAGE REFACTORING USING COMMUNITY DETECTION FOR TOO LARGE PACKAGES) อ.ที่ปรึกษาหลัก : รศ. ดร.พรศิริ หมื่นไชยศรี

อาคิเทกเจอร์สเมลเป็นการออกแบบโครงสร้างสถาปัตยกรรมที่ไม่ดีซึ่งเปิดเผยถึงผลกระทบด้านลบต่อคุณภาพซอฟต์แวร์และยังเพิ่มความเสี่ยงในการเกิดข้อบกพร่องตามมาในอนาคต งานวิจัยที่ผ่านมาได้กำหนดและนำเสนออาคิเทกเจอร์สเมลไว้ในหลายรูปแบบและแต่ละรูปแบบส่งผลกระทบต่อคุณภาพซอฟต์แวร์ในมุมที่แตกต่างกัน กระบวนการรีแฟคทอริงเชิงสถาปัตยกรรมเป็นกระบวนการที่ใช้กำจัดอาคิเทกเจอร์สเมลและปรับปรุงคุณภาพของระบบโดยไม่เกิดการเปลี่ยนแปลงพฤติกรรมการทำงานของระบบ อย่างไรก็ตามการกำจัดอาคิเทกเจอร์สเมลรูปแบบหนึ่งอาจก่อให้เกิดอาคิเทกเจอร์สเมลในรูปแบบอื่นขึ้นในระบบ ซึ่งส่งผลกระทบโดยตรงต่อคุณภาพซอฟต์แวร์

งานวิจัยนี้นำเสนอวิธีการในการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินซึ่งเป็นรูปแบบหนึ่งของอาคิเทกเจอร์สเมล โดยใช้การตรวจหาคอมมิวนิตีในกระบวนการรีแฟคทอริงแบบเอ็กซ์แทรกต์แพ็กเกจ กระบวนการนี้ใช้สำหรับการแยกแพ็กเกจที่ถูกตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินให้มีขนาดเล็กลง และหลีกเลี่ยงผลกระทบอื่น ๆ จากอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพา โดยมุ่งเน้นที่อาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

วิธีการที่นำเสนอนี้ได้ให้แนวทางที่เป็นไปได้ที่จะแก้ไขผลกระทบจากอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน ซึ่งสามารถประเมินผลโดยการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบและค่าตัวชี้วัดซอฟต์แวร์ ก่อนและหลังใช้กระบวนการรีแฟคทอริง นอกจากนี้ในงานวิจัยนี้ยังพัฒนาเครื่องมือบนพื้นฐานของวิธีการที่นำเสนอเพื่อที่จะเสนอวิธีการรีแฟคทอริงและสนับสนุนนักพัฒนาเพื่อปรับปรุงคุณภาพซอฟต์แวร์โดยการแก้ไขผลกระทบจากอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินในระหว่างขั้นตอนการพัฒนาและการบำรุงรักษา

Chulalongkorn University

สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่อนิสิต .....
ปีการศึกษา	2562	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6070934321 : MAJOR COMPUTER SCIENCE

KEYWORD: Architecture Smells, Architecture Refactoring, Extract Package Refactoring, Community Detection

Bunyachai Chantian : EXTRACT-PACKAGE REFACTORING USING COMMUNITY DETECTION FOR TOO LARGE PACKAGES. Advisor: Assoc. Prof. Pornsiri Muenchaisri, Ph.D.

Architecture smells are poor design of architectural structures which reveal negative impacts on software quality and increase risks of bugs in the future. Previous research defines and proposes various architecture smells and each one affects software quality in different aspects. Architecture refactoring is a process that removes an architecture smell and improves quality of system without any changes in system behavior. However, removing one of the architecture smells may cause creating of another smell in the system which directly affects software quality.

This thesis proposes a refactoring approach for Too Large Packages smell which is one of the architecture smells by using community detection in extract-package refactoring process. This process is used for extracting the package which is identified as Too Large Packages smell into small ones and avoids additional impacts from newly appeared dependency-based architecture smells by focusing on Unstable Dependency smell.

The proposed approach provides feasible solution to resolve impact from Too Large Packages smell which can be evaluated by comparing number of detected architecture smells and software metrics of before and after applying refactoring process. In addition, this thesis also develops a software tool based on the proposed approach to suggest the refactoring solution and support developers to improve software quality by resolving impacts from Too Large Packages smell during implementation and maintenance phase.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study:	Computer Science	Student's Signature .....
Academic Year:	2019	Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีจากคำแนะนำ ความช่วยเหลือ และความเมตตาจากรองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่สละเวลาให้คำปรึกษา ให้ความรู้และให้คำแนะนำต่าง ๆ อันเป็นประโยชน์ในการทำวิจัย ตลอดช่วงระยะเวลาการทำวิจัยนี้

กราบขอบพระคุณ รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒ ประธานกรรมการสอบ ศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล และผู้ช่วยศาสตราจารย์ ดร.บัณฑิต ฐานะโสภณ คณะกรรมการสอบวิทยานิพนธ์ ที่ได้สละเวลาในการตรวจสอบความถูกต้องสมบูรณ์ของวิทยานิพนธ์ และให้คำแนะนำต่าง ๆ ในการสอบวิทยานิพนธ์

ขอขอบคุณคณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ความรู้ซึ่งมีส่วนสำคัญที่สามารถนำมาใช้ประโยชน์ในการทำวิจัยนี้

ขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ทุกคนที่ให้การสนับสนุน ให้กำลังใจ และความช่วยเหลือตลอดระยะเวลาที่ศึกษาและทำงานวิจัยนี้

ท้ายที่สุด ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และคนในครอบครัวที่คอยเป็นกำลังใจและสนับสนุนกันมาตลอดจนงานวิจัยนี้ได้สำเร็จลุล่วงไปได้ด้วยดี

บุณยไชย จันทร์เทียน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค

บทคัดย่อภาษาอังกฤษ ..... ง

กิตติกรรมประกาศ ..... จ

สารบัญ ..... ฉ

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฐ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตการดำเนินงาน 2

1.4 ขั้นตอนการดำเนินงาน 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 บทความวิจัยได้รับการตีพิมพ์ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

2.1.1 โค้ดสเมล 5

2.1.2 อาคิเทกเจอร์สเมล 5

2.1.3 แพ็กเกจขนาดใหญ่เกิน 6

2.1.4 อันสเตเบิลดีเพนเดนซี 6

2.1.5 ตัวชี้วัดซอฟต์แวร์ 7

2.1.6 หลักการการพึ่งพาที่มีเสถียรภาพ 9

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.7 การรีแฟคทอริงเชิงสถาปัตยกรรม 9
2.1.8 การตรวจหาคอมมิวนิตี 10
2.1.9 อัลกอริทึมลูแวน 11
2.1.10 การประเมินประสิทธิภาพ 12
2.2 งานวิจัยที่เกี่ยวข้อง 13
2.2.1 งานวิจัยที่เกี่ยวข้องกับการตรวจหาอาคิเทกเจอร์สเมล 13
2.2.2 งานวิจัยที่เกี่ยวข้องกับกระบวนการรีแฟคทอริงเชิงสถาปัตยกรรม 14
บทที่ 3 วิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี 18
3.1. กำหนดโปรเจคโอเพนซอร์สที่นำมาใช้ในงานวิจัย 18
3.2. การวิเคราะห์โปรเจคโอเพนซอร์ส 20
3.3. การเตรียมโปรเจคโอเพนซอร์สให้อยู่ในรูปแบบโครงสร้างและความสัมพันธ์ของข้อมูล 21
3.3.1. การเตรียมโครงสร้างข้อมูลของโปรเจค 22
3.3.2. การเตรียมความสัมพันธ์ของข้อมูลระหว่างคลาส 23
3.4. การตรวจหาอาคิเทกเจอร์สเมล 24
3.4.1. การตรวจหาแพ็กเกจขนาดใหญ่เกิน 24
3.4.2. การตรวจหาอันสเตเบิลดีเพนเดนซี 25
3.4.3. การเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมล 26
3.5. การตรวจหาคอมมิวนิตีภายในเนมสเปซ 28
3.5.1. การแปลงข้อมูลให้อยู่ในรูปแบบกราฟมีทิศทางและมีน้ำหนัก 28
3.5.2. การตรวจหาคอมมิวนิตีด้วยอัลกอริทึมลูแวน 30
3.6. การประเมินผลกระทบของอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพา 30
3.6.1. การจัดเตรียมกลุ่มของคอมมิวนิตี 31
3.6.2. ทดสอบการแยกเนมสเปซย่อยจากกลุ่มของคอมมิวนิตี 34
3.7. การพัฒนาเครื่องมือส่วนแสดงผล 37

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.8. การทดสอบและประเมินผลเครื่องมือ 37
บทที่ 4 การออกแบบและพัฒนาเครื่องมือ 39
4.1. สภาพแวดล้อมของการพัฒนาเครื่องมือ 39
4.2. ความต้องการของการพัฒนาเครื่องมือเบื้องต้น 40
4.3. แผนภาพยูสเคส 40
4.3.1. ส่วนการนำเข้าข้อมูล 41
4.3.2. ส่วนการแสดงข้อมูลก่อนกระบวนการรีแฟคทอริง 41
4.3.3. ส่วนการวิเคราะห์หาวิธีการรีแฟคทอริง 43
4.3.4. ส่วนการออกรายงานผล 43
4.4. แผนภาพกิจกรรม 43
4.5. แผนภาพคลาส 44
4.5.1. แผนภาพคลาสของการจัดเก็บข้อมูลโครงสร้างโปรเจค 45
4.5.2. แผนภาพคลาสของการตรวจหาอาคิเทกเจอร์สเมล 46
4.5.3. แผนภาพคลาสของวิธีการรีแฟคทอริงโดยใช้การตรวจหาคอมมิวนิตี 47
บทที่ 5 การทดสอบและประเมินผลความสามารถของเครื่องมือ 49
5.1. ประเมินผลโปรเจคที่ใช้ในงานวิจัย 50
5.1.1. รายละเอียดโปรเจคโปรเจคที่ใช้ในงานวิจัย 50
5.1.2. การตรวจหาอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์ 51
5.2. ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนา 52
5.3. ประเมินผลวิธีการรีแฟคทอริงของเครื่องมือที่พัฒนา 58
5.4. ประเมินผลการรีแฟคทอริงโปรเจคตามที่เครื่องมือแนะนำ 62
บทที่ 6 บทสรุปและข้อเสนอแนะ 64
6.1. สรุปผลการวิจัย 64
6.2. ข้อจำกัดของงานวิจัย 65

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

6.3. แนวทางการพัฒนางานวิจัยต่อ 66
บรรณานุกรม 67
ภาคผนวก ก คู่มือการใช้งานเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกิน โดยใช้การตรวจหาคอมมิวนิตี 69
ก.1. การนำข้อมูลโปรเจคการทำงานหลักเข้าระบบ 69
ก.2. การแสดงผลวิเคราะห์โปรเจคขั้นพื้นฐานและการตรวจหาอาคิเทกเจอร์สเมล 71
ก.3. การตรวจหาและแสดงผลวิธีการรีแฟคทอริง 75
ก.4. การออกรายงานวิธีการรีแฟคทอริง 79
ภาคผนวก ข การทดสอบและประเมินผลความสามารถของเครื่องมือเพิ่มเติม 82
ข.1. ผลการตรวจหาอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์เพิ่มเติม 82
ข.2. ผลการประเมินผลการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนาเพิ่มเติม 85
ข.2.1. ประเมินผลการเตรียมโครงสร้างข้อมูลโปรเจค 85
ข.2.2. ประเมินผลการเตรียมความสัมพันธ์ของข้อมูลระหว่างคลาส 88
ข.2.3. ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน 91
ข.2.4. ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี 91
ข.2.5. ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน 93
ข.2.6. ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี 97
ข.3. ผลการประเมินวิธีการรีแฟคทอริงของเครื่องมือที่พัฒนาเพิ่มเติม 100
ข.3.1. ประเมินผลขั้นตอนการตรวจหาคอมมิวนิตีภายในเนมสเปซ 100
ข.3.2. ประเมินผลขั้นตอนการแยกเนมสเปซย่อย 106
ประวัติผู้เขียน 115

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 2.2.1 ตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้องกับงานวิจัย 8

ตารางที่ 3.3.1 ข้อกำหนดเพิ่มเติมสำหรับโปรเจคที่ใช้ในการทดลอง 20

ตารางที่ 3.2 รายละเอียดโครงสร้างข้อมูลที่มีลำดับชั้น 22

ตารางที่ 4.1 รายละเอียดของสภาพแวดล้อมของการพัฒนาเครื่องมือด้านซอฟต์แวร์ 39

ตารางที่ 5.1 รายละเอียดโปรเจค Hyperion 50

ตารางที่ 5.2 รายละเอียดโปรเจค Hangfire.Core 51

ตารางที่ 5.3 รายละเอียดโปรเจค Nancy 51

ตารางที่ 5.4 ผลการวิเคราะห์และตรวจหาอาคิเทกเจอร์สเมลในโปรเจคด้วยเครื่องมือพาณิชย์ 52

ตารางที่ 5.5 ผลการวิเคราะห์และตรวจหาอาคิเทกเจอร์สเมลในโปรเจคด้วยเครื่องมือที่พัฒนา 53

ตารางที่ 5.6 ผลการเปรียบเทียบจำนวนเนมสเปซและจำนวนคลาสที่ตรวจพบ 56

ตารางที่ 5.7 ผลการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบ 57

ตารางที่ 5.8 แสดงผลการประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมล รูปแบบแพ็กเกจขนาดใหญ่เกิน 58

ตารางที่ 5.9 แสดงผลการประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมล รูปแบบอันสเตเบิลดีเพนเดนซี 58

ตารางที่ 5.10 แสดงการเปรียบเทียบค่าตัวชี้วัดพื้นฐานก่อนและหลังกระบวนการรีแฟคทอริง 59

ตารางที่ 5.11 แสดงการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลและค่าตัวชี้วัดก่อนและหลังกระบวนการรีแฟคทอริง 61

ตารางที่ 5.12 ผลการเปรียบเทียบการตรวจหาจำนวนอาคิเทกเจอร์สเมลหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริง 62

ตารางที่ 5.13 แสดงผลการประเมินประสิทธิภาพเครื่องมือหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริงสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน 63

ตารางที่ 5.14 แสดงผลการประเมินประสิทธิภาพเครื่องมือหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริงสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี..... 63

ตารางที่ ข.1 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินด้วยเครื่องมือพาณิชย์ 82

ตารางที่ ข.2 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีด้วยเครื่องมือพาณิชย์ 82

ตารางที่ ข.3 ผลการวิเคราะห์ข้อมูลระดับเนมสเปซของโปรเจค Hyperion จากเครื่องมือที่พัฒนา.. 85

ตารางที่ ข.4 ผลการวิเคราะห์ข้อมูลระดับเนมสเปซของโปรเจค Hangfire.Core จากเครื่องมือที่พัฒนา 86

ตารางที่ ข.5 ผลการวิเคราะห์ข้อมูลระดับเนมสเปซของโปรเจค Nancy จากเครื่องมือที่พัฒนา 87

ตารางที่ ข.6 เมทริกซ์การพึ่งพาของเนมสเปซในโปรเจค Hyperion จากเครื่องมือที่พัฒนา..... 88

ตารางที่ ข.7 เมทริกซ์การพึ่งพาของเนมสเปซในโปรเจค Hangfire.Core จากเครื่องมือที่พัฒนา 89

ตารางที่ ข.8 เมทริกซ์การพึ่งพาของเนมสเปซในโปรเจค Nancy จากเครื่องมือที่พัฒนา 90

ตารางที่ ข.9 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินด้วยเครื่องมือที่พัฒนา 91

ตารางที่ ข.10 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีด้วยเครื่องมือที่พัฒนา 91

ตารางที่ ข.11 ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน 94

ตารางที่ ข.12 ผลการเปรียบเทียบจำนวนแพ็กเกจขนาดใหญ่เกินรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hyperion 96

ตารางที่ ข.13 ผลการเปรียบเทียบจำนวนแพ็กเกจขนาดใหญ่เกินรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hangfire.Core 96

ตารางที่ ข.14 ผลการเปรียบเทียบจำนวนแพ็กเกจขนาดใหญ่เกินรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Nancy 96

ตารางที่ ข.15 ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี..... 97

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.16 ผลการเปรียบเทียบจำนวนอันสเตเบิลดีเพนเดนซีรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hyperion 99

ตารางที่ ข.17 ผลการเปรียบเทียบจำนวนอันสเตเบิลดีเพนเดนซีรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hangfire.Core 99

ตารางที่ ข.18 ผลการเปรียบเทียบจำนวนอันสเตเบิลดีเพนเดนซีรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Nancy 100

ตารางที่ ข.19 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hyperion.Internal 100

ตารางที่ ข.20 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hyperion.ValueSerializers 101

ตารางที่ ข.21 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hangfire 102

ตารางที่ ข.22 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hangfire.Dashboard 103

ตารางที่ ข.23 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hangfire.Server 104

ตารางที่ ข.24 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Nancy 105

ตารางที่ ข.25 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Nancy.Diagnostics 106

ตารางที่ ข.26 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hyperion.Internal 107

ตารางที่ ข.27 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hyperion.ValueSerializers 108

ตารางที่ ข.28 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hangfire 109

ตารางที่ ข.29 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hangfire.Server 111

ตารางที่ ข.30 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Nancy 113

ตารางที่ ข.31 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Nancy.Diagnostics 114

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญภาพ

หน้า

รูปที่ 2.1 ตัวอย่างการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี 7

รูปที่ 2.2 ตัวอย่างการจัดกลุ่มบนพื้นฐานของความหนาแน่น 10

รูปที่ 2.3 ตัวอย่างการจัดกลุ่มบนพื้นฐานของรูปแบบ..... 11

รูปที่ 2.4 ตัวอย่างการตรวจหาคอมมิวนิตีในเครือข่ายด้วยอัลกอริทึมลูแวน 12

รูปที่ 2.5 แผนภาพและสมการการคำนวณค่าอินเตอร์เซกชันโอเวอร์ยูเนียน 13

รูปที่ 3.1 ภาพรวมการดำเนินงาน 8 ขั้นตอนหลัก..... 19

รูปที่ 3.2 เครื่องมือดีซิกไนต์สำหรับการตรวจหาอาคิเทกเจอร์สเมล [18]..... 21

รูปที่ 3.3 แผนภาพโครงสร้างของข้อมูลที่มีลำดับชั้น..... 22

รูปที่ 3.4 การพิจารณาความสัมพันธ์ของข้อมูลระหว่างคลาส 23

รูปที่ 3.5 การพิจารณาความสัมพันธ์ของข้อมูลระหว่างเนมสเปซ..... 24

รูปที่ 3.6 ตัวอย่างการตรวจหาอันสเตเบิลดีเพนเดนซี 25

รูปที่ 3.7 ภาพรวมการทำงานขั้นตอนการตรวจหาคอมมิวนิตีภายในเนมสเปซ..... 27

รูปที่ 3.8 ภาพรวมการทำงานขั้นตอนการประเมินผลกระทบของอาคิเทกเจอร์สเมล บนพื้นฐานการพึ่งพา 27

รูปที่ 3.9 กราฟความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเอ และระหว่างคลาสภายในเนมสเปซเอกับคลาสในเนมสเปซรอบข้าง 29

รูปที่ 3.10 กราฟความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเอ และระหว่างคลาสภายในเนมสเปซเอกับเนมสเปซรอบข้าง 29

รูปที่ 3.11 ตัวอย่างผลลัพธ์การตรวจหาคอมมิวนิตีจากข้อมูลกราฟแบบมีทิศทางและมีน้ำหนัก 30

รูปที่ 3.12 ตัวอย่างการนำเนมสเปซรอบข้างออกจากคอมมิวนิตี 31

รูปที่ 3.13 ขั้นตอนการจัดเตรียมกลุ่มคอมมิวนิตีที่เป็นไปได้สำหรับการแยกเนมสเปซ..... 33

รูปที่ 3.14 ตัวอย่างการแยกเนมสเปซย่อยจากกลุ่มคอมมิวนิตี..... 35

Chulalongkorn University

รูปที่ 3.15 ตัวอย่างการเปรียบเทียบโครงสร้างความสัมพันธ์ก่อนและหลังการแยกเนมสเปซ 35

รูปที่ 3.16 ตัวอย่างการพิจารณาวิธีการรีแฟคทอริงที่เป็นไปได้ตามเงื่อนไข 36

รูปที่ 3.17 ตัวอย่างการเปรียบเทียบวิธีการรีแฟคทอริงที่เป็นไปได้เพื่อหาวิธีที่ดีที่สุด 37

รูปที่ 4.1 แผนภาพยูสเคสของเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกิน โดยใช้การตรวจหาคอมมิวนิตี 41

รูปที่ 4.2 แผนภาพกิจกรรมของเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกิน โดยใช้การตรวจหาคอมมิวนิตี 44

รูปที่ 4.3 แผนภาพคลาสแสดงภาพรวมการทำงานของเครื่องมือ 45

รูปที่ 4.4 แผนภาพคลาสของการจัดเก็บข้อมูลโครงสร้างโปรเจค 45

รูปที่ 4.5 แผนภาพคลาสของการตรวจหาอาคิเทกเจอร์สเมล 46

รูปที่ 4.6 แผนภาพคลาสของวิธีการรีแฟคทอริงโดยใช้การตรวจหาคอมมิวนิตี 48

รูปที่ 5.1 ภาพรวมขั้นตอนการประเมินผล 49

รูปที่ 5.2 กราฟการพึ่งพาของเนมสเปซในโปรเจค Hyperion จากเครื่องมือที่พัฒนา 53

รูปที่ 5.3 กราฟการพึ่งพาของเนมสเปซในโปรเจค Hangfire.Core จากเครื่องมือที่พัฒนา..... 54

รูปที่ 5.4 กราฟการพึ่งพาของเนมสเปซในโปรเจค Nancy จากเครื่องมือที่พัฒนา..... 55

รูปที่ ก.1 หน้าจอเริ่มต้นสำหรับผู้ใช้งาน..... 69

รูปที่ ก.2 เมนูสำหรับนำเข้าข้อมูลโปรเจคใหม่หรือเลือกโปรเจคล่าสุด 70

รูปที่ ก.3 เมนูเส้นทางลัดในหน้าจอเริ่มต้นสำหรับนำเข้าข้อมูลโปรเจคใหม่หรือเลือกโปรเจคล่าสุด .. 70

รูปที่ ก.4 หน้าต่างสำหรับผู้ใช้งานเลือกนำเข้าข้อมูลโปรเจคใหม่..... 70

รูปที่ ก.5 เครื่องมือแสดงสถานะการทำงานในแถบสถานะระหว่างการประมวลผล..... 71

รูปที่ ก.6 ข้อมูลโปรเจคในระดับเนมสเปซและตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง..... 72

รูปที่ ก.7 ข้อมูลโปรเจคในระดับคลาสและตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง 72

รูปที่ ก.8 ผลการตรวจหาอาคิเทกเจอร์สเมลในโปรเจค..... 73

รูปที่ ก.9 กราฟการพึ่งพาระหว่างเนมสเปซภายในโปรเจค 74

รูปที่ ก.10 กราฟการพึ่งพาระหว่างคลาสภายในโปรเจครวมถึงความสัมพันธ์กับเนมสเปซข้างเคียง . 74
รูปที่ ก.11 เมทริกซ์การพึ่งพาของเนมสเปซภายในโปรเจค 75
รูปที่ ก.12 แสดงความหมายของความสัมพันธ์ระหว่างเนมสเปซในเมทริกซ์การพึ่งพา 75
รูปที่ ก.13 แสดงปุ่มสำหรับการเริ่มตรวจหาวิธีการรีแฟคทอริง 75
รูปที่ ก.14 แสดงแถบผลวิเคราะห์การรีแฟคทอริง 76
รูปที่ ก.15 แถบผลการตรวจหาคอมมิวนิตีโดยแสดงความสัมพันธ์ของคอมมิวนิตี 77
รูปที่ ก.16 แถบผลการตรวจหาคอมมิวนิตีโดยแสดงความสัมพันธ์ของคลาส 77
รูปที่ ก.17 แถบแสดงวิธีการรีแฟคทอริงโดยแสดงความสัมพันธ์ของคอมมิวนิตีในเนมสเปซใหม่ 78
รูปที่ ก.18 แถบแสดงวิธีการรีแฟคทอริงโดยแสดงความสัมพันธ์ของเนมสเปซที่ถูกแยกออกมาใหม่ . 78
รูปที่ ก.19 แสดงผลการตรวจหาอาคิเทกเจอร์สเมลและค่าตัวชี้วัดหลังจากการแยกเนมสเปซ 78
รูปที่ ก.20 การออกรายงานวิธีการรีแฟคทอริง 79
รูปที่ ก.21 รายงานส่วนข้อมูลโปรเจค 79
รูปที่ ก.22 รายงานส่วนผลการตรวจหาอาคิเทกเจอร์สเมล 80
รูปที่ ก.23 รายงานส่วนแสดงวิธีการรีแฟคทอริงของเนมสเปซ 81
รูปที่ ข.1 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hyperion.Internal 101
รูปที่ ข.2 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hyperion.ValueSerializers 101
รูปที่ ข.3 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hangfire 102
รูปที่ ข.4 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hangfire.Dashboard 103
รูปที่ ข.5 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hangfire.Server 104
รูปที่ ข.6 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Nancy 105
รูปที่ ข.7 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Nancy.Diagnostics 106
รูปที่ ข.8 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hyperion.Internal กับเนมสเปซรอบข้าง 107
รูปที่ ข.9 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hyperion.Internal กับเนมสเปซรอบข้าง 107

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ ข.10 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hyperion.ValueSerializers กับเนมสเปซรอบข้าง 108

รูปที่ ข.11 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hyperion.ValueSerializers กับเนมสเปซรอบข้าง 109

รูปที่ ข.12 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hangfire กับเนมสเปซรอบข้าง 110

รูปที่ ข.13 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hangfire กับเนมสเปซรอบข้าง 110

รูปที่ ข.14 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hangfire.Dashboard กับเนมสเปซรอบข้าง 111

รูปที่ ข.15 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hangfire.Server กับเนมสเปซรอบข้าง 112

รูปที่ ข.16 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hangfire.Server กับเนมสเปซรอบข้าง 112

รูปที่ ข.17 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Nancy กับเนมสเปซรอบข้าง 113

รูปที่ ข.18 กราฟการพึ่งพาของเนมสเปซใหม่ของ Nancy กับเนมสเปซรอบข้าง 113

รูปที่ ข.19 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Nancy.Diagnostics กับเนมสเปซรอบข้าง 114

รูปที่ ข.20 กราฟการพึ่งพาของเนมสเปซใหม่ของ Nancy.Diagnostics กับเนมสเปซรอบข้าง 114

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

คุณภาพซอฟต์แวร์ (software quality) จัดเป็นสิ่งสำคัญในการพัฒนาระบบซอฟต์แวร์ เนื่องจากระบบที่มีคุณภาพซอฟต์แวร์ที่ต่ำมีแนวโน้มที่จะก่อให้เกิดปัญหาในด้านต่าง ๆ ตามมาในภายหลัง การตรวจโค้ดสเมล (code smell) [1] ถูกนำมาใช้โดยทั่วไปสำหรับการประเมินคุณภาพซอฟต์แวร์ โดยพิจารณาคุณภาพซอฟต์แวร์จากโครงสร้างในระดับโค้ดเพื่อบ่งชี้ถึงโครงสร้างที่ควรจะถูกรีแฟคทอริง แต่ในระดับโครงสร้างสถาปัตยกรรม อาคิเทกเจอร์สเมล (architecture smell) [2][3] ยังถูกนำมาพิจารณาคุณภาพซอฟต์แวร์ไม่มากนัก ถึงแม้ว่าจะส่งผลกระทบด้านลบต่อคุณภาพของระบบในระดับที่รุนแรงกว่าผลกระทบของโค้ดสเมล ในแง่ของผลกระทบต่อการบำรุงรักษาปัญหาที่เกิดขึ้น [4]

การปรับปรุงคุณภาพซอฟต์แวร์จากผลกระทบที่เกิดขึ้นจากอาคิเทกเจอร์สเมลสามารถทำได้โดยกระบวนการรีแฟคทอริงเชิงสถาปัตยกรรม (architecture refactoring) อย่างไรก็ตามอาคิเทกเจอร์สเมลนั้นสามารถถูกพบได้ในหลายระดับ [2] เช่น ระหว่างแพ็กเกจ (package), ระหว่างระบบย่อย (subsystem), ระหว่างเลเยอร์ (layer) เป็นต้น ดังนั้นกระบวนการรีแฟคทอริงสำหรับอาคิเทกเจอร์สเมลนั้นขึ้นกับรูปแบบและระดับที่ตรวจพบ โดยอาคิเทกเจอร์สเมลรูปแบบหนึ่งที่สามารถตรวจพบได้โดยทั่วไป คือ แพ็กเกจขนาดใหญ่เกิน (too large package) หรือในบางเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลเรียกว่า ก็อดคอมโพเนนต์ (god component) ซึ่งเป็นโครงสร้างในระดับแพ็กเกจขนาดใหญ่เกิน ทำให้เป็นเรื่องยากในการทำความเข้าใจและส่งผลต่อการบำรุงรักษา โดยการปรับปรุงอาคิเทกเจอร์สเมลรูปแบบนี้สามารถทำได้โดยการแบ่งแพ็กเกจที่มีขนาดใหญ่ให้มีขนาดเล็กลงเป็นแพ็กเกจย่อยหลาย ๆ แพ็กเกจ โดยเรียกกระบวนการนี้ว่า การรีแฟคทอริงด้วยการแยกแพ็กเกจ (extract package refactoring) แต่อย่างไรก็ตามยังไม่มีงานวิจัยไหนที่นำเสนอถึงผลกระทบที่เกิดขึ้นหลังจากกระบวนการรีแฟคทอริงนี้ โดยอาจจะก่อให้เกิดอาคิเทกเจอร์สเมลรูปแบบอื่น ๆ เกิดมากขึ้น ตัวอย่างเช่น อันสเตเบิลดีเพนเดนซี (unstable dependency) เป็นรูปแบบของอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาที่เกิดจากแพ็กเกจในระบบพึ่งพาแพ็กเกจอื่น ๆ ที่มีความเสถียรภาพน้อยกว่า กล่าวคือการแยกแพ็กเกจใหม่ขึ้นมาในระบบก่อให้เกิดการเปลี่ยนแปลงด้านโครงสร้างและส่งผลกระทบต่อการพึ่งพาระหว่างแพ็กเกจต่าง ๆ ในระบบ ซึ่งอาจจะก่อให้เกิดอาคิเทกเจอร์สเมลในรูปแบบนี้เพิ่มมากขึ้นหลังกระบวนการรีแฟคทอริง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ดังนั้นในงานวิจัยนี้จะศึกษาและนำเสนอวิธีการรีแฟคทอริงด้วยการแยกแพ็กเกจ ในโครงสร้างแพ็กเกจขนาดใหญ่เกินด้วยวิธีการตรวจหาคอมมิวนิตี จากนั้นจะพิจารณาถึงผลกระทบของอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาในรูปแบบอันสเตเบิลดีเพนเดนซี ซึ่งเป็นรูปแบบที่อาจจะเกิดขึ้นหลังจากกระบวนการรีแฟคทอริงได้ ซึ่งจากงานวิจัยที่ผ่านมาที่นำเสนอวิธีการรีแฟคทอริงในระดับแพ็กเกจด้วยวิธีการแยกแพ็กเกจนั้น ยังไม่มีการนำเสนอถึงผลกระทบของอาคิเทกเจอร์สเมลรูปแบบอื่น ๆ ที่เกิดขึ้นตามมาหลังจากกระบวนการรีแฟคทอริง ดังนั้นในงานวิจัยนี้จึงเลือกที่จะนำเสนอกระบวนการรีแฟคทอริงด้วยการแยกแพ็กเกจที่พิจารณาถึงผลกระทบของอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีเพื่อไม่ให้มีการเกิดอาคิเทกเจอร์สเมลรูปแบบนี้เพิ่มมากขึ้น ซึ่งช่วยให้การปรับคุณภาพซอฟต์แวร์มีประสิทธิภาพที่ดีขึ้น

1.2 วัตถุประสงค์

1. เพื่อออกแบบวิธีการรีแฟคทอริงด้วยการแยกแพ็กเกจย่อยสำหรับแพ็กเกจขนาดใหญ่เกิน หรืออาคิเทกเจอร์สเมลรูปแบบก็อดคอมโพเนนต์ด้วยอัลกอริทึมการตรวจหาคอมมิวนิตีโดยประเมินผลกระทบของอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาที่เกิดขึ้นหลังกระบวนการรีแฟคทอริง
2. เพื่อออกแบบและพัฒนาเครื่องมือที่ช่วยสนับสนุนวิธีการรีแฟคทอริงด้วยการแยกแพ็กเกจย่อยด้วยการตรวจหาคอมมิวนิตี

1.3 ขอบเขตการดำเนินงาน

1. งานวิจัยนี้นำเสนอวิธีการรีแฟคทอริงด้วยการตรวจหาคอมมิวนิตีสำหรับอาติเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน
2. โปรเจคโอเพนซอร์สที่นำมาใช้ในการทดลองจะต้องพัฒนาด้วยภาษาซีชาร์ปบนพื้นฐานของดอตเน็ตเฟรมเวิร์กเวอร์ชัน 4.5 หรือสูงกว่า และสามารถคอมไพล์ (compile) ได้อย่างสมบูรณ์ผ่านโปรแกรมไมโครซอฟวิชวลสตูดิโอเวอร์ชัน 2017
3. งานวิจัยนี้จะวิเคราะห์โครงสร้างความสัมพันธ์ของโปรเจคโอเพนซอร์สเฉพาะในส่วนโปรเจคการทำงานหลักเพียงโปรเจคเดียวเท่านั้น โดยไม่พิจารณาความสัมพันธ์ระหว่างโปรเจคต่าง ๆ
4. การประเมินผลกระทบที่เกิดจากการแยกแพ็กเกจจะพิจารณาจากอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาในรูปแบบอันสเตเบิลดีเพนเดนซีเท่านั้น
5. การตรวจสอบความถูกต้องของวิธีการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนาจะเปรียบเทียบปริมาณอาคิเทกเจอร์สเมลที่ตรวจพบกับเครื่องมือเชิงพาณิชย์ชื่อ ดีซิกไนต์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

6. ผลลัพธ์ของเครื่องมือจะนำเสนอกลุ่มของคลาสที่ถูกแนะนำให้แยกไปยังเนมสเปซหรือแพ็กเกจใหม่ และปริมาณอาคิเทกเจอร์สเมลที่ตรวจพบหลังจากการปรับปรุงโครงสร้าง กรณีที่สามารถตรวจพบกลุ่มคอมมิวนิตีที่สามารถแยกออกมาได้โดยไม่ก่อให้เกิดอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีเพิ่มมากขึ้นเท่านั้น
7. เครื่องมือที่พัฒนาจะทดสอบการทำงานบนระบบปฏิบัติการวินโดวส์
8. เครื่องมือที่พัฒนาจะนำเสนอผลลัพธ์เพื่อช่วยสนับสนุนกระบวนการรีแฟคทอริง โดยไม่รวมถึงการแก้ไขซอร์สโค้ดของโปรเจคโอเพนซฮร์สที่นำมาทดลองแบบอัตโนมัติ

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษางานวิจัยที่เกี่ยวข้องกับการตรวจหาอาคิเทกเจอร์สเมล เพื่อศึกษารูปแบบและผลกระทบของอาคิเทกเจอร์สเมล รวมถึงขั้นตอนและวิธีการตรวจหา
2. ศึกษางานวิจัยที่เกี่ยวข้องกับการรีแฟคทอริงเชิงสถาปัตยกรรม เพื่อศึกษากระบวนการรีแฟคทอริงรูปแบบต่าง ๆ
3. กำหนดโอเพนซอร์สโปรเจคที่จะใช้ในการทดลองในงานวิจัย
4. กำหนดรูปแบบอาคิเทกเจอร์สเมลที่จะนำมาพิจารณาในงานวิจัย
5. วิเคราห์โครงสร้างความสัมพันธ์ในระดับคลาสและในระดับเนมสเปซที่ส่งผลต่อการเกิดอาคิเทกเจอร์สเมล
6. ออกแบบวิธีการทดลอง และพัฒนาเครื่องมือสำหรับการทดลอง
7. ทำการทดลองโดยใช้เครื่องมือ และเก็บรวบรวมข้อมูลที่ได้จากการทดลอง
8. วิเคราะห์ข้อมูล และสรุปผลการทดลอง เพื่อประเมินผลเครื่องมือ
9. จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำวิธีการรีแฟคทอริงด้วยการแยกแพ็กเกจที่นำเสนอไปใช้ในกระบวนการรีแฟคทอริงเพื่อปรับปรุงโครงสร้างสถาปัตยกรรมของระบบซอฟต์แวร์ โดยไม่ก่อให้เกิดผลกระทบจากอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาเพิ่มมากขึ้น และหลังจากแยกแพ็กเกจระบบซอฟต์แวร์มีคุณภาพซอฟต์แวร์ที่ดีขึ้น

1.6 บทความวิจัยได้รับการตีพิมพ์

งานวิจัยนี้ได้รับการคัดเลือกและตีพิมพ์เป็นบทความวิชาการเรื่อง A Refactoring Approach for Too Large Packages using Community Detection and Dependency-based Impacts โดยนายบุณยไชย จันทร์เทียน และรองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี ในการประชุมวิชาการ “The World Symposium on Software Engineering (WSSE 2019)” ระหว่างวันที่ 20-23 กันยายน 2562 ณ เมืองอู่ฮั่น ประเทศจีน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

งานวิจัยนี้จะศึกษาและนำเสนอวิธีการรีแฟคทอริงสำหรับอาคิเทกเจอร์สเมลในระบบซอฟต์แวร์ที่มีโครงสร้างแพ็กเกจที่มีขนาดใหญ่เกินไป โดยวิธีการรีแฟคทอริงด้วยการแยกแพ็กเกจ โดยนำวิธีการตรวจหาคอมมิวนิตีมาประยุกต์ใช้ในการปรับปรุงโครงสร้างของระบบซอฟต์แวร์ โดยทฤษฎีที่เกี่ยวข้องมีรายละเอียดดังนี้

2.1.1 โค้ดสเมล

โค้ดสเมล (Code smell) [1] หมายถึง โครงสร้างที่ชี้แน่ชัดในระดับโค้ดและถูกแนะนำว่ามีความเป็นไปได้ที่จะต้องรีแฟคทอริง (refactoring) โดยในบางงานวิจัยอาจมีการนิยามชื่อเรียกที่แตกต่างกัน เช่น ร่องรอยที่ผิดพลาด (bad smell) หรือซอฟต์แวร์สเมล (software smell) โดยโค้ดสเมลนั้นจัดเป็นตัวชี้วัดคุณภาพซอฟต์แวร์ที่สำคัญตัวหนึ่ง ซึ่งเป็นที่นิยมอย่างแพร่หลาย และเมื่อพิจารณาจากขอบเขตของผลกระทบที่เกิดจากโค้ดสเมล จะสามารถแบ่งประเภทของโค้ดสเมลออกได้อีกหลายประเภท [5] ประกอบด้วย

- อาคิเทกเจอร์สเมล (architecture smell)
- ดีไซน์สเมล (design smell)
- อิมพลีเม้นเทชันสเมล (implementation smell)
- เอนเนอร์จีสเมล (energy smell)
- เพอร์ฟอร์แมนซ์สเมล (performance smell)
- เทสสเมล (test smell)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.2 อาคิเทกเจอร์สเมล

อาคิเทกเจอร์สเมล (Architecture smell) [6] ถูกใช้โดยทั่วไปในการตัดสินใจเกี่ยวกับผลกระทบด้านลบที่มีต่อคุณภาพของระบบซอฟต์แวร์ โดยชี้ให้เห็นถึงจุดที่ควรจะถูกปรับปรุงในเชิงของโครงสร้างสถาปัตยกรรม และสามารถเกิดขึ้นได้ในหลายระดับ เช่น ระหว่างแพ็กเกจ, ระหว่างระบบย่อย หรือระหว่างเลเยอร์ เป็นต้น โดยในบางงานวิจัยอาจมีการนิยามชื่อเรียกที่แตกต่างกัน เช่น อาคิเทคเจอรัลสเมล (architectural smell)

จากงานวิจัยที่ผ่านมาได้นำเสนอรูปแบบและผลกระทบของอาคิเทกเจอร์สเมลในหลายรูปแบบ รวมถึงวิธีการตรวจหาอาคิเทกเจอร์สเมลในหลายวิธี ตัวอย่างเช่น งานวิจัยที่นำเสนอแผนภาพ

โครงสร้างที่บ่งชี้การเกิดอาคิเทกเจอร์สเมล [6] โดยอาศัยสถาปนิกระบบ (system architect) เป็นผู้ประเมินโครงสร้างของระบบเปรียบเทียบกับแผนภาพ, งานวิจัยที่นำเสนอเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลในรูปแบบต่าง ๆ ขึ้น [4, 7] เพื่อช่วยลดภาระของสถาปนิกระบบในการทำความเข้าใจโครงสร้างและโค้ดของระบบซอฟต์แวร์ที่มีจำนวนมาก โดยอาศัยตัวชี้วัดรวมถึงกฎต่าง ๆ ในการบ่งชี้ว่ามีการเกิดขึ้นของอาคิเทกเจอร์สเมลแฝงอยู่ในระบบ ต่อมาได้มีงานวิจัยที่พยายามจัดกลุ่มของอาคิเทกเจอร์สเมลเพื่อแบ่งประเภทของอาคิเทกเจอร์สเมลรูปแบบต่าง ๆ [8] ประกอบด้วย

- กลุ่มของสเมลบนพื้นฐานความกังวล (concern-based smells)
- กลุ่มของสเมลบนพื้นฐานการพึ่งพา (dependency-based smells)
- กลุ่มของสเมลบนพื้นฐานอินเตอร์เฟซ (interface-based smells)
- กลุ่มของสเมลบนพื้นฐานการเข้าคู่ (coupling-based smells)

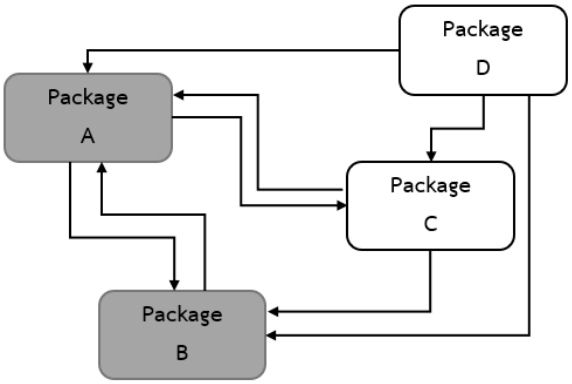
2.1.3 แพ็กเกจขนาดใหญ่เกิน

แพ็กเกจขนาดใหญ่เกิน (Too large package) [2] เป็นอาคิเทกเจอร์สเมลในระดับแพ็กเกจ โดยแพ็กเกจหมายถึงกลุ่มของคลาสตามแนวคิดในภาษาจาวา (Java) โดยคลาสในที่นี้จะรวมถึงอินเตอร์เฟซ (interface) หรือโครงสร้างอื่น ๆ ที่เทียบเท่าในภาษาอื่น ๆ โดยจะเรียกโดยรวมว่าคลาส โดยแพ็กเกจที่มีขนาดใหญ่หรือมีจำนวนคลาสภายในจำนวนมากย่อมส่งผลต่อการทำความเข้าใจและการบำรุงรักษา ซึ่งจัดเป็นอาคิเทกเจอร์สเมลรูปแบบหนึ่ง ดังนั้นจึงควรที่จะแยกคลาสภายในแพ็กเกจออกเป็นแพ็กเกจย่อย ๆ โดยค่าตัวชี้วัดที่บ่งชี้ถึงอาคิเทกเจอร์สเมลชนิดนี้ คือ แพ็กเกจที่มีจำนวนคลาสมากกว่า 30 คลาส หรือจำนวนบรรทัดของโค้ดที่มากกว่า 27,000 บรรทัด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.4 อันสเตเบิลดีเพนเดนซี

อันสเตเบิลดีเพนเดนซี (Unstable dependency) [4] เป็นอาคิเทกเจอร์สเมลที่เกิดขึ้นจากแพ็กเกจภายในระบบมีการพึ่งพาแพ็กเกจอื่น ๆ ที่มีความเสถียรภาพน้อยกว่า ซึ่งจัดเป็นข้อบกพร่องในระดับแพ็กเกจ วิธีการตรวจหาสามารถทำได้โดยใช้กราฟหาความสัมพันธ์ระหว่างแพ็กเกจและการคำนวณค่าตัวชี้วัดความไม่แน่นอนของทุกแพ็กเกจในระบบ เพื่อที่จะพิจารณาความเสถียรภาพของแพ็กเกจต่าง ๆ ที่มีการพึ่งพาอยู่ โดยตัวอย่างการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี แสดงดังรูปที่ 2.1



Package	Ce	Ca	I
Package A	2	3	0.40
Package B	1	3	0.25
Package C	2	2	0.50
Package D	3	0	1.00

รูปที่ 2.1 ตัวอย่างการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

จากตัวอย่างการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีในระบบซอฟต์แวร์ที่ประกอบด้วย 4 แพ็กเกจ โดยที่แพ็กเกจเอ (package A) มีค่าเอฟเฟอเรนท์คัปปลิง (efferent coupling หรือ Ce) เป็น 2 เนื่องจากแพ็กเกจเอพึ่งพาแพ็กเกจบี (package B) และแพ็กเกจซี (package C) และมีค่าแอฟเฟอเรนท์คัปปลิง (afferent coupling หรือ Ca) เป็น 3 เนื่องจากการที่แพ็กเกจบี, แพ็กเกจซี และแพ็กเกจดี มีการพึ่งพาแพ็กเกจเออยู่ โดยสามารถคำนวณค่าความไม่แน่นอน (instability หรือ I) เป็น 0.4 หลังจากคำนวณค่าความไม่แน่นอนของทุกแพ็กเกจในระบบเรียบร้อยแล้ว จะเป็นการตรวจหาอันสเตเบิลดีเพนเดนซีในแพ็กเกจต่าง ๆ โดยในระบบนี้แพ็กเกจเอถูกตรวจพบอันสเตเบิลดีเพนเดนซี เนื่องจากแพ็กเกจเอพึ่งพาแพ็กเกจซีที่มีค่าความไม่แน่นอนสูงกว่า หรือกล่าวคือแพ็กเกจซีมีความเสถียรภาพน้อยกว่าแพ็กเกจเอ และเช่นเดียวกับแพ็กเกจบีถูกตรวจพบอันสเตเบิลดีเพนเดนซี เนื่องจากแพ็กเกจบีพึ่งพาแพ็กเกจเอที่มีค่าความไม่แน่นอนสูงกว่า

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.5 ตัวชี้วัดซอฟต์แวร์

ตัวชี้วัดซอฟต์แวร์ (Software metrics) [9] เป็นมาตรฐานการวัดของระบบซอฟต์แวร์เพื่อที่จะบอกปริมาณของคุณลักษณะต่าง ๆ เช่น ความเชื่อมแน่น (cohesion), การคู่ควบ (coupling) หรือความเสถียรภาพ (stability) โดยตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้องกับงานวิจัยนี้ แสดงในตารางที่ 2.1

ตารางที่ 2.2.1 ตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้องกับงานวิจัย

ตัวชี้วัดซอต์แวร์	**ตัวย่อ**	**รายละเอียด**
ความเชื่อมแน่นเชิงสัมพันธ์ (Relational Cohesion)	H	ค่าตัวชี้วัดที่พิจารณาจากค่าเฉลี่ยของความสัมพันธ์ภายในต่อคลาสในแพ็กเกจ ดังสมการที่ 2.1 $$H = \frac{(R+1)}{N} \quad (2.1)$$ โดยที่ค่า R คือจำนวนความสัมพันธ์ของคลาสภายในแพ็กเกจ และค่า N คือจำนวนคลาสในแพ็กเกจ
เอฟเฟอเรนท์คัปปลิง (Efferent Coupling)	Ce	ค่าตัวชี้วัดที่วัดจากความสัมพันธ์ระหว่างคลาสในแพ็กเกจ โดยพิจารณาจากแพ็กเกจที่มีคลาสในระบบพึ่งพาคลาสภายนอกหรือจากอีกแพ็กเกจหนึ่ง เมื่อคลาสที่พึ่งพาในอีกแพ็กเกจเกิดการเปลี่ยนแปลงขึ้น ตัวแพ็กเกจนั้นย่อมมีความเสี่ยงต่อการแก้ไข โดยค่าตัวชี้วัดนี้มีค่าที่เหมาะสมอยู่ในช่วง 0 ถึง 20
แอฟเฟอเรนท์คัปปลิง (Afferent Coupling)	Ca	ค่าตัวชี้วัดจากความสัมพันธ์ระหว่างคลาสในแพ็กเกจ โดยพิจารณาตรงกันข้ามกับตัวชี้วัดเอฟเฟอเรนท์คัปปลิง กล่าวคือตัวชี้วัดนี้พิจารณาจากแพ็กเกจที่มีคลาสภายนอกจากแพ็กเกจอื่นพึ่งพาคลาสในแพ็กเกจนั้น ค่าตัวชี้วัดนี้ที่เหมาะสมจะอยู่ในช่วง 0 ถึง 500
ความไม่แน่นอน (Instability)	I	ค่าตัวชี้วัดที่เกี่ยวข้องกับสัดส่วนความสัมพันธ์ของเอฟเฟอเรนท์คัปปลิงและแอฟเฟอเรนท์คัปปลิง โดยมีสัมพันธ์ดังสมการที่ 2.2 $$I = \frac{Ce}{Ce+Ca} \quad (2.2)$$ โดยค่าความไม่แน่นอนนี้อยู่ในช่วง 0 ถึง 1 โดยที่ค่าความไม่แน่นอนเข้าใกล้ 1 หมายความว่าแพ็กเกจนั้นมีความเป็นไปได้ง่ายที่จะเกิดการเปลี่ยนแปลงหรือมีความไม่เสถียร แต่ถ้าค่าความไม่แน่นอนเข้าใกล้ 0 หมายความว่าแพ็กเกจนั้นมีความเสถียรและมีความเป็นไปได้ยากที่จะถูกเปลี่ยนแปลงแก้ไข

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.6 หลักการการพึ่งพาที่มีเสถียรภาพ

หลักการการพึ่งพาที่มีเสถียรภาพ (the stable-dependencies principle หรือ SDP) [9] เป็นหลักการที่ว่าด้วยแพ็กเกจในระบบควรที่จะพึ่งพาแพ็กเกจที่มีความเสถียรภาพไม่น้อยกว่าความเสถียรภาพของตัวแพ็กเกจเอง โดยตัวชี้วัดความเสถียรภาพ (stability metrics) ประกอบด้วย

- เอฟเฟอเรนท์คัปปลิง (Efferent Coupling หรือ Ce)
- แอฟเฟอเรนท์คัปปลิง (Afferent Coupling หรือ Ca)
- ความไม่แน่นอน (Instability หรือ I)

2.1.7 การรีแฟคทอริงเชิงสถาปัตยกรรม

การรีแฟคทอริงเชิงสถาปัตยกรรม (Architecture refactoring) [1] หมายถึง กระบวนการเปลี่ยนแปลงของระบบซอฟต์แวร์เพื่อปรับปรุงโครงสร้างภายในของระบบซอฟต์แวร์ โดยที่ไม่ก่อให้เกิดการเปลี่ยนแปลงทางพฤติกรรมการทำงานของระบบ โดยกระบวนการรีแฟคทอริงเพื่อปรับปรุงปัญหาทางโครงสร้างสถาปัตยกรรมจะเรียกว่า การรีแฟคทอริงเชิงสถาปัตย กรรม (architecture refactoring) หรือในบางงานวิจัยอาจมีการเรียกในชื่อที่แตกต่างกัน เช่น การรีแฟคทอริงที่มีผลกระทบสูง (high-impact refactoring), การแปลงทางสถาปัตยกรรม (architecture transformation) หรือการรีแฟคทอริงขนาดใหญ่ (large refactoring) [10]

วิธีการรีแฟคทอริงเชิงสถาปัตยกรรมโดยพิจารณาจากโครงสร้างและความหมาย (structural and semantic information) ประกอบด้วย 2 ขั้นตอน คือ การระบุส่วนประกอบที่ควรจะถูกรีแฟคทอริง และการดำเนินการรีแฟคทอริงแบบอัตโนมัติ โดยวิธีการรีแฟคทอริงที่น่าสนใจ คือ การรีแฟคทอริงด้วยการแยกแพ็กเกจ (extract package refactoring หรือ EPR) โดยจะประเมินความสัมพันธ์ของคลาสในในแพ็กเกจและเก็บความสัมพันธ์ระหว่างคลาสในรูปแบบเมทริกซ์ (class-by-class matrix) โดยพิจารณาว่าคลาสที่มีความคล้ายคลึงกันควรอยู่ในแพ็กเกจเดียวกัน ซึ่งจัดเป็นวิธีการจัดการรีแฟคทอริงรูปแบบหนึ่งที่ส่งผลกระทบต่อตัวชี้วัดคุณภาพ [11]

กระบวนการรีแฟคทอริงเชิงสถาปัตยกรรมในปัจจุบัน ถึงแม้ว่าจะมีเครื่องมือที่สนับสนุนการรีแฟคทอริงเชิงสถาปัตยกรรมรวมถึงเครื่องมือในเชิงพาณิชย์ที่ช่วยสนับสนุนการตรวจหาอาคิเทกเจอร์สเมล แต่ก็ยังไม่คลอบคลุมในหลายแง่มุม [10] เช่น จำนวนของอาคิเทกเจอร์สเมลที่ตรวจหายังไม่คลอบคลุมเพียงพอ หรือความสามารถของเครื่องมือที่ใช้ในการพัฒนา (Integrated Development Environments หรือ IDE) ถึงแม้ว่าจะสามารถช่วยนักพัฒนาในกระบวนการรีแฟคทอริงแบบอัตโนมัติได้ แต่ก็ยังขาดการสนับสนุนในการตรวจหาอาคิเทกเจอร์สเมลที่เกิดขึ้นในขณะกำลังพัฒนาระบบซอฟต์แวร์

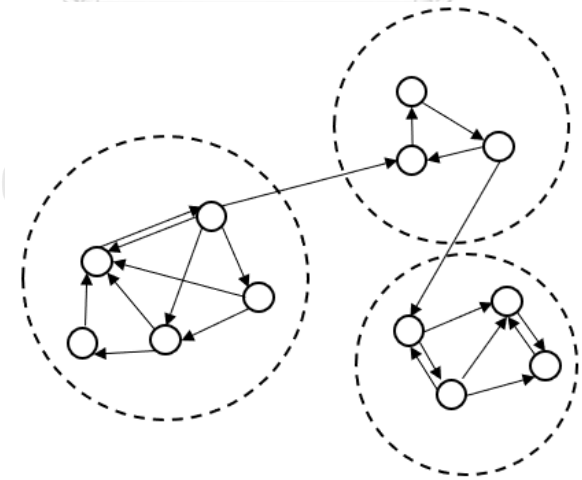
2.1.8 การตรวจหาคอมมิวนิตี

คอมมิวนิตี (community) หรือกลุ่ม (cluster) ของข้อมูลในเครือข่าย (network) หรือกราฟ เป็นการจัดกลุ่มข้อมูลที่มีความน่าจะเป็นสูงที่จะมีการเชื่อมต่อกันของสมาชิกภายในกลุ่มมากกว่าสมาชิกในกลุ่มอื่น ๆ โดยเครือข่ายนั้นถูกใช้อย่างแพร่หลายในการแสดงความสัมพันธ์ของข้อมูลขนาดใหญ่ ตัวอย่างเช่น เครือข่ายสังคม (social network) ที่แสดงความสัมพันธ์ของคนจำนวนมาก โดยเครือข่ายต่าง ๆ นั้นจะมีรูปแบบโครงสร้างของคอมมิวนิตี (community structure) เพื่อแสดงถึงความสัมพันธ์ของข้อมูลในเครือข่าย

การตรวจหาคอมมิวนิตี (community detection) [12, 13] ในเครือข่ายนั้น ในบางงานวิจัยอาจจะเรียกว่าการจัดกลุ่มกราฟ (graph clustering) หรือการจัดกลุ่มเครือข่าย (network clustering) จะพิจารณาถึงกลุ่มของโหนดที่มีการแบ่งปันหรือมีลักษณะบางอย่างที่ร่วมกันหรือคล้ายคลึงกัน โดยประเภทของการจัดกลุ่มในเครือข่ายที่มีทิศทาง (directed network) นั้นประกอบด้วย 2 ประเภทหลัก

(1) การจัดกลุ่มบนพื้นฐานของความหนาแน่น (Density-based clusters)

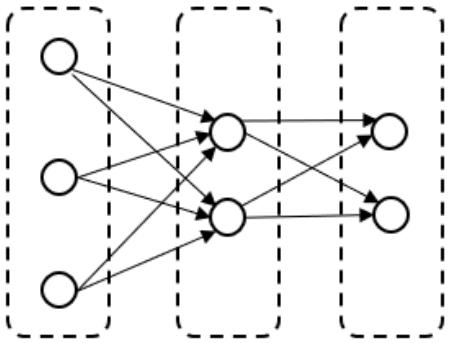
การจัดกลุ่มของโหนดที่ถูกจัดในรูปแบบดั้งเดิมที่อยู่บนพื้นฐานของการพิจารณาความหนาแน่นของเส้นเชื่อม ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างการจัดกลุ่มบนพื้นฐานของความหนาแน่น

(2) การจัดกลุ่มบนพื้นฐานของรูปแบบ (Pattern-based clusters)

การจัดกลุ่มของโหนดที่ไม่พิจารณาความหนาแน่นของเส้นเชื่อมแต่พิจารณาจากรูปแบบการเคลื่อนที่การไหลเวียนที่เกิดขึ้นภายในโหนดต่าง ๆ ของแต่ละกลุ่ม ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างการจัดกลุ่มบนพื้นฐานของรูปแบบ

2.1.9 อัลกอริทึมลูแวน

อัลกอริทึมลูแวน (Louvain algorithm) [14, 15] เป็นวิธีการตรวจหาคอมมิวนิตีในเครือข่ายที่พิจารณาจากคะแนนความเป็นโมดูล (modularity score หรือค่า Q) ของแต่ละคอมมิวนิตี โดยที่ความเป็นโมดูลจะประเมินจากคุณภาพของโหนดในแต่ละคอมมิวนิตีว่ามีการเชื่อมต่อกันในคอมมิวนิตีหนาแน่นมากเพียงใดเมื่อเปรียบเทียบกับการเชื่อมต่อกับคอมมิวนิตีอื่น ๆ ในแบบสุ่ม โดยอัลกอริทึมนี้มีขั้นตอนการทำงานดังนี้

(1) เริ่มจากให้แต่ละโหนดในกราฟเป็นคอมมิวนิตี

(2) พิจารณาแต่ละโหนดในกราฟ

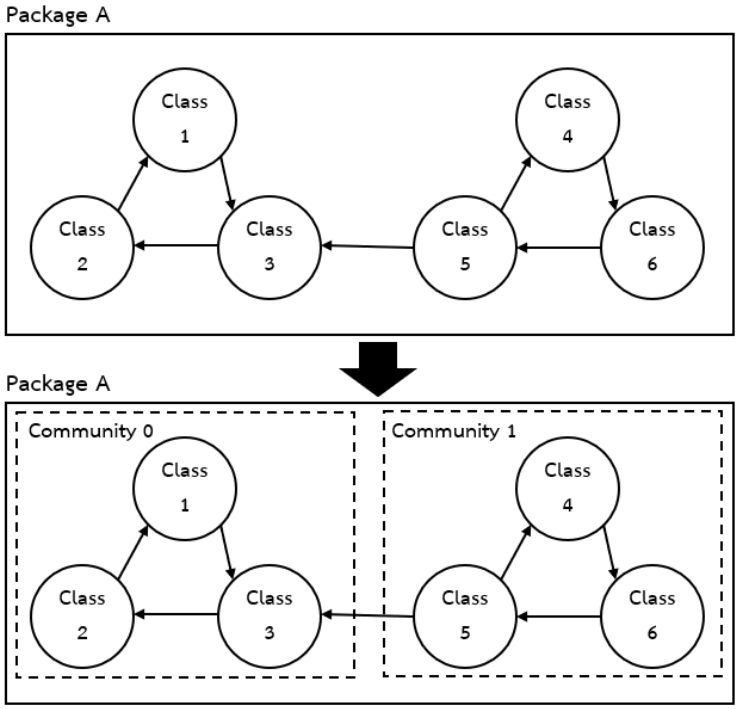
(2.1) นำโหนดออกจากคอมมิวนิตีปัจจุบัน

(2.2) เพิ่มโหนดไปยังคอมมิวนิตีที่เชื่อมต่อกับโหนดนั้นที่ให้ค่าความเป็นโมดูลสูงที่สุด

(3) ทำซ้ำในขั้นตอนที่ (2) จนกระทั่งไม่เกิดการเปลี่ยนแปลงคอมมิวนิตี

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โดยข้อดีของอัลกอริทึมการตรวจหาคอมมิวนิตีแบบลูแวนคือเป็นอัลกอริทึมที่ทำงานได้รวดเร็วและทำงานได้ดีในเครือข่ายขนาดใหญ่ โดยตัวอย่างการตรวจหาคอมมิวนิตีด้วยอัลกอริทึมลูแวนแสดงดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างการตรวจหาคอมมิวนิตีในเครือข่ายด้วยอัลกอริทึมลูแวน

จากรูปที่ 2.4 แสดงความสัมพันธ์ของคลาส 6 คลาสในแพ็กเกจเอ ซึ่งจัดเป็นเครือข่ายความสัมพันธ์หนึ่ง และเมื่อใช้อัลกอริทึมลูแวนในการตรวจหาคอมมิวนิตีแล้ว จะตรวจพบจำนวน 2 คอมมิวนิตีซึ่งประกอบด้วย คอมมิวนิตี 0 มีสมาชิกคือ คลาส 1, คลาส 2 และคลาส 3 และคอมมิวนิตี 1 มีสมาชิกคือ คลาส 4, คลาส 5 และคลาส 6

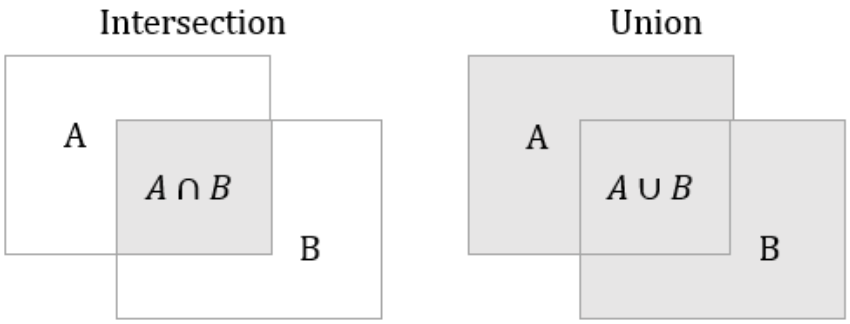
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.10 การประเมินประสิทธิภาพ

การประเมินประสิทธิภาพเครื่องมือหรือระบบที่ได้พัฒนาขึ้นว่ามีประสิทธิภาพในการทำงานเพียงใด เมื่อต้องเปรียบเทียบกับอีกเครื่องมือหนึ่งสามารถพิจารณาได้จากค่าอินเตอร์เซกชันโอเวอร์ยูเนียน

ค่าอินเตอร์เซกชันโอเวอร์ยูเนียน (Intersection over Union หรือ IoU) หรือในบางงานวิจัยเรียกว่าค่าสัมประสิทธิ์ความละม้ายแจ็กการ์ด (Jaccard similarity coefficient) ถูกใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างชุดตัวอย่างเพื่อใช้ในการประเมินประสิทธิภาพ ค่าอินเตอร์เซกชันโอเวอร์ยูเนียนนี้เป็นสัดส่วนของขนาดชุดตัวอย่างที่มีการอินเตอร์เซกชันกับขนาดของชุดตัวอย่างที่มีการยูเนียน โดยที่ค่าอินเตอร์เซกชันโอเวอร์ยูเนียนอยู่ในช่วง 0 ถึง 1 โดยที่ค่าเข้าใกล้หนึ่งแสดงถึงชุดตัวอย่างทั้งสองมีความคล้ายคลึงกันมาก และในทางกลับกันค่าเข้าใกล้ศูนย์แสดงถึงความแตกต่างกัน

ของชุดตัวอย่างทั้งสองที่นำมาทดสอบ โดยรูปที่ 2.5 แสดงแผนภาพและการคำนวณค่าอินเตอร์เซกชันโอเวอร์ยูเนียน



$$\text{Intersection over Union (IoU)} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

รูปที่ 2.5 แผนภาพและสมการการคำนวณค่าอินเตอร์เซกชันโอเวอร์ยูเนียน

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ สามารถแบ่งออกเป็น 2 ส่วนประกอบด้วย งานวิจัยที่เกี่ยวข้องกับการตรวจหาอาคิเทกเจอร์สเมลและงานวิจัยที่เกี่ยวข้องกับกระบวนการรีแฟคทอริงในเชิงสถาปัตยกรรม โดยมีรายละเอียดดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.2.1 งานวิจัยที่เกี่ยวข้องกับการตรวจหาอาคิเทกเจอร์สเมล

งานวิจัยในกลุ่มนี้ เป็นงานวิจัยเพื่อศึกษารูปแบบอาคิเทกเจอร์สเมลต่าง ๆ ในงานวิจัยที่ผ่านมา เพื่อศึกษารูปแบบปัญหาของอาคิเทกเจอร์สเมล และวิธีการตรวจหาที่แต่ละงานวิจัยได้นำเสนอ โดยมีรายละเอียดงานวิจัยที่เกี่ยวข้องดังนี้

(1) งานวิจัย “Identifying Architectural Bad Smells”

งานวิจัยของ Joshua Garcia และคณะ [6] ในปี 2009 ได้นำเสนอเกี่ยวกับการออกแบบในสถาปัตยกรรมซอฟต์แวร์ (software architecture) สามารถมีผลกระทบด้านลบต่อการบำรุงรักษาระบบ โดยนำเสนอแนวคิดของอาคิเทกเจอร์สเมลซึ่งเป็นสิ่งที่แฝงอยู่และมีผลกระทบร้ายแรงต่อระบบ นอกจากนี้ยังเป็นสิ่งที่เกิดขึ้นบ่อยครั้งตั้งแต่การออกแบบซอฟต์แวร์ โดยงานวิจัยนี้ได้นำเสนอรูปแบบของอาคิเทกเจอร์สเมลใน 4 รูปแบบ ประกอบด้วย

- คอนเนคเตอร์เอ็นวี (Connector Envy)
- สแคตเตอร์ฟังก์ชันนอลริตี (Scattered Functionality)
- แอมบิกิวอัสอินเตอร์เฟส (Ambiguous Interfaces)
- เอ็กซ์ตร้านิวอัสคอนเนคเตอร์ (Extraneous Connector)

โดยงานวิจัยนี้ได้นำเสนอแผนภาพ (schematic diagram) ของทั้ง 4 อาคิเทกเจอร์สเมล เพื่อให้สถาปนิกระบบ (system architects) ใช้ในการตรวจหาอาคิเทกเจอร์สเมลโดยการเปรียบเทียบกับโครงสร้างของระบบซอฟต์แวร์ที่ตรวจสอบ

(2) งานวิจัย "Automatic Detection of Instability Architectural Smells"

งานวิจัยของ Francesca Arcelli Fontana และคณะ [4] ในปี 2016 ได้นำเสนอการตรวจหาอาคิเทกเจอร์สเมลโดยใช้เทคนิคบนพื้นฐานของกราฟการพึ่งพา (dependency graph) และการดึงข้อมูลของระบบไปจัดเก็บในรูปแบบของฐานข้อมูลกราฟเพื่อคำนวณหาค่าตัวชี้วัด โดยงานวิจัยนี้นำเสนออาคิเทกเจอร์สเมลใน 3 รูปแบบ ประกอบด้วย

- อันสเตเบิลดีเพนเดนซี (Unstable Dependency)
- ฮับไลค์ดีเพนเดนซี (Hub Like Dependency)
- ไซคลิกดีเพนเดนซี (Cyclic Dependency)

โดยงานวิจัยนี้แสดงให้เห็นถึงการนำกราฟและอัลกอริทึมที่เกี่ยวข้องกับกราฟมาใช้ในการตรวจหาอาคิเทกเจอร์สเมล นอกจากนี้ในการทดลองได้พัฒนาเครื่องมือที่เรียกว่า อาคาน (Arcan) โดยเปรียบเทียบผลการทดลองกับผลลัพธ์ที่ได้จากเครื่องมืออื่น ๆ ประกอบด้วย อินฟิวชัน (infusion) และฮอตสปอตดีเทคเตอร์ (Hotspot Detector) [7] โดยให้ผลลัพธ์ไปในทิศทางเดียวกัน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.2.2 งานวิจัยที่เกี่ยวข้องกับกระบวนการรีแฟคทอริงเชิงสถาปัตยกรรม

งานวิจัยในกลุ่มนี้ เป็นงานวิจัยเพื่อศึกษากระบวนการสำหรับการรีแฟคทอริงเชิงสถาปัตยกรรม ในงานวิจัยที่ผ่านมา โดยมีรายละเอียดงานวิจัยที่เกี่ยวข้องดังนี้

(1) งานวิจัย "Support for architectural smell refactoring"

งานวิจัยของ Luca Rizzi และคณะ [16] ในปี 2018 ได้นำเสนอขั้นตอนการกำจัดอาคิเทกเจอร์สเมลในรูปแบบไซคลิกดีเพนเดนซี ในโปรแกรมที่พัฒนาด้วยภาษาจาวา โดยเครื่องมือที่พัฒนานั้นเป็นส่วนขยายของเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลที่ชื่อว่าอาคาน [4]

อาคิเทกเจอร์สเมลรูปแบบไซคลิกดีเพนเดนซีเป็นอาคิเทกเจอร์สเมลที่ถูกพบเจอได้ทั่วไป โดยหมายถึงระบบย่อยหรือส่วนประกอบที่มีการเกี่ยวข้องเชื่อมโยงกันเป็นห่วงโซ่ความสัมพันธ์ ในงานวิจัย

นี้ได้เสนอเครื่องมือที่สนับสนุนกระบวนการรีแฟคทอริงโดยหลังจากตรวจพบอาคิเทกเจอร์สเมลนี้ผ่านเครื่องมืออาคาน จากนั้นตัวเครื่องมือจะแนะนำขั้นตอนการรีแฟคทอริงให้นักพัฒนา โดยอาศัยค่าฟังก์ชันความสำคัญ (priority function) เป็นตัวระบุเส้นเชื่อมความสัมพันธ์ที่เกี่ยวข้องกับอาคิเทกเจอร์สเมลรูปแบบไซคลิกดีเพนเดนซี โดยมีขั้นตอนการทำงานประกอบด้วย

1. ใช้เครื่องมืออาคานในการวิเคราะห์เพื่อที่จะตรวจหาอาคิเทกเจอร์สเมลรูปแบบไซคลิกดีเพนเดนซี โดยการสร้างกราฟความสัมพันธ์และเก็บข้อมูลลงในฐานข้อมูลกราฟ
2. กำหนดค่าฟังก์ชันความสำคัญ ซึ่งอยู่บนพื้นฐานของตัวชี้วัดที่ถูกนำเสนอโดยมาร์ติน (Robert Cecil Martin)
3. กระบวนการกำจัดอาคิเทกเจอร์สเมลรูปแบบไซคลิกดีเพนเดนซีด้วยกระบวนการรีแฟคทอริงประกอบด้วย 2 ขั้นตอน คือ

    (1) รีแฟคทอริงโดยสเมล (refactoring by smells) โดยเครื่องมือจะคำนวณค่าฟังก์ชันความสำคัญ โดยพิจารณาจากอาคิเทกเจอร์สเมลที่เกิดขึ้นทุก ๆ โหนดในกราฟ

    (2) รีแฟคทอริงโดยรูปร่าง (refactoring by shapes) โดยเครื่องมือจะคำนวณค่าฟังก์ชันความสำคัญโดยพิจารณาจากรูปร่างของโหนด

จากงานวิจัยนี้แสดงให้เห็นถึงกระบวนการรีแฟคทอริงเพื่อกำจัดอาคิเทกเจอร์สเมลนั้นเป็นกระบวนการที่มีความซับซ้อนสูง เมื่อกำจัดอาคิเทกเจอร์สเมลที่เกิดขึ้นแล้ว มีโอกาสที่จะทำให้อาคิเทกเจอร์สเมลรูปแบบอื่นเกิดขึ้นตามมาได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

(2) งานวิจัย “Refactoring Software Packages via Community Detection in Complex Software Networks”

งานวิจัยของ Wei-Feng Pan และคณะ [17] ในปี 2013 ได้นำเสนอวิธีการรีแฟคทอริงระบบซอฟต์แวร์ในระดับแพ็กเกจ โดยใช้เครือข่ายการพึ่งพาของคลาสต่าง ๆ และให้ผลลัพธ์ออกมาเป็นคลาสที่ควรจะถูกรีแฟคทอริง วิธีการที่ถูกนำเสนอในงานวิจัยนี้เป็นการตรวจหาคอมมิวนิตี (community detection) ประกอบด้วย 3 ขั้นตอน

1. สร้างเครือข่ายการพึ่งพาของคลาสต่าง ๆ แบบมีน้ำหนัก แต่ไม่มีทิศทาง ในระบบซอฟต์แวร์ โดยที่โหนดต่าง ๆ ในเครือข่ายแทนด้วยคลาส และเส้นเชื่อมคือความสัมพันธ์ของคลาสคู่ต่าง ๆ และน้ำหนักของเส้นเชื่อมแทนความแข็งแกร่งในการพึ่งพาของการเชื่อมต่อของคลาส
2. ใช้อัลกอริทึมการตรวจหาคอมมิวนิตีเพื่อตรวจหาโครงสร้างของกลุ่มต่าง ๆ ที่สอดคล้องกับโครงสร้างของแพ็กเกจที่เหมาะสม

3. นำผลลัพธ์ที่ได้คือรายการของคลาสที่ถูกแนะนำว่าควรจะรีแฟคทอริงมาเพื่อเปรียบเทียบโครงสร้างแพ็กเกจที่เหมาะสมกับโครงสร้างแพ็กเกจจริงในระบบซอฟต์แวร์

ในงานวิจัยนี้นำวิธีการที่นำเสนอมาทดสอบกับระบบที่พัฒนาด้วยภาษาจาวาจำนวน 2 ระบบ และตรวจสอบผลการรีแฟคทอริงโดยการให้วิศวกรซอฟต์แวร์จำนวน 5 คนช่วยในการยืนยันว่าผลลัพธ์ที่ได้จากการรีแฟคทอริงนั้น คลาสต่าง ๆ ในแพ็กเกจมีความเชื่อมแน่นกัน (cohesion) มากขึ้น และมีการคู่ควบกัน (coupling) น้อยลง ดังนั้นจึงช่วยปรับปรุงโครงสร้างของซอฟต์แวร์ให้ดีขึ้น

จากการนำเสนอในงานวิจัยนี้ แสดงให้เห็นถึงการจัดกลุ่มในระดับคลาสนั้นสามารถปรับปรุงคุณภาพซอฟต์แวร์ให้ดีขึ้นได้ แต่อย่างไรก็ตามในงานวิจัยนี้ยังไม่ได้มีการนำเสนอถึงผลกระทบในระดับโครงสร้างสถาปัตยกรรมที่เกิดการเปลี่ยนแปลงหลังจากการจัดกลุ่มจึงเป็นจุดที่น่าสนใจที่งานวิจัยฉบับนี้จะนำเสนอต่อไป

จากการพิจารณางานวิจัยที่เกี่ยวข้องใน 2 ส่วนที่ได้นำเสนอไปสามารถสรุปรายละเอียดที่เกี่ยวข้องกับงานวิจัยได้ดังนี้

ส่วนแรกเป็นงานวิจัยที่เกี่ยวข้องกับการตรวจหาอาคิเทกเจอร์สเมล ซึ่งจากงานวิจัยที่ผ่านมาได้นำเสนอรูปแบบอาคิเทกเจอร์สเมลรูปแบบต่าง ๆ รวมถึงผลกระทบที่เกิดขึ้นจากอาคิเทกเจอร์สเมลนั้น รวมถึงนำเสนอวิธีการตรวจหาอาคิเทกเจอร์สเมลในวิธีการที่แตกต่างกัน โดยในงานวิจัยนี้พบว่าอาคิเทกเจอร์สเมลในรูปแบบอันสเตเบิลดีเพนเดนซี เป็นอาคิเทกเจอร์สเมลที่สามารถเกิดขึ้นได้โดยทั่วไปและเป็นเรื่องยากในการตรวจหาและคาดการณ์การเกิดโดยนักพัฒนาซอฟต์แวร์เนื่องจากอาคิเทกเจอร์สเมลรูปแบบนี้เกิดจากการพึ่งพากันของแพ็กเกจต่าง ๆ ในระบบซอฟต์แวร์ โดยการพึ่งพาของแพ็กเกจต้องพิจารณาถึงความสัมพันธ์ในระดับคลาสต่าง ๆ ภายในแพ็กเกจ ดังนั้นเมื่อมีการเปลี่ยนแปลงทางโครงสร้างสถาปัตยกรรม เช่น การพัฒนาแพ็กเกจใหม่ หรือการพัฒนาคลาสใหม่ย่อมส่งผลกระทบต่อการพึ่งพากันระหว่างแพ็กเกจ และสามารถนำไปสู่การเกิดอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีขึ้นได้ ดังนั้นงานวิจัยนี้จึงเลือกที่จะประเมินผลกระทบของอาคิเทกเจอร์สเมลรูปแบบนี้หลังจากกระบวนการรีแฟคทอริง เพื่อให้แน่ใจว่ากระบวนการรีแฟคทอริงไม่ก่อให้เกิดอาคิเทกเจอร์สเมลรูปแบบนี้เพิ่มขึ้น

ส่วนที่สองเป็นงานวิจัยที่เกี่ยวข้องกับกระบวนการรีแฟคทอริงเชิงสถาปัตยกรรม โดยงานวิจัยที่ผ่านมาได้นำเสนอวิธีการรีแฟคทอริงเพื่อปรับปรุงคุณภาพซอฟต์แวร์ โดยในงานวิจัยแรกได้นำเสนอการรีแฟคทอริงเพื่อกำจัดอาคิเทกเจอร์สเมลรูปแบบไซคลิกดีเพนเดนซี ซึ่งเป็นอาคิเทกเจอร์สเมลรูปแบบการพึ่งพารูปแบบหนึ่งที่สามารถพบได้โดยทั่วไป โดยนำเสนอวิธีการที่จะลดอาคิเทกเจอร์สเมลรูปแบบนี้ลง โดยที่กระบวนการรีแฟคทอริงนั้นมุ่งไปที่การกำจัดอาคิเทกเจอร์สเมลรูปแบบนี้โดยไม่ได้

พิจารณาถึงอาคิเทกเจอร์สเมลในรูปแบบอื่น ๆ ที่อาจเกิดขึ้นตามมา และงานวิจัยที่สองได้นำเสนอการตรวจหาคอมมิวนิตีในการจัดกลุ่มตามโครงสร้างความสัมพันธ์ของคลาสเพื่อที่จะแนะนำแพ็กเกจที่เหมาะสมสำหรับกลุ่มของคลาสต่าง ๆ โดยในงานวิจัยนี้ใช้วิธีการตรวจหาคอมมิวนิตีเพื่อการย้ายคลาสไปยังแพ็กเกจที่เหมาะสมโดยไม่ได้พิจารณาถึงผลกระทบทางอาคิเทกเจอร์สเมลที่อาจเกิดขึ้นก่อนและหลังกระบวนการรีแฟคทอริง ดังนั้นในงานวิจัยนี้จึงสนใจวิธีการตรวจหาคอมมิวนิตีที่ช่วยในการจัดกลุ่มคลาสมาใช้ในงานวิจัย โดยจะนำมาใช้กับกระบวนการรีแฟคทอริงด้วยการแยกแพ็กเกจ และจะพิจารณาถึงผลกระทบที่อาจจะก่อให้เกิดอาคิเทกเจอร์สเมลในรูปแบบอื่นเกิดขึ้นตามมา โดยในเบื้องต้นงานวิจัยนี้จะพิจารณาการเกิดขึ้นของอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี โดยกระบวนการแยกแพ็กเกจจะพิจารณาจากแพ็กเกจขนาดใหญ่เกินเพื่อแยกแพ็กเกจย่อยออกมา เพื่อช่วยให้ระบบง่ายต่อการทำความเข้าใจและการบำรุงรักษา ซึ่งทำให้คุณภาพซอฟต์แวร์ดีขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

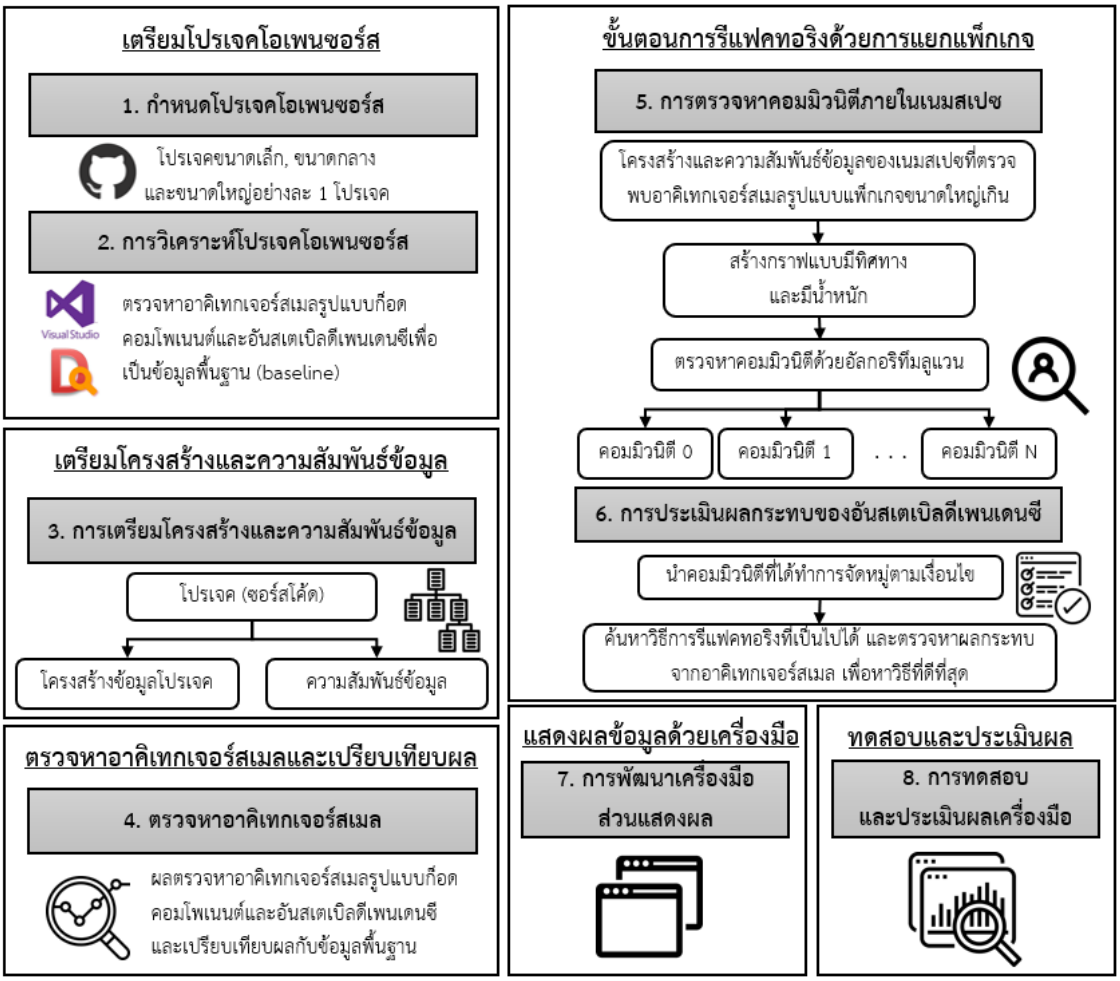
บทที่ 3
วิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี

งานวิจัยนี้นำเสนอวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี อาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน หรือในบางเครื่องมือเรียกว่า ก็อดคอมโพเนนต์ (god component) เป็นอาคิเทกเจอร์สเมลที่เกิดขึ้นได้ง่ายและสามารถพบเห็นได้โดยทั่วไป ซึ่งเกิดจากแพ็กเกจในระบบซอฟต์แวร์มีจำนวนคลาสภายในมากเกินไป ทำให้เป็นเรื่องยากในการทำความเข้าใจและการบำรุงรักษา วิธีการรีแฟคทอริงสามารถทำได้โดยการแยกแพ็กเกจย่อยเพื่อลดขนาดของแพ็กเกจ แต่อย่างไรก็ตามการแยกแพ็กเกจย่อมส่งผลต่อการเปลี่ยนแปลงการพึ่งพาของคลาสต่าง ๆ ในแต่ละแพ็กเกจ ซึ่งอาจก่อให้เกิดผลกระทบหลังจากกระบวนการรีแฟคทอริงในรูปแบบต่าง ๆ ตามมา โดยในงานวิจัยนี้สนใจผลกระทบจากอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาในรูปแบบอันสเตเบิลดีเพนเดนซีที่อาจเกิดขึ้นใหม่หลังจากกระบวนการรีแฟคทอริงที่มีการเปลี่ยนแปลงสัมพันธ์ระหว่างแพ็กเกจ โดยอันสเตเบิลดีเพนเดนซีเป็นอาคิเทกเจอร์สเมลที่เกิดจากแพ็กเกจในระบบซอฟต์แวร์พึ่งพาแพ็กเกจอื่น ๆ ที่มีความเสถียรภาพน้อยกว่า โดยความเสถียรภาพสามารถพิจารณาได้จากค่าตัวชี้วัดความไม่แน่นอน โดยวิธีการรีแฟคทอริงที่นำเสนอในงานวิจัยนี้แบ่งออกเป็น 8 ขั้นตอนหลัก ดังแสดงในรูปที่ 3.1 โดยมีรายละเอียดดังนี้

ส่วนแรกเป็นส่วนการเตรียมโปรเจคโอเพนซอร์ส ซึ่งประกอบด้วย 2 ขั้นตอน คือ ขั้นตอนที่ 1 กำหนดโปรเจคโอเพนซอร์สที่นำมาใช้ในงานวิจัย เพื่อที่จะกำหนดกฎเกณฑ์ในการพิจารณาเลือกโปรเจคที่ใช้ในการทดลองในงานวิจัยนี้ และขั้นตอนที่ 2 การวิเคราะห์โปรเจคโอเพนซอร์ส เพื่อนำโปรเจคที่เลือกมาตรวจหาอาคิเทกเจอร์สเมลและค่าตัวชี้วัดต่าง ๆ ที่เกี่ยวข้องเพื่อนำมาเป็นข้อมูลพื้นฐานในการทดลองสำหรับเปรียบเทียบในขั้นตอนการประเมินผล

3.1.กำหนดโปรเจคโอเพนซอร์สที่นำมาใช้ในงานวิจัย

โปรเจคโอเพนซอร์สที่นำมาใช้ในงานวิจัยนี้ทางผู้วิจัยเลือกพิจารณาโปรเจคโอเพนซอร์สที่ถูกจัดเก็บอยู่บนเว็บไซต์กิตฮับ ซึ่งเว็บไซต์กิตฮับเป็นเว็บไซต์ที่ใช้ในการเก็บข้อมูลรวมถึงควบคุมเวอร์ชันของโปรเจคต่าง ๆ ดังนั้นในงานวิจัยนี้จะพิจารณาเลือกโปรเจคโอเพนซอร์สที่ถูกพัฒนาขึ้นและถูกนำมาใช้งานจริง โดยกำหนดกฎเกณฑ์เบื้องต้นสำหรับการพิจารณาโปรเจคโอเพนซอร์สที่จะนำมาใช้ในงานวิจัยประกอบด้วย



รูปที่ 3.1 ภาพรวมการดำเนินงาน 8 ขั้นตอนหลัก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- ต้องเป็นโปรเจคที่ถูกพัฒนาขึ้นด้วยภาษาซีชาร์ป (C# programing language) บนพื้นฐานของดอตเน็ตเฟรมเวิร์ก (.NET Framework) ตั้งแต่เวอร์ชัน 4.5 หรือสูงกว่า
- ต้องมีโปรเจคการทำงานหลักที่มีจำนวนบรรทัดมากกว่า 3,000 บรรทัด และมีจำนวนเนมสเปซภายในโปรเจคการทำงานหลักตั้งแต่ 3 เนมสเปซขึ้นไป
- ต้องเป็นโปรเจคที่สามารถคอมไพล์ (compile) ผ่านโปรแกรมไมโครซอฟวิชวลสตูดิโอเวอร์ชัน 2017 (Microsoft Visual Studio 2017) ได้อย่างสมบูรณ์

ในงานวิจัยนี้เลือกโปรเจคโอเพนซอร์สจำนวน 3 โปรเจคมาใช้ในการทดลองโดยพิจารณาจากขนาดของโปรเจคการทำงานหลักเป็นข้อกำหนดเพิ่มเติม ดังตาราง 3.1

ตารางที่ 3.3.1 ข้อกำหนดเพิ่มเติมสำหรับโปรเจคที่ใช้ในการทดลอง

ลำดับ	ขนาดโปรเจค	จำนวนเนมสเปซในโปรเจคการทำงานหลัก	จำนวนโปรเจค
1	โปรเจคขนาดเล็ก	ตั้งแต่ 3 ถึง 10 เนมสเปซ	1
2	โปรเจคขนาดกลาง	ตั้งแต่ 11 ถึง 30 เนมสเปซ	1
3	โปรเจคขนาดใหญ่	ตั้งแต่ 31 เนมสเปซขึ้นไป	1

อย่างไรก็ตามเนื่องจากผู้วิจัยเลือกโปรเจคที่ถูกพัฒนาด้วยภาษาซีชาร์ปมาใช้ในงานวิจัย ดังนั้นผู้วิจัยจึงนำเสนอการเปรียบเทียบขอบเขตของโครงสร้างข้อมูลในเบื้องต้นที่ใช้ในงานวิจัยกับภาษาซีชาร์ปที่จะใช้ในการทดลอง ดังต่อไปนี้

- แพ็กเกจ เป็นแนวคิดในการนำเสนอกลุ่มของคลาส โดยในภาษาซีชาร์ป เทียบเคียงได้กับเนมสเปซ
- คลาส จากงานวิจัยที่ผ่านมา ได้ถูกนำมาใช้และเป็นตัวแทนของโครงสร้างที่เทียบเคียงได้ เช่น อินเตอร์เฟซ ดังนั้นในงานวิจัยนี้ จะเรียกรวมโครงสร้างต่าง ๆ ที่เทียบเคียงกับคลาสในภาษาซีชาร์ป ประกอบด้วย คลาส, อินเตอร์เฟซ, สตรัค (struct) และอีนัม (enum) ว่าคลาส เพื่อให้ง่ายและสะดวกในการอ้างอิง

3.2.การวิเคราะห์โปรเจคโอเพนซอร์ส

หลังจากกำหนดโปรเจคโอเพนซอร์สที่นำมาใช้ในการทดลอง ต่อมาเป็นขั้นตอนการวิเคราะห์โปรเจคโอเพนซอร์สที่กำหนด โดยนำโปรเจคโอเพนซอร์สมาวิเคราะห์รายละเอียด และตรวจหาอาคิเทกเจอร์สเมลที่มีแพ็กเกจขนาดใหญ่เกินไปในโปรเจคการทำงานหลัก เพื่อที่จะนำเข้าสู่กระบวนการรีแฟคทอริงด้วยการแยกแพ็กเกจในขั้นตอนถัดไป

การตรวจหาอาคิเทกเจอร์สเมลในเบื้องต้นจะใช้เครื่องมือที่เป็นส่วนขยายของไมโครซอฟวิชวลสตูดิโอที่ชื่อว่า ดีซิกไนต์ (Designite) [18] ซึ่งเป็นเครื่องมือสำหรับการประเมินคุณภาพซอฟต์แวร์เชิงพาณิชย์ที่รองรับการตรวจหาอาคิเทกเจอร์สเมลจำนวนหนึ่ง สำหรับโปรเจคที่ถูกพัฒนาบนพื้นฐานของภาษาซีชาร์ป โดยเวอร์ชันที่นำมาใช้ในการทดลองคือเวอร์ชัน 3.0.3.0 รุ่นทดลองใช้งาน ดังรูปที่ 3.2

โดยผลลัพธ์ที่ได้ในขั้นตอนนี้เป็นผลการตรวจหาอาคิเทกเจอร์สเมลในโปรเจคการทำงานหลักด้วยเครื่องมือนี้ โดยในงานวิจัยนี้จะพิจารณาผลการตรวจหาอาคิเทกเจอร์สเมลใน 2 รูปแบบ เพื่อเป็นข้อมูลพื้นฐาน (baseline) สำหรับการเปรียบเทียบกับเครื่องมือที่จะพัฒนาขึ้น โดยประกอบด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 3.2 เครื่องมือดีซิกไนต์สำหรับการตรวจหาอาคิเทกเจอร์สเมล [18]

- ก็อดคอมโพเนนต์ หรือแพ็กเกจขนาดใหญ่เกิน เป็นอาคิเทกเจอร์สเมลรูปแบบหนึ่ง ซึ่งบ่งชี้ถึงเนมสเปซที่มีจำนวนคลาสภายในที่มากเกินไป หรือมีจำนวนบรรทัดรวมเกินข้อกำหนด
- อันสเตเบิลดีเพนเดนซี เป็นอาคิเทกเจอร์สเมลรูปแบบหนึ่ง ซึ่งบ่งชี้ถึงเนมสเปซที่มีการพึ่งพาเนมสเปซอื่น ๆ ภายในระบบที่มีความเสถียรภาพน้อยกว่า

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

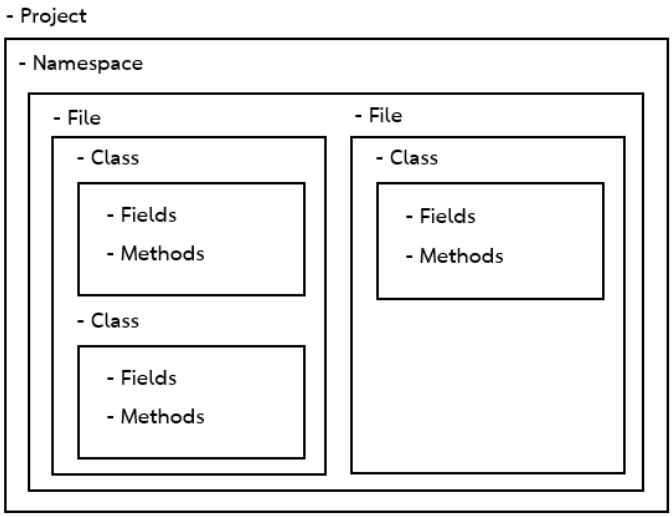
ส่วนถัดมาเป็นส่วนการเตรียมโครงสร้างและความสัมพันธ์ข้อมูล ซึ่งประกอบด้วย ขั้นตอนที่ 3 การเตรียมโปรเจคโอเพนซอร์สให้อยู่ในรูปแบบโครงสร้างและความสัมพันธ์ของข้อมูล เพื่อที่จะแปลงข้อมูลโปรเจคที่ใช้ในการทดลองให้อยู่ในรูปแบบโครงสร้างข้อมูลที่สามารถนำไปวิเคราะห์ต่อได้

3.3.การเตรียมโปรเจคโอเพนซอร์สให้อยู่ในรูปแบบโครงสร้างและความสัมพันธ์ของข้อมูล

ขั้นตอนนี้เป็นขั้นตอนการพัฒนาเครื่องมือสำหรับกระบวนการแปลงโครงสร้างของโปรเจคโอเพนซอร์สจากรูปแบบไฟล์ต่าง ๆ ที่ประกอบด้วยรูปแบบวากยสัมพันธ์ (syntax) ให้อยู่ในรูปแบบโครงสร้างข้อมูลของโปรเจค (project structure) และความสัมพันธ์ของข้อมูลระหว่างคลาส (class relationship) เพื่อที่จะสามารถนำข้อมูลไปใช้ในการวิเคราะห์และกระบวนการรีแฟคทอริงในขั้นตอนถัดไป

3.3.1. การเตรียมโครงสร้างข้อมูลของโปรเจค

ขั้นตอนนี้เป็นการเตรียมโครงสร้างข้อมูลของโปรเจค (project structure) โดยการพัฒนาเครื่องมือบนพื้นฐานของไลบรารีที่ชื่อว่า ไมโครซอฟโค้ดอนาไลซิส (Microsoft.CodeAnalysis) เพื่อช่วยในการวิเคราะห์รูปแบบวากยสัมพันธ์จากไฟล์ต่าง ๆ ในโปรเจคการทำงานหลักให้อยู่ในรูปแบบโครงสร้างของข้อมูลที่มีลำดับชั้น ดังตัวอย่างในรูปที่ 3.3 และมีรายละเอียดของแต่ละชนิดข้อมูล ดังตารางที่ 3.2 นอกจากนี้เครื่องมือที่พัฒนาขึ้นยังเก็บข้อมูลตัวชี้วัดซอฟต์แวร์ในระดับเนมสเปซและระดับคลาส เพื่อใช้ในการวิเคราะห์ในขั้นตอนถัดไป ตัวอย่างเช่น จำนวนคลาส (NOC) ในเนมสเปซ, จำนวนบรรทัด (LOC) ในระดับเนมสเปซและระดับคลาส, เอฟเฟอเรนท์คัปปลิง (Ce), แอฟเฟอเรนท์คัปปลิง(Ca), ความไม่แน่นอน (I), ความเชื่อมแน่นเชิงสัมพันธ์ (H) เป็นต้น



รูปที่ 3.3 แผนภาพโครงสร้างของข้อมูลที่มีลำดับชั้น

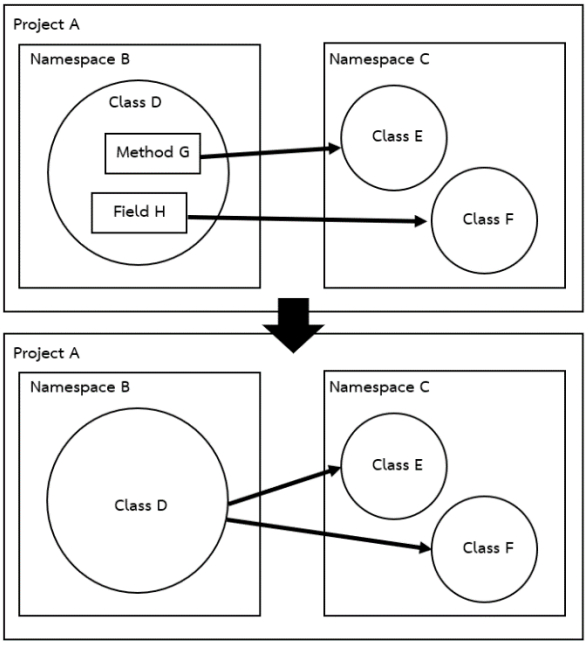
ตารางที่ 3.2 รายละเอียดโครงสร้างข้อมูลที่มีลำดับชั้น

ชนิดข้อมูล	**รายละเอียด**
ข้อมูลโปรเจค (project)	เก็บข้อมูลชื่อโปรเจค และข้อมูลเนมสเปซภายในโปรเจค
ข้อมูลเนมสเปซ (namespace)	เก็บข้อมูลชื่อเนมสเปซ, ข้อมูลคลาสภายในโปรเจค และข้อมูลไฟล์ที่เกี่ยวข้องกับเนมสเปซ
ข้อมูลไฟล์ (file)	เก็บตำแหน่งที่ตั้งของไฟล์ และจำนวนบรรทัด
ข้อมูลคลาส (class)	เก็บข้อมูลชื่อคลาส, ข้อมูลเนมสเปซของคลาส, ข้อมูลไฟล์ของคลาส, ข้อมูลเมธอดและข้อมูลฟิลด์ภายในคลาส

ข้อมูลเมธอด (method)	เก็บข้อมูลชื่อเมธอด, ข้อมูลคลาสของเมธอด และเก็บข้อมูลคลาสต่าง ๆ ของตัวแปรต่าง ๆ ภายในเมธอดที่อ้างอิงถึง
ข้อมูลฟิลด์ (field)	เก็บข้อมูลชื่อฟิลด์ และข้อมูลคลาสของฟิลด์

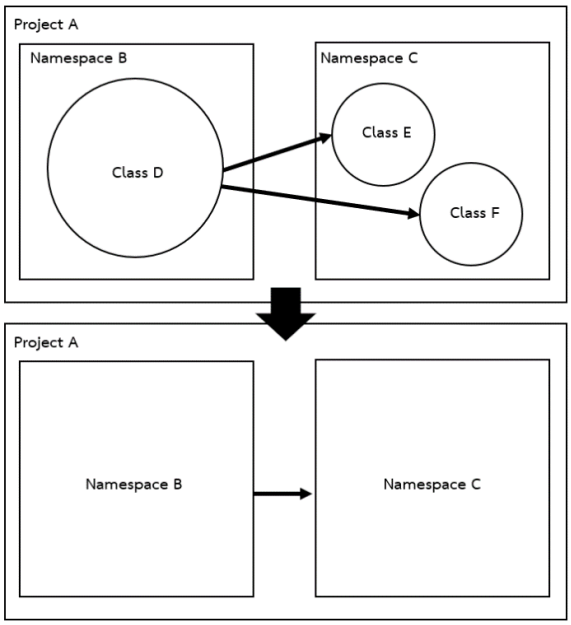
3.3.2. การเตรียมความสัมพันธ์ของข้อมูลระหว่างคลาส

หลังจากได้โครงสร้างของข้อมูลโปรเจคแล้ว ขั้นตอนนี้จะพิจารณาความสัมพันธ์ระหว่างคลาสต่าง ๆ ที่มีการอ้างอิงถึงกันในระดับเมธอดและในระดับฟิลด์ โดยจะพิจารณาถึงความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเดียวกัน และต่างเนมสเปซ เพื่อสร้างความสัมพันธ์ของข้อมูลในระดับคลาสต่าง ๆ ในโปรเจค ดังรูปที่ 3.4 และหลังจากนั้นจะพิจารณาความสัมพันธ์ระดับเนมสเปซ เพื่อสร้างความสัมพันธ์ระหว่างเนมสเปซต่าง ๆ ในโปรเจค ดังรูปที่ 3.5



รูปที่ 3.4 การพิจารณาความสัมพันธ์ของข้อมูลระหว่างคลาส

จากรูปที่ 3.4 การพิจารณาความสัมพันธ์ระหว่างคลาส แสดงให้เห็นว่าเมธอดจี (method G) มีการพึ่งพาคลาสอี (class E) และฟิลด์เอช (field H) มีการพึ่งพาคลาสเอฟ (class F) ซึ่งเมธอดจีและฟิลด์เอชเป็นสมาชิกในคลาสดี (class D) ดังนั้นสามารถสรุปความสัมพันธ์ของข้อมูลในระดับคลาสได้คือ คลาสดีพึ่งพาคลาสอี และคลาสดีพึ่งพาคลาสเอฟ



รูปที่ 3.5 การพิจารณาความสัมพันธ์ของข้อมูลระหว่างเนมสเปซ

จากรูปที่ 3.5 การพิจารณาความสัมพันธ์ระหว่างเนมสเปซ แสดงให้เห็นถึงคลาสอี ในเนมสเปซบี (namespace B) มีการพึ่งพาคลาสอีและคลาสเอฟ ในเนมสเปซซี (namespace C) ดังนั้นสามารถสรุปความสัมพันธ์ในระดับเนมสเปซได้ คือ เนมสเปซบีพึ่งพาเนมสเปซซี

ส่วนถัดมาเป็นส่วนการตรวจหาอาคิเทกเจอร์สเมลและการเปรียบเทียบผล ประกอบด้วย ขั้นตอนที่ 4 การตรวจหาอาคิเทกเจอร์สเมลจากโครงสร้างและความสัมพันธ์ของข้อมูล เพื่อพัฒนาเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลจากโครงสร้างข้อมูลของโปรเจคและความสัมพันธ์ของข้อมูลระหว่างคลาส และเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมลกับเครื่องมือเชิงพาณิชย์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.4.การตรวจหาอาคิเทกเจอร์สเมล

ในขั้นตอนนี้จะพัฒนาฟังก์ชันในการตรวจหาอาคิเทกเจอร์สเมลจากโครงสร้างข้อมูลของโปรเจคและความสัมพันธ์ของข้อมูลระหว่างคลาส

3.4.1. การตรวจหาแพ็กเกจขนาดใหญ่เกิน

การตรวจหาแพ็กเกจขนาดใหญ่เกินสามารถพิจารณาจากเนมสเปซในโปรเจคที่มีจำนวนคลาสภายในเนมสเปซที่มากกว่า 30 คลาสหรือมีจำนวนบรรทัดของโค้ดในไฟล์ต่าง ๆ ภายใน

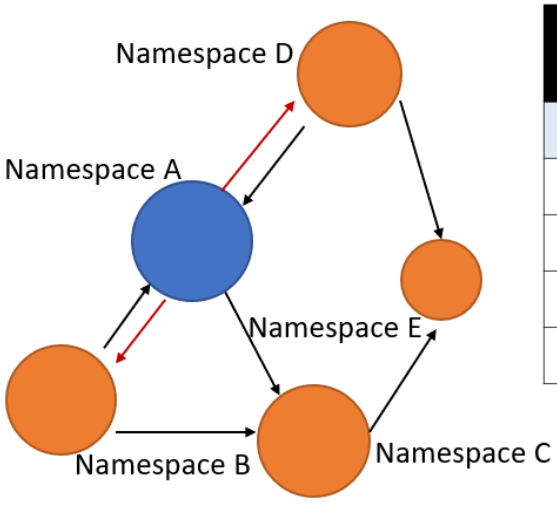
เนมสเปซ มากกว่า 27,000 บรรทัด และเมื่อพบเนมสเปซตามเงื่อนไขจะเก็บข้อมูลเนมสเปซนั้นไว้เพื่อเปรียบเทียบกับข้อมูลพื้นฐาน

โดยการตรวจหาแพ็กเกจขนาดใหญ่เกินในงานวิจัยนี้ จะนำข้อมูลจำนวนคลาสและจำนวนบรรทัดของโค้ดในระดับเนมสเปซที่ถูกจัดเก็บในโครงสร้างข้อมูลโปรเจคมาตรวจสอบหาเนมสเปซที่เป็นไปตามเงื่อนไขที่กำหนด

3.4.2. การตรวจหาอันสเตเบิลดีเพนเดนซี

การตรวจหาอันสเตเบิลดีเพนเดนซีจะพิจารณาจากความสัมพันธ์ของเนมสเปซในโปรเจคโดยการคำนวณค่าตัวชี้วัดความไม่แน่นอนซึ่งสามารถพิจารณาจากความสัมพันธ์ขาเข้าและขาออกจากเนมสเปซ โดยได้อธิบายรายละเอียดไว้ก่อนหน้าในตารางที่ 2.1 หลังจากได้ค่าความไม่แน่นอนของแต่ละเนมสเปซในโปรเจคแล้ว จะพิจารณาทีละเนมสเปซโดยเปรียบเทียบว่าเนมสเปซที่พึ่งพาอยู่มีค่าความไม่แน่นอนสูงกว่าหรือไม่ ถ้ามีค่าความไม่แน่นอนสูงกว่าแสดงถึงเนมสเปซที่พึ่งพานั้นมีความเสถียรภาพน้อยกว่า และเมื่อพบเนมสเปซตามเงื่อนไขจะเก็บข้อมูลเนมสเปซนั้นไว้เพื่อเปรียบเทียบกับข้อมูลพื้นฐาน

โดยการตรวจหาอันสเตเบิลดีเพนเดนซีในงานวิจัยนี้ จะนำข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์ของข้อมูลระหว่างคลาสมาใช้ในการพิจารณา เริ่มจากการพิจารณาค่าเอฟเฟอเรนท์คัปปลิง (Ce) ที่บ่งชี้จำนวนการพึ่งพาคลาสภายนอกของเนมสเปซอื่น ๆ และนำค่าความไม่แน่นอนของเนมสเปซที่กำลังพิจารณาเปรียบเทียบกับเนมสเปซที่พึ่งพา โดยเนมสเปซที่เป็นอันสเตเบิลดีเพนเดนซีจะพึ่งพาเนมสเปซที่มีความเสถียรน้อยกว่า หรือมีค่าความไม่แน่นอนที่สูงกว่า



Namespace	Ce	Ca	I	Dependency NS
Namespace A	3	2	0.60	B,C,D
Namespace B	2	1	0.66	A,C
Namespace C	1	2	0.33	E
Namespace D	2	1	0.66	A,E
Namespace E	0	2	0.00	

รูปที่ 3.6 ตัวอย่างการตรวจหาอันสเตเบิลดีเพนเดนซี

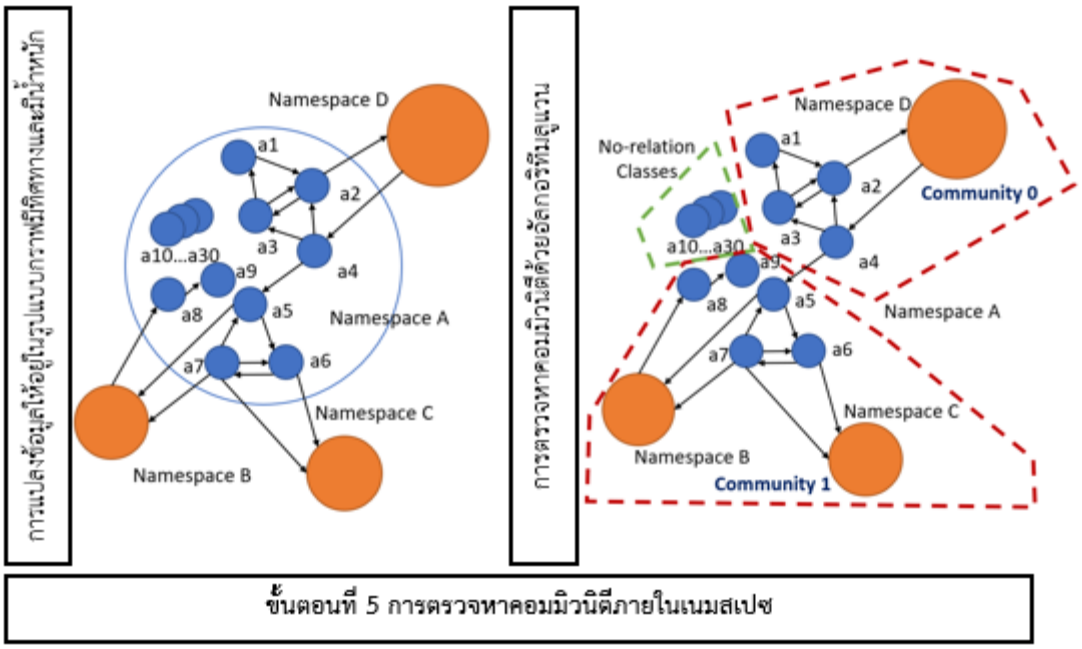
จากตัวอย่างการตรวจหาอันสเตเบิลดีเพนเดนซีในรูปที่ 3.6 โดยแสดงกราฟการพึ่งพาของเนมสเปซต่าง ๆ และตารางค่าตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้องของแต่ละเนมสเปซ ซึ่งประกอบด้วย ค่าเอฟเฟอเรนท์คัปปลิง (Ce) ค่าแอฟเฟอเรนท์คัปปลิง (Ca) และค่าความไม่แน่นอน (I) รวมถึงแสดงเนมสเปซที่เนมสเปซนั้นพึ่งพาอยู่ โดยจากกราฟทิศทางของเส้นเชื่อมแสดงถึงทิศทางการพึ่งพา และสีแดงแสดงถึงการพึ่งพาเนมสเปซที่มีความเสถียรภาพน้อยกว่า การพิจารณาตรวจหาอันสเตเบิลดีเพนเดนซีจะเริ่มจากพิจารณาค่าเอฟเฟอเรนท์คัปปลิงในแต่ละเนมสเปซ จากนั้นพิจารณาว่าเนมสเปซดังกล่าวมีการพึ่งพาเนมสเปซใดบ้างที่มีความเสถียรภาพน้อยกว่า ซึ่งพิจารณาได้จากการเปรียบเทียบค่าความไม่แน่นอน โดยค่าความไม่แน่นอนที่สูงกว่าแสดงถึงความเสถียรภาพที่น้อยกว่า จากตัวอย่างเนมสเปซเอ พึ่งพาเนมสเปซบี, เนมสเปซซี และเนมสเปซดี จากนั้นเปรียบเทียบค่าความไม่แน่นอนของเนมสเปซเอกับเนมสเปซที่เนมสเปซเอพึ่งพา ซึ่งพบว่าเนมสเปซบีมีค่าความไม่แน่นอนที่ 0.66 (I_B=0.66) และเนมสเปซดีมีค่าความไม่แน่นอนที่ 0.66 (I_D=0.66) ซึ่งมีค่าความไม่แน่นอนที่สูงกว่าเนมสเปซเอ ที่มีค่าความไม่แน่นอนที่ 0.6 (I_A=0.6) ดังนั้นเนมสเปซเอจะถูกพิจารณาว่าตรวจพบอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี และทำเช่นนี้ไปจนครบทุกเนมสเปซในโปรเจค

3.4.3. การเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมล

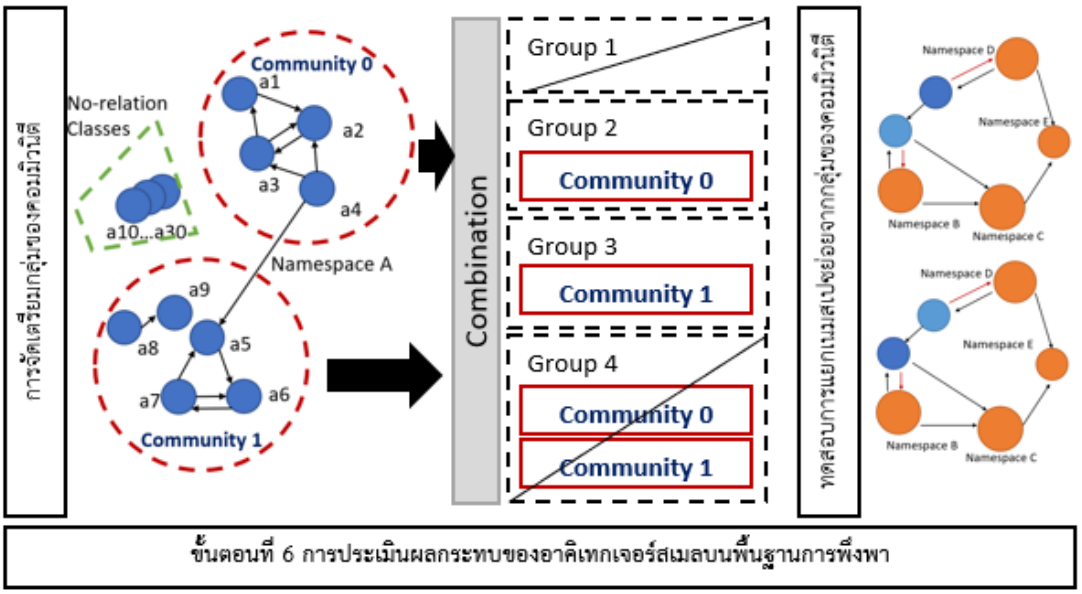
ในขั้นตอนนี้จะเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมลที่ถูกพัฒนาขึ้นในขั้นตอนที่ 3.4.1 และ 3.4.2 โดยเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมลจากเครื่องมือเชิงพาณิชย์ในขั้นตอน 3.2 ที่ถูกเก็บไว้เป็นข้อมูลพื้นฐานในการเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมล เพื่อพิจารณาประสิทธิภาพในส่วนการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนาขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ส่วนถัดมาเป็นส่วนขั้นตอนการรีแฟคทอริงด้วยการแยกแพ็กเกจ ซึ่งประกอบด้วย 2 ขั้นตอนสำคัญ คือ ขั้นตอนที่ 5 การตรวจหาคอมมิวนิตีภายในเนมสเปซ โดยการนำข้อมูลโครงสร้างโปรเจคและความสัมพันธ์ของคลาสมาสร้างกราฟแบบมีทิศทางและมีน้ำหนัก จากนั้นตรวจหาคอมมิวนิตีภายในกราฟโดยใช้อัลกอริทึมลูแวน ซึ่งภาพรวมขั้นตอนการทำงานแสดงดังรูปที่ 3.7 และขั้นตอนที่ 6 การประเมินผลกระทบของอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพา โดยนำคอมมิวนิตีที่ตรวจพบในกราฟมาทำการจัดหมู่ตามเงื่อนไขที่กำหนดเพื่อให้ได้กลุ่มของเนมสเปซ และทำการทดสอบโดยการค้นหาวิธีการรีแฟคทอริงที่ให้ผลลัพธ์ที่ดีที่สุด ซึ่งภาพรวมขั้นตอนการทำงานแสดงดังรูปที่ 3.8



รูปที่ 3.7 ภาพรวมการทำงานขั้นตอนการตรวจหาคอมมิวนิตีภายในเนมสเปซ



รูปที่ 3.8 ภาพรวมการทำงานขั้นตอนการประเมินผลกระทบของอาคิเทกเจอร์สเมล บนพื้นฐานการพึ่งพา

3.5.การตรวจหาคอมมิวนิตีภายในเนมสเปซ

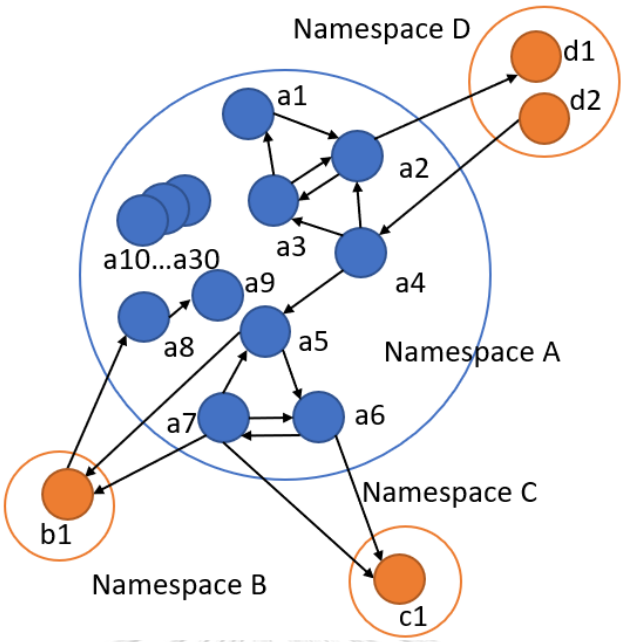
ในขั้นตอนนี้จะพิจารณาเนมสเปซที่ถูกตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจที่มีขนาดใหญ่เกินจากขั้นตอนก่อนหน้า นำมาวิเคราะห์ความสัมพันธ์ของคลาสต่าง ๆ ภายในเนมสเปซ โดยนำข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์ของข้อมูลระดับคลาสของเนมสเปซนั้นมาแปลงให้อยู่ในรูปของกราฟแบบมีทิศทางและมีน้ำหนัก (directed weighted graph) เพื่อที่จะตรวจหาคอมมิวนิตีโดยใช้อัลกอริทึมลูแวน

3.5.1. การแปลงข้อมูลให้อยู่ในรูปแบบกราฟมีทิศทางและมีน้ำหนัก

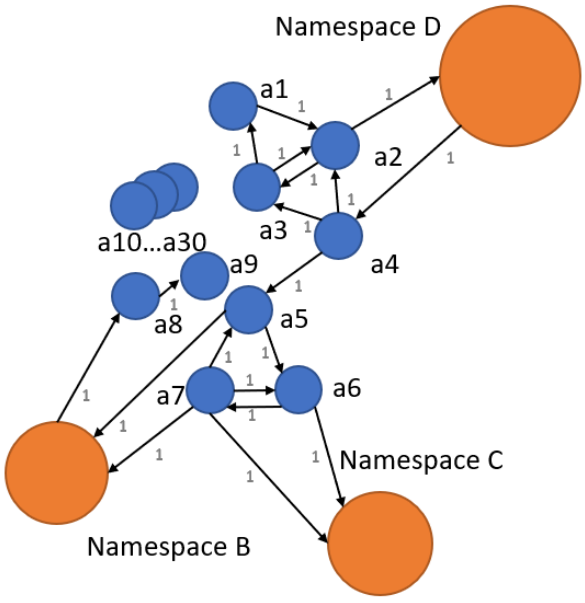
การแปลงข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์ของคลาสต่าง ๆ ภายในเนมสเปซ รวมถึงความสัมพันธ์ระหว่างเนมสเปซให้อยู่ในรูปกราฟแบบมีทิศทางและมีน้ำหนักนั้น จะเริ่มจากการเตรียมการจัดเก็บข้อมูลของกราฟแบบมีทิศทางและมีน้ำหนักในรูปแบบของแอดจาเซนซีเมทริกซ์ (adjacency matrix) โดยรายละเอียดของกราฟ ประกอบด้วย

- โหนด คือ คลาสต่าง ๆ ภายในเนมสเปซ และเนมสเปซรอบข้างที่พึ่งพาและถูกพึ่งพาจากคลาสภายในเนมสเปซ
- เส้นเชื่อม คือ ความสัมพันธ์ระหว่างคลาส และความสัมพันธ์ระหว่างคลาสกับเนมสเปซรอบข้าง โดยทิศทางของเส้นเชื่อมแสดงถึงทิศการการพึ่งพา
- น้ำหนัก คือ ปริมาณความสัมพันธ์ที่เกิดขึ้นระหว่างโหนดต่าง ๆ

ตัวอย่างการแปลงข้อมูลจากความสัมพันธ์ของคลาสของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินให้อยู่ในรูปแบบกราฟที่มีทิศทางและมีน้ำหนัก เริ่มจากนำข้อมูลความสัมพันธ์ของคลาสในเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินมาพิจารณา โดยตัวอย่างนี้คือเนมสเปซเอ (namespace A) เมื่อพิจารณาความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเอและระหว่างคลาสภายในเนมสเปซเอกับคลาสในเนมสเปซรอบข้างที่พึ่งพาหรือถูกพึ่งพาเพื่อนำมาสร้างเป็นกราฟ แสดงตัวอย่างดังรูปที่ 3.9 จากนั้นแปลงความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเอกับคลาสในเนมสเปซรอบข้างที่พึ่งพาหรือถูกพึ่งพา ให้อยู่ในระดับความสัมพันธ์ของคลาสภายในเนมสเปซและเนมสเปซรอบข้างแทน รวมถึงพิจารณาจำนวนของความสัมพันธ์ เพื่อให้ได้กราฟแบบมีทิศทางและมีน้ำหนัก ซึ่งแสดงตัวอย่างดังรูปที่ 3.10 และจัดเก็บข้อมูลกราฟแบบมีทิศทางและมีน้ำหนักที่ได้ในรูปแบบของแอดจาเซนซีเมทริกซ์เพื่อนำไปใช้ในขั้นตอนถัดไป



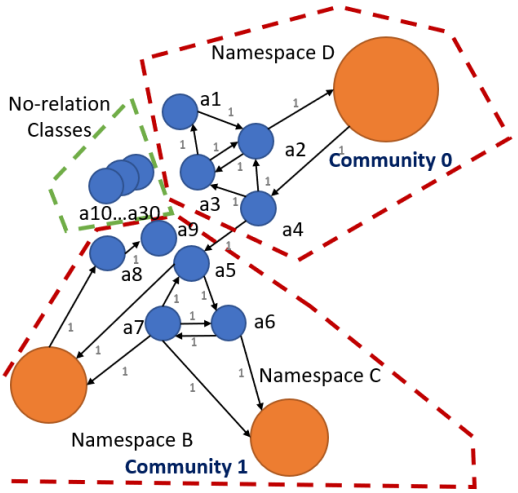
รูปที่ 3.9 กราฟความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเอ และระหว่างคลาสภายในเนมสเปซเอกับคลาสในเนมสเปซรอบข้าง



รูปที่ 3.10 กราฟความสัมพันธ์ระหว่างคลาสภายในเนมสเปซเอ และระหว่างคลาสภายในเนมสเปซเอกับเนมสเปซรอบข้าง

3.5.2. การตรวจหาคอมมิวนิตีด้วยอัลกอริทึมลูแวน

ขั้นตอนนี้จะเป็นการตรวจหาคอมมิวนิตีจากข้อมูลกราฟแบบมีทิศทางและมีน้ำหนักที่ได้จัดเตรียมไว้ โดยผลลัพธ์ของอัลกอริทึมลูแวนจะแสดงกลุ่มของคอมมิวนิตีทั้งหมดที่มีภายในเครือข่าย และแสดงสมาชิกภายในของแต่ละคอมมิวนิตี ดังรูปที่ 3.11 แสดงผลลัพธ์การตรวจหาคอมมิวนิตีจากข้อมูลกราฟแบบมีทิศทางและมีน้ำหนักที่เตรียมไว้ โดยตรวจพบ 2 คอมมิวนิตี ประกอบด้วย คอมมิวนิตีศูนย์ (community 0) ซึ่งมีสมาชิก คือ คลาส a1 a2 a3 a4 และ Namespace D และคอมมิวนิตีหนึ่ง (community 1) ซึ่งมีสมาชิก คือ คลาส a5 a6 a7 a8 a9 Namespace B และ Namespace C โดยสมาชิกภายในแต่ละคอมมิวนิตีจะประกอบด้วยกลุ่มของโหนดที่มีความสัมพันธ์กันแน่นหนา แต่อย่างไรก็ตามบางโหนดที่ไม่มีความสัมพันธ์ใด ๆ เลย จะไม่ถูกจัดลงในกลุ่มคอมมิวนิตีใด ดังนั้นในงานวิจัยนี้จะจัดกลุ่มสำหรับโหนดที่ไม่มีความสัมพันธ์เลยเข้าด้วยกันและเรียกว่ากลุ่มคลาสที่ไม่มีความสัมพันธ์ (no-relation class group) ซึ่งจากตัวอย่างในรูปที่ 3.11 มีสมาชิก คือ คลาส a10 ไปจนถึงคลาส a30



รูปที่ 3.11 ตัวอย่างผลลัพธ์การตรวจหาคอมมิวนิตีจากข้อมูลกราฟแบบมีทิศทางและมีน้ำหนัก

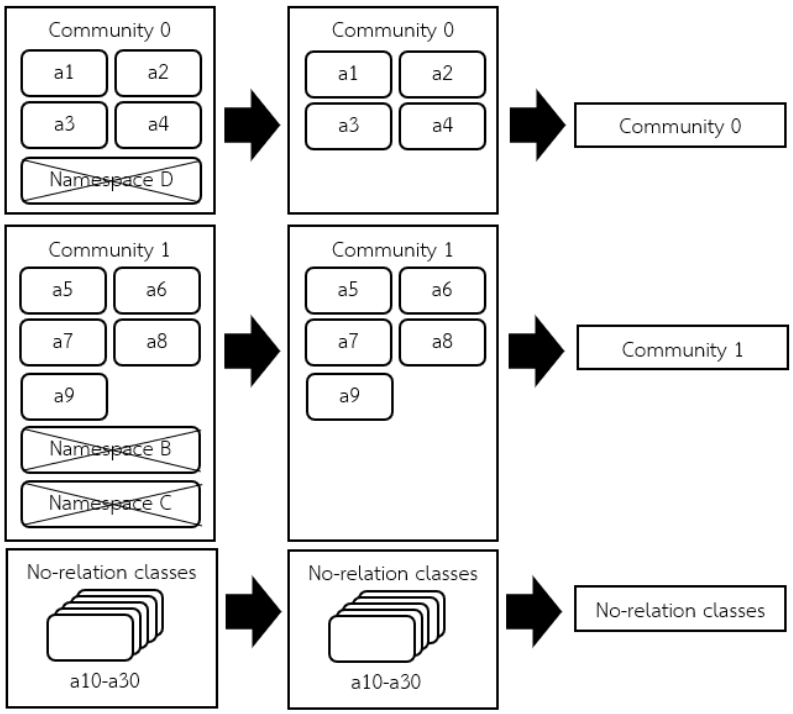
3.6.การประเมินผลกระทบของอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพา

ขั้นตอนนี้จะประเมินผลกระทบของอาคิเทกเจอร์สเมลลนพื้นฐานการพึ่งพาหลังจากการแยกเนมสเปซย่อย โดยที่งานวิจัยนี้จะมุ่งความสนใจไปที่อาคิเทกเจอร์สเมลในรูปแบบอันสเตเบิลดีเพนเดนซีที่เกิดขึ้น เพื่อค้นหาวิธีการรีแฟคทอริงที่เป็นไปได้จากการจัดกลุ่มของคอมมิวนิตีที่ตรวจพบที่สามารถแยกออกไปสร้างเนมสเปซใหม่ได้ โดยไม่ก่อให้เกิดอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีเพิ่มขึ้น

3.6.1. การจัดเตรียมกลุ่มของคอมมิวนิตี

ในขั้นตอนนี้เป็นการจัดเตรียมกลุ่มคอมมิวนิตีเพื่อค้นหาวิธีการรีแฟคทอริงที่เป็นไปได้ทั้งหมดจากกลุ่มของคอมมิวนิตีจากการจัดกลุ่มที่เป็นไปได้โดยการจัดหมู่ (combination) และรีเคอร์ซีฟฟังก์ชัน (recursive function) โดยประกอบด้วย 2 ขั้นตอนย่อยดังนี้

(1) นำเนมสเปซรอบข้างที่เป็นสมาชิกของคอมมิวนิตีออกไปจากคอมมิวนิตีให้เหลือเฉพาะโหนดที่เป็นคลาสภายในเนมสเปซเท่านั้น ดังรูปที่ 3.12



รูปที่ 3.12 ตัวอย่างการนำเนมสเปซรอบข้างออกจากคอมมิวนิตี

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

(2) การจัดเตรียมกลุ่มคอมมิวนิตีที่เป็นไปได้สำหรับการแยกเนมสเปซ จะพิจารณาการแยกเนมสเปซย่อยออกจากเนมสเปซเดิมเป็นตั้งแต่ 1 เนมสเปซ ไปจนถึงจำนวนคอมมิวนิตีที่พบทั้งหมด โดยมีขั้นตอนโดยสรุปดังนี้ และแสดงแผนภาพการทำงานดังรูปที่ 3.13

(2.1) พิจารณาจำนวนการแยกเนมสเปซโดยเริ่มต้นที่ 1 เนมสเปซ

(2.2) นำคอมมิวนิตีที่ตรวจพบทำการจัดหมู่เพื่อหารูปแบบการกลุ่มคอมมิวนิตีที่สามารถแยกไปยังเนมสเปซใหม่

(2.3) กรองกลุ่มคอมมิวนิตี ที่มีเงื่อนไขดังต่อไปนี้ออกจากกลุ่มคอมมิวนิตีที่เป็นไปได้

- กลุ่มที่ไม่มีสมาชิก หรือกลุ่มว่าง (empty set)
- กลุ่มที่มีสมาชิกทั้งหมด (entire set)

- กลุ่มที่มีขนาดเล็กเกินไป คือ มีสมาชิกในกลุ่มน้อยกว่า 3 คลาส
- กลุ่มที่มีขนาดใหญ่เกินไป คือ มีสมาชิกในกลุ่มมากกว่า 30 คลาส

(2.4) พิจารณากลุ่มคอมมิวนิตีที่เป็นไปตามเงื่อนไขทีละกลุ่ม โดยเก็บกลุ่มคอมมิวนิตีไว้สำหรับการแยกเนมสเปซ

(2.5) พิจารณาจำนวนการแยกเนมสเปซ โดยที่

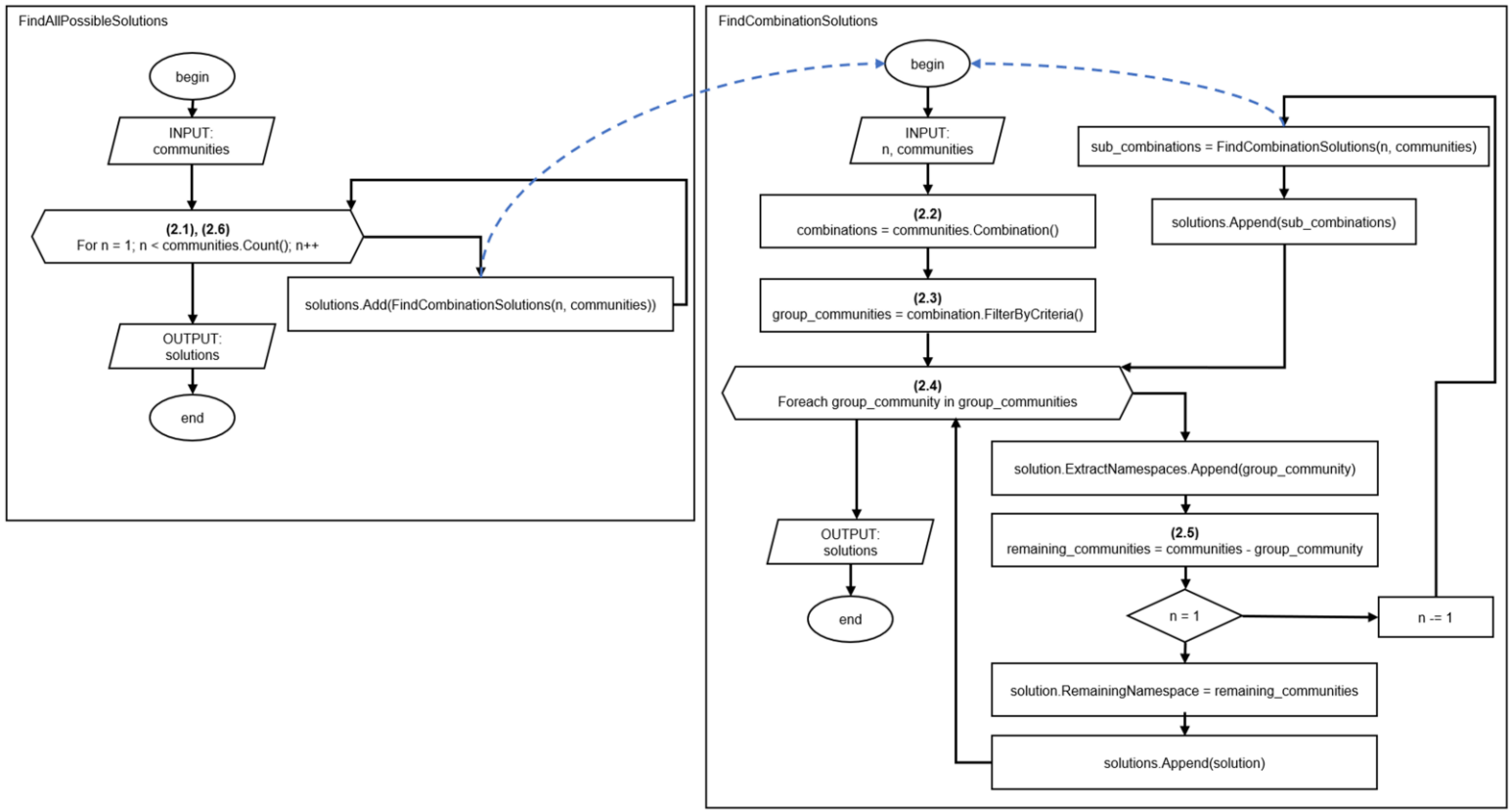
- จำนวนการแยกเนมสเปซเป็น 1 จะพิจารณากลุ่มคอมมิวนิตีที่เหลือและกลุ่มคลาสที่ไม่มีความสัมพันธ์คงไว้ในเนมสเปซเดิม
- จำนวนการแยกเนมสเปซมากกว่า 1 จะพิจารณาการแยกเนมสเปซถัดไป โดยการนำกลุ่มคอมมิวนิตีที่เหลือ และจำนวนการแยกเนมสเปซลดลง 1 กลับไปทำซ้ำในขั้นตอนที่ (2.2) – (2.5) จนจำนวนการแยกเนมสเปซเป็น 1 และหยุดการทำงานซ้ำ

(2.6) วนกลับไปเริ่มต้นที่ (2.1) โดยพิจารณาจำนวนการแยกเนมสเปซเพิ่มขึ้นอีก 1 เนมสเปซ และทำซ้ำจนกว่าจำนวนการแยกเนมสเปซเท่ากับจำนวนคอมมิวนิตีที่ตรวจพบ

จากตัวอย่างในรูปที่ 3.12 พบว่ามีจำนวน 2 คอมมิวนิตีที่ตรวจพบ คือ คอมมิวนิตีศูนย์ และคอมมิวนิตีหนึ่ง เมื่อเข้าสู่ขั้นตอนการจัดกลุ่มคอมมิวนิตีที่เป็นไปได้สำหรับการแยกเนมสเปซ จะมีลำดับขั้นตอนการทำงานดังนี้

- ขั้นตอนที่ (2.1) พิจารณาจำนวนการแยกเนมสเปซที่ 1 เนมสเปซ
- ขั้นตอนที่ (2.2) ทำการจัดหมู่คอมมิวนิตี ได้ผลลัพธ์ประกอบด้วย 4 กลุ่มคอมมิวนิตี ดังนี้ (1) กลุ่มว่าง, (2) กลุ่มที่มีสมาชิกประกอบด้วย คอมมิวนิตีศูนย์, (3) กลุ่มที่มีสมาชิกประกอบด้วย คอมมิวนิตีหนึ่ง, (4) กลุ่มที่มีสมาชิกประกอบด้วย คอมมิวนิตีศูนย์และคอมมิวนิตีหนึ่ง
- ขั้นตอนที่ (2.3) กรองกลุ่มคอมมิวนิตีที่เป็นไปตามเงื่อนไข โดยเหลือกลุ่มคอมมิวนิตีที่ (2) และกลุ่มคอมมิวนิตีที่ (3)
- ขั้นตอนที่ (2.4) พิจารณากลุ่มคอมมิวนิตีที่ (2) สำหรับการแยกเนมสเปซใหม่
- ขั้นตอนที่ (2.5) จำนวนการแยกเนมสเปซเป็น 1 ดังนั้นกลุ่มคอมมิวนิตีที่ (3) และกลุ่มคลาสที่ไม่มีความสัมพันธ์จะคงไว้ในเนมสเปซเดิม
- ขั้นตอนที่ (2.4) พิจารณากลุ่มคอมมิวนิตีที่ (3) สำหรับการแยกเนมสเปซใหม่
- ขั้นตอนที่ (2.5) จำนวนการแยกเนมสเปซเป็น 1 ดังนั้นกลุ่มคอมมิวนิตีที่ (2) และกลุ่มคลาสที่ไม่มีความสัมพันธ์จะคงไว้ในเนมสเปซเดิม
- ขั้นตอนที่ (2.6) วนกลับไปที่ขั้นตอนที่ (2.1) โดยเพิ่มจำนวนการแยกเนมสเปซเป็น 2 ซึ่งเท่ากันจำนวนคอมมิวนิตีจึงหยุดการทำงาน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 3.13 ขั้นตอนการจัดเตรียมกลุ่มคอมมิวนิตีที่เป็นไปได้สำหรับการแยกเนมสเปซ

3.6.2. ทดสอบการแยกเนมสเปซย่อยจากกลุ่มของคอมมิวนิตี

ในขั้นตอนนี้จะทดสอบกลุ่มของคอมมิวนิตีต่าง ๆ ที่เป็นไปได้ ด้วยการปรับปรุงข้อมูลโครงสร้างและความสัมพันธ์ของข้อมูล โดยการสร้างเนมสเปซย่อยขึ้นมาใหม่และย้ายข้อมูลคลาสต่าง ๆ ภายในกลุ่มของคอมมิวนิตีไปยังเนมสเปซย่อยใหม่ หลังจากนั้นจะปรับปรุงค่าความสัมพันธ์ระหว่างคลาสและระหว่างเนมสเปซทั้งหมด ดังตัวอย่างในรูปที่ 3.14 และรูปที่ 3.15 โดยมีขั้นตอนโดยสรุปดังนี้

(1) พิจารณาวิธีการแยกเนมสเปซจากการจัดกลุ่มคอมมิวนิตีจากขั้นตอนก่อนหน้าทีละวิธี

(2) โคลนโครงสร้างข้อมูลโปรเจค

(3) สร้างเนมสเปซที่ต้องการแยก และย้ายคลาสต่าง ๆ จากเนมสเปซเดิมมายังเนมสเปซใหม่

(4) ปรับปรุงความสัมพันธ์ในระดับคลาสทั้งหมด

(5) คำนวณค่าตัวชี้วัดในระดับคลาส และในระดับเนมสเปซทั้งหมดใหม่

(6) ตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซีของโครงสร้างโปรเจคใหม่

(7) พิจารณาค่าความเชื่อมแน่นของเนมสเปซที่แยกมีค่ามากกว่าหรือเท่ากับค่าความเชื่อมแน่นของเนมสเปซเดิม และค่าการคู่ควบของเนมสเปซที่แยกมีค่าน้อยกว่าหรือเท่ากับค่าการคู่ควบของเนมสเปซเดิม ถ้าไม่เป็นไปตามเงื่อนไขวิธีการนี้จะไม่ถูกนำมาพิจารณาต่อ

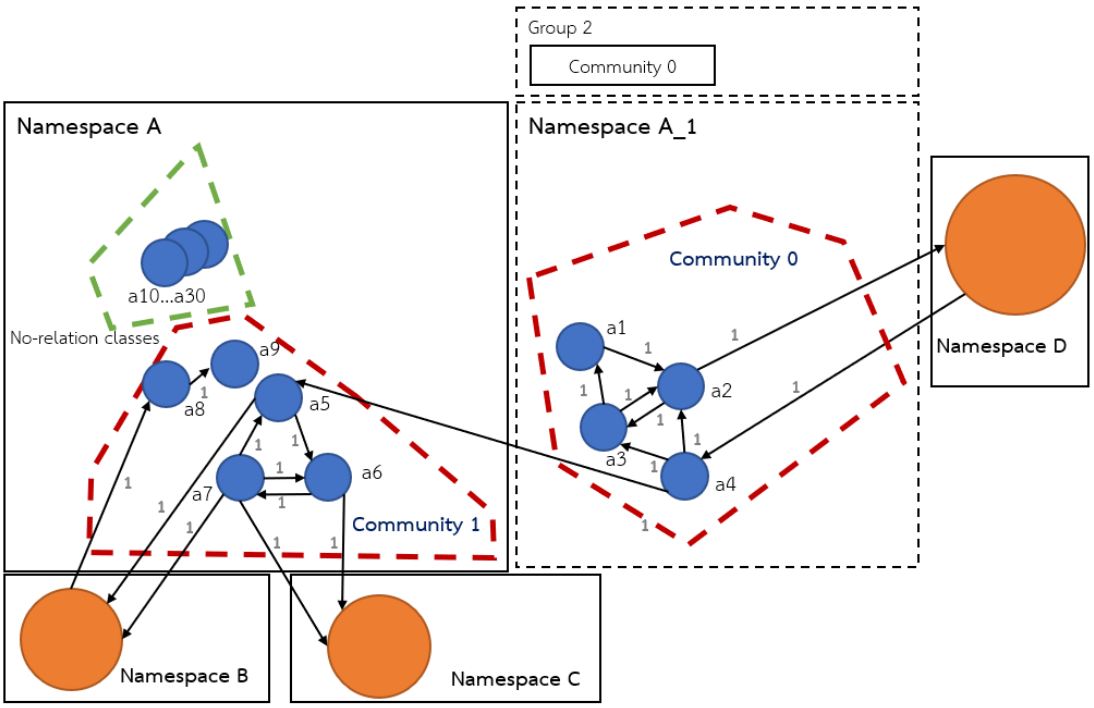
(8) เปรียบเทียบหาวิธีการที่ให้ผลลัพธ์ที่ดีที่สุดในการแยกเนมสเปซ โดยมีลำดับการพิจารณาดังต่อไปนี้

(8.1) จำนวนอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินลดลง และจำนวนอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีไม่เพิ่มขึ้นมาใหม่ โดยเปรียบเทียบกับจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบก่อนกระบวนการรีแฟคทอริง

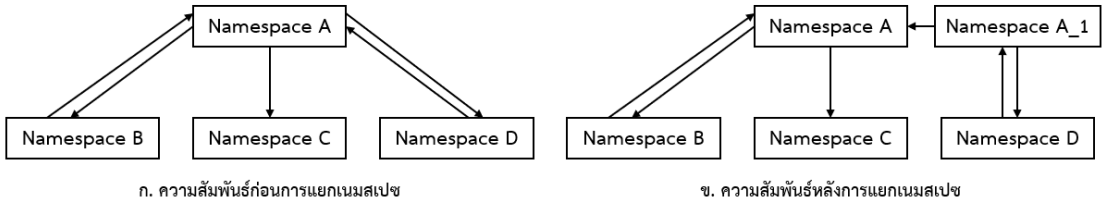
(8.2) กรณีพบมากกว่าหนึ่งวิธีการรีแฟคทอริงที่เป็นไปได้ จะพิจารณาโดยการเปรียบเทียบกลุ่มคอมมิวนิตีที่มีค่าความเชื่อมแน่น (cohesion) ที่สูงที่สุดในเนมสเปซที่แยกออกมา

(8.3) กรณีที่ยังตรวจพบมากกว่าหนึ่งวิธีการรีแฟคทอริงที่เป็นไปได้ จะพิจารณาโดยการเปรียบเทียบกลุ่มคอมมิวนิตีที่มีค่าการคู่ควบ (coupling) ที่มีค่าต่ำที่สุดในเนมสเปซที่แยกออกมา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 3.14 ตัวอย่างการแยกเนมสเปซย่อยจากกลุ่มคอมมิวนิตี

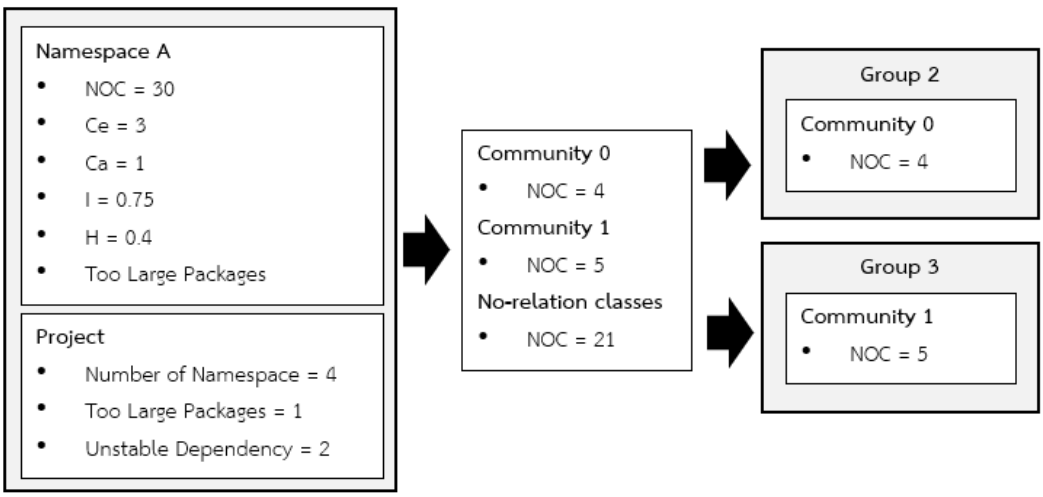


รูปที่ 3.15 ตัวอย่างการเปรียบเทียบโครงสร้างความสัมพันธ์ก่อนและหลังการแยกเนมสเปซ

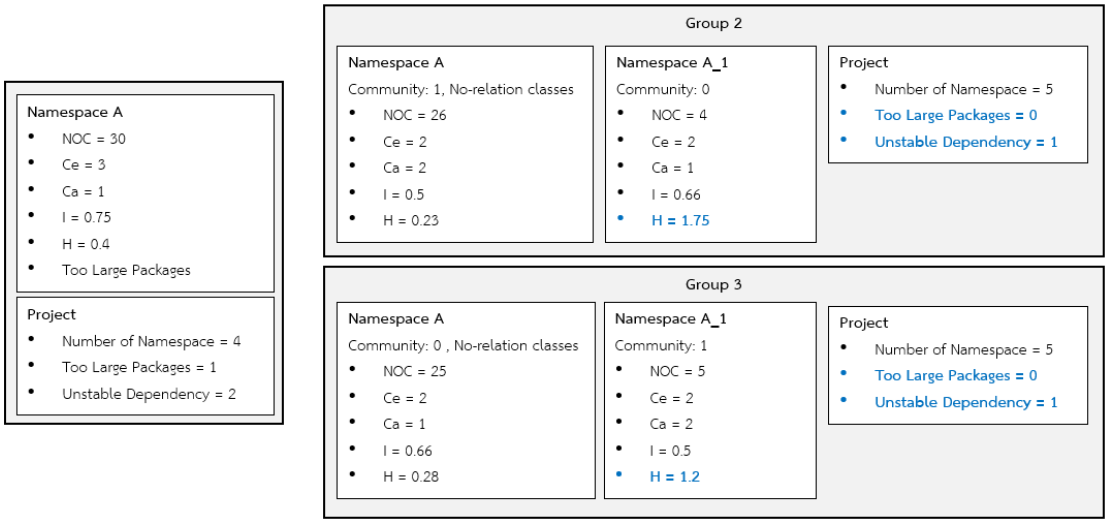
Chulalongkorn University

จากรูปที่ 3.14 แสดงตัวอย่างเป็นการแยกเนมสเปซย่อยจากการนำกลุ่มคอมมิวนิตีที่สอง (group 2) ที่ประกอบด้วยสมาชิกคือคอมมิวนิตีศูนย์มาแยกไปยังเนมสเปซย่อยใหม่ (namespace A_1) ตัวอย่างการเปรียบเทียบโครงสร้างความสัมพันธ์ของเนมสเปซก่อนการแยกเนมสเปซในรูปที่ 3.15 ก. และหลังการแยกเนมสเปซในรูปที่ 3.15 ข. แสดงถึงการเปลี่ยนแปลงความสัมพันธ์ที่เกิดขึ้นในโปรเจค จากการเปลี่ยนแปลงโครงสร้างความสัมพันธ์ทำให้ต้องปรับปรุงค่าความสัมพันธ์ต่าง ๆ ในระดับคลาสและในระดับเนมสเปซ เพื่อทำการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและอันสเตเบิลดีเพนเดนซี รวมถึงการคำนวณค่าตัวชี้วัดความเชื่อมแน่นและการคู่ควบอีกครั้งเพื่อพิจารณาว่าวิธีการรีแฟคทอริงที่เป็นไปได้นั้นไม่ทำให้เกิดอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีเพิ่มขึ้นมาใหม่

กลุ่มของคอมมิวนิตีที่ถูกแยกไปยังเนมสเปซใหม่แล้วไม่ก่อให้เกิดอาคิเทกเจอร์รูปแบบอันสเตเบิลดีเพนเดนซีเพิ่มขึ้นมาใหม่ และมีค่าความเชื่อมแน่นที่สูงขึ้นและมีค่าการควบคู่ที่ลดลง กลุ่มของคอมมิวนิตีนั้นจะถูกเลือกมาพิจารณาเป็นวิธีการรีแฟคทอริงที่เป็นไปได้ ดังตัวอย่างในรูปที่ 3.16 ซึ่งแสดงตัวอย่างของเนมสเปซเอ (namespace A) ที่มีจำนวนคลาส 30 คลาสและถูกตรวจพบว่าอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยในโปรเจคก่อนการรีแฟคทอริงตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินจำนวน 1 เนมสเปซ และอาคิเทกเจอร์รูปแบบอันสเตเบิลดีเพนเดนซีจำนวน 2 เนมสเปซ ซึ่งจากขั้นตอนการตรวจพบคอมมิวนิตี สามารถตรวจพบคอมมิวนิตีภายในเนมสเปซเอจำนวน 2 คอมมิวนิตี และจากการจัดเตรียมกลุ่มคอมมิวนิตีทำให้พบกลุ่มคอมมิวนิตีที่เป็นไปตามเงื่อนไขจำนวน 2 กลุ่มคอมมิวนิตี ซึ่งประกอบด้วย กลุ่มคอมมิวนิตีที่สอง (group 2) และกลุ่มคอมมิวนิตีที่สาม (group 3) เพื่อที่จะพิจารณาหาวิธีการรีแฟคทอริงที่ดีที่สุดที่เป็นไปได้ตามขั้นตอนการพิจารณาที่นำเสนอ จะต้องปรับปรุงค่าความสัมพันธ์และคำนวณค่าตัวชี้วัดต่าง ๆ โดยจากรูปที่ 3.17 แสดงค่าตัวชี้วัดและจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบหลังการแยกเนมสเปซของแต่ละวิธี โดยเมื่อเปรียบเทียบทั้ง 2 วิธี พบว่าการแยกเนมสเปซย่อยด้วยกลุ่มคอมมิวนิตีที่สอง ให้ค่าความเชื่อมแน่นที่สูงที่สุด และเมื่อเปรียบเทียบก่อนการแยกเนมสเปซ พบว่านอกจากค่าความเชื่อมแน่นของเนมสเปซย่อยใหม่จะสูงกว่าเนมสเปซก่อนการแยกแล้ว จำนวนอาคิเทกเจอร์สเมลในโปรเจคก็มีจำนวนลดลงด้วย จากตัวอย่างนี้จะแนะนำวิธีการแยกเนมสเปซโดยการใช้กลุ่มคอมมิวนิตีที่สองเป็นวิธีการรีแฟคทอริงที่ดีที่สุดที่เป็นไปได้



รูปที่ 3.16 ตัวอย่างการพิจารณาวิธีการรีแฟคทอริงที่เป็นไปได้ตามเงื่อนไข



รูปที่ 3.17 ตัวอย่างการเปรียบเทียบวิธีการรีแฟคทอริงที่เป็นไปได้เพื่อหาวิธีที่ดีที่สุด

ส่วนถัดมาเป็นส่วนการแสดงผลด้วยเครื่องมือ ซึ่งประกอบด้วยขั้นตอนที่ 7 การพัฒนาเครื่องมือส่วนแสดงผล

3.7.การพัฒนาเครื่องมือส่วนแสดงผล

ขั้นตอนนี้จะเป็นส่วนการพัฒนาเครื่องมือในส่วนแสดงผลเพื่อเชื่อมต่อกับการทำงานในส่วนก่อนหน้าและอำนวยความสะดวกสำหรับผู้ใช้งาน โดยเครื่องมือที่ถูกพัฒนาขึ้นอยู่ในรูปแบบของวินโดวส์ฟอร์มแอปพลิเคชัน (Windows Forms Application) ที่ทำงานบนระบบปฏิบัติการวินโดวส์ 10 บนพื้นฐานของดอตเน็ตเฟรมเวิร์ก ที่ถูกพัฒนาด้วยภาษาซีชาร์ป โดยจะนำเสนอรายละเอียดการออกแบบและพัฒนาเครื่องมือในบทที่ 4

Chulalongkorn University

ส่วนสุดท้ายเป็นส่วนการทดสอบและประเมินผล ซึ่งเป็นขั้นตอนที่ 8 สำหรับการประเมินผลวิธีการที่นำเสนอและเครื่องมือที่พัฒนาขึ้นในงานวิจัยนี้

3.8.การทดสอบและประเมินผลเครื่องมือ

ขั้นตอนนี้เป็นการทดสอบและประเมินผลเครื่องมือ โดยจะนำวิธีการรีแฟคทอริงที่ได้นำเสนอและถูกพัฒนาขึ้นเป็นเครื่องมือ มาทดลองกับโปรเจคที่ถูกเลือกมาในงานวิจัยนี้ ผลลัพธ์ที่ได้จากเครื่องมือเป็นวิธีการรีแฟคทอริงเพื่อกำจัดอาคิเทกเจอร์สเมลในรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยมีรายละเอียดของกลุ่มของคลาสที่ถูกแนะนำให้แยกไปยังเนมสเปซใหม่ จากนั้นในงานวิจัยนี้จะใช้โปรแกรมไมโครซอฟวิชวลสตูดิโอในการแก้ไขโครงสร้างโปรเจคตามคำแนะนำที่ได้จากเครื่องมือ

หลังจากนั้นจะทำการตรวจหาอาคิเทกเจอร์สเมลอีกครั้งด้วยเครื่องมือที่พัฒนาขึ้น เพื่อเปรียบเทียบปริมาณอาคิเทกเจอร์สเมลก่อนและหลังกระบวนการรีแฟคทอริง ว่ามีปริมาณลดลงและเป็นไปตามที่เครื่องมือนำเสนอผลการตรวจหาอาคิเทกเจอร์สเมลหรือไม่ โดยคำนวณค่าอินเตอร์เซกชันโอเวอร์ยูเนียน ซึ่งจะนำเสนอรายละเอียดการทดสอบความสามารถของเครื่องมือในบทที่ 5

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
การออกแบบและพัฒนาเครื่องมือ

ในบทนี้จะอธิบายในส่วนวิธีการออกแบบและพัฒนาเครื่องมือที่ถูกพัฒนาขึ้นในงานวิจัยนี้เพื่อสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี โดยมีรายละเอียดที่เกี่ยวข้องกับเครื่องมือที่พัฒนา ประกอบด้วย 4 หัวข้อ ได้แก่ สภาพแวดล้อมของการพัฒนาเครื่องมือ ความต้องการของการพัฒนาเครื่องมือเบื้องต้น แผนภาพยูสเคส (use case diagram) แผนภาพกิจกรรม (activity diagram) และแผนภาพคลาส (class diagram)

4.1.สภาพแวดล้อมของการพัฒนาเครื่องมือ

เครื่องมือที่พัฒนาขึ้นในงานวิจัยนี้ถูกพัฒนาขึ้นเพื่อสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี โดยมีรายละเอียดของสภาพแวดล้อมด้านซอฟต์แวร์ ดังตารางที่ 4.1 และรายละเอียดของสภาพแวดล้อมด้านฮาร์ดแวร์ ดังตารางที่ 4.2

ตารางที่ 4.1 รายละเอียดของสภาพแวดล้อมของการพัฒนาเครื่องมือด้านซอฟต์แวร์

สภาพแวดล้อมด้านซอฟต์แวร์	**รายละเอียด**
ภาษาสำหรับการพัฒนาเครื่องมือ	ซีชาร์ป (C#)
เครื่องมือที่ใช้ในการพัฒนา	Microsoft Visual Studio 2017
เฟรมเวิร์ก	.NET Framework 4.7
ไลบรารี่ที่ใช้ในการพัฒนาเครื่องมือ	Microsoft.CodeAnalysis (Roslyn) Microsoft.Msagl EPPlus
ระบบปฏิบัติการ	Windows 10 Professional 64-bit
การควบคุมเวอร์ชัน	VisualSVN Server TortoiseSVN

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4.2 รายละเอียดของสภาพแวดล้อมของการพัฒนาเครื่องมือด้านฮาร์ดแวร์

สภาพแวดล้อมด้านฮาร์ดแวร์	**รายละเอียด**
คอมพิวเตอร์ส่วนบุคคล	หน่วยประมวลผล Intel Core i7-8550U
หน่วยความจำ	16 GB

4.2.ความต้องการของการพัฒนาเครื่องมือเบื้องต้น

จากวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตีที่นำเสนอในบทที่ 3 สามารถนำมากำหนดความต้องการของเครื่องมือ เพื่อให้การพัฒนาเครื่องมือเป็นไปตามที่ได้เสนอวิธีการเอาไว้ โดยจะต้องมีรายละเอียดดังนี้

4.2.1. เครื่องมือจะต้องมีความสามารถในการนำเข้าข้อมูลของซอร์สโค้ดของโปรเจคที่พัฒนาด้วยภาษาซีชาร์ปที่เป็นไปตามข้อกำหนด ในรูปแบบไฟล์นามสกุล .csproj (Visual C# Project File) เพื่อใช้สำหรับการเตรียมข้อมูลขาเข้าเพื่อสร้างโครงสร้างของโปรเจคและความสัมพันธ์ของข้อมูลระดับคลาส

4.2.2. เครื่องมือจะต้องมีความสามารถในการเตรียมโครงสร้างข้อมูลขาเข้าที่กำหนด รวมถึงการสร้างกราฟมีทิศทางและมีน้ำหนัก

4.2.3. เครื่องมือจะต้องมีความสามารถในการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและอันสเตเบิลดีเพนเดนซี รวมถึงการคำนวณค่าตัวชี้วัดความเชื่อมแน่นและการคู่ควบ เพื่อใช้ประกอบการวิเคราะห์ในกระบวนการรีแฟคทอริง

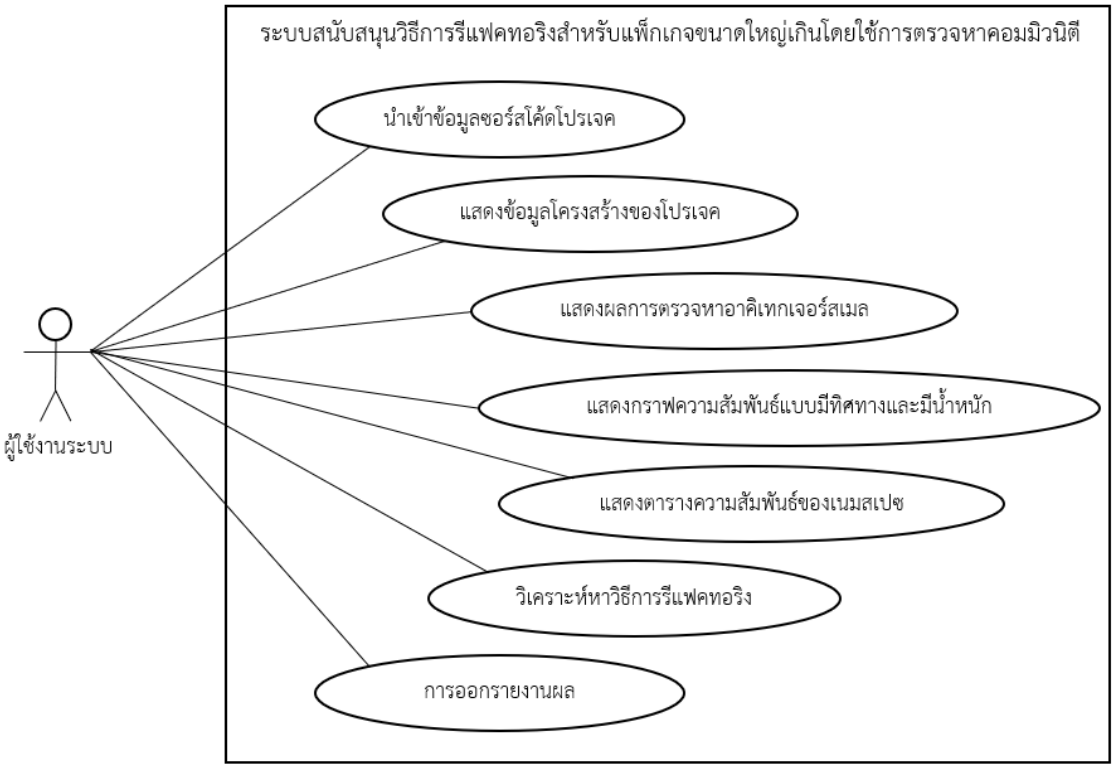
4.2.4. เครื่องมือจะต้องมีความสามารถในการค้นหาวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี เสนอแนะวิธีการรีแฟคทอริงที่ให้ผลลัพธ์ดีที่สุด และแสดงผลวิธีการรีแฟคทอริงในรูปแบบกราฟมีทิศทางและมีน้ำหนัก

4.2.5. เครื่องมือจะต้องสามารถออกรายงานผลการตรวจหาอาคิเทกเจอร์สเมลและวิธีการรีแฟคทอริงที่เสนอแนะในรูปแบบของไฟล์เอ็กซ์เซล (Excel file)

4.3.แผนภาพยูสเคส

แผนภาพยูสเคสของเครื่องมือที่พัฒนาในงานวิจัยนี้จะแสดงความต้องการของระบบสำหรับการใช้งานในมุมมองของผู้ใช้งาน เพื่ออธิบายฟังก์ชันการทำงานพื้นฐานของเครื่องมือ ดังรูปที่ 4.1 โดยแบ่งคำอธิบายการทำงานยูสเคสออกเป็น 4 กลุ่มการทำงานหลัก ได้แก่ ส่วนการนำเข้าข้อมูล ส่วนการแสดงข้อมูลก่อนกระบวนการรีแฟคทอริง ส่วนการวิเคราะห์หาวิธีการรีแฟคทอริง และส่วนการออกรายงานผล โดยมีรายละเอียดดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 4.1 แผนภาพยูสเคสของเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี

4.3.1. ส่วนการนำเข้าข้อมูล

ส่วนการนำเข้าข้อมูล คือ ส่วนการนำเข้าข้อมูลซอร์สโค้ดโปรเจค ซึ่งระบบจะให้ผู้ใช้งานเลือกไฟล์โปรเจคที่พัฒนาด้วยภาษาซีชาร์ป ในรูปแบบไฟล์นามสกุล .csproj นอกจากนี้ระบบจะจดจำการนำเข้าข้อมูลซอร์สโค้ดโปรเจคล่าสุดจำนวน 5 โปรเจคเพื่อให้ผู้ใช้งานสามารถนำเข้าโปรเจคที่เคยวิเคราะห์ล่าสุดได้รวดเร็วขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

4.3.2. ส่วนการแสดงข้อมูลก่อนกระบวนการรีแฟคทอริง

ส่วนของการแสดงข้อมูลก่อนกระบวนการรีแฟคทอริง คือ ส่วนของการประผลหลังจากนำเข้าข้อมูลโปรเจค โดยระบบจะทำการวิเคราะห์ข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์ของคลาสต่าง ๆ ในโปรเจค เพื่อแสดงรายละเอียดของโปรเจค รวมถึงการตรวจหาอาคิเทกเจอร์สเมลที่ตรวจพบในโปรเจคปัจจุบันก่อนกระบวนการรีแฟคทอริง โดยมีส่วนการแสดงผลดังนี้

(1) ส่วนแสดงข้อมูลโครงสร้างของโปรเจค

ในส่วนนี้จะนำข้อมูลโครงสร้างของโปรเจคที่ทำการวิเคราะห์มาแสดงรายละเอียด เพื่อให้ผู้ใช้งานได้ทราบถึงจำนวนเนมสเปซที่มีในโปรเจค และจำนวนคลาสทั้งหมดในโปรเจค

นอกจากนี้ยังมีการแสดงรายละเอียดของค่าตัวชี้วัดซอฟต์แวร์ในระดับเนมสเปซ ซึ่งประกอบด้วย จำนวนคลาส (NOC) ในเนมสเปซ, จำนวนบรรทัด (LOC), เอฟเฟอเรนท์คัปปลิง (Ce), แอฟเฟอเรนท์คัปปลิง (Ca), ความไม่แน่นอน (I) และความเชื่อมแน่นเชิงสัมพันธ์ (H) และค่าตัวชี้วัดซอฟต์แวร์ในระดับคลาส ซึ่งประกอบด้วย จำนวนบรรทัด, แฟนอิน (Fan-in) และแฟนเอาท์ (Fan-out)

(2) ส่วนแสดงผลการตรวจหาอาคิเทกเจอร์สเมล

ในส่วนนี้จะนำข้อมูลโครงสร้างของโปรเจคและข้อมูลความสัมพันธ์ของคลาสมาวิเคราะห์ตรวจหาอาคิเทกเจอร์สเมลใน 2 รูปแบบ คือ รูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี โดยการแสดงผลจะแสดงจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบในแต่ละประเภท รวมถึงแสดงรายละเอียดของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมล

- เนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน จะแสดงค่าจำนวนคลาสและจำนวนบรรทัดที่เกินข้อกำหนดและบ่งชี้ว่าเป็นอาคิเทกเจอร์สเมลรูปแบบนี้
- เนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี จะแสดงค่าความไม่แน่นอน เอฟเฟอเรนท์คัปปลิง และแอฟเฟอเรนท์คัปปลิง รวมถึงรายการเนมสเปซที่พึ่งพาและมีค่าความไม่แน่นอนสูงกว่าที่บ่งชี้ว่าเป็นอาคิเทกเจอร์สเมลรูปแบบนี้

(3) ส่วนแสดงการฟความสัมพันธ์แบบมีทิศทางและมีน้ำหนัก

ในส่วนนี้จะนำข้อมูลโครงสร้างของโปรเจคและข้อมูลความสัมพันธ์ของคลาสมาสร้างเป็นกราฟความสัมพันธ์ในระดับโปรเจค และในระดับเนมสเปซ

- กราฟความสัมพันธ์ในระดับโปรเจค จะแสดงการพึ่งพาของเนมสเปซต่าง ๆ ภายในโปรเจค โดยจะเน้นกรอบสีแดงที่โหนดที่แสดงถึงเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลในรูปแบบแพ็กเกจขนาดใหญ่เกิน และจะเน้นเส้นเชื้อมความสัมพันธ์เป็นสีแดงเพื่อแสดงว่าเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินกำลังพึ่งพาเนมสเปซที่มีความเสถียรน้อยกว่า
- กราฟความสัมพันธ์ในระดับเนมสเปซ จะแสดงความสัมพันธ์ระหว่างคลาสภายในเนมสเปซนั้น และจะแสดงโหนดของเนมสเปซรอบข้างที่มีความสัมพันธ์กับคลาสภายในเนมสเปซ

(4) ส่วนการแสดงตารางความสัมพันธ์ของเนมสเปซ

ในส่วนนี้จะแสดงความสัมพันธ์ของเนมสเปซต่าง ๆ ภายในโปรเจคในรูปแบบแอดจาเซนซีเมทริกซ์ เพื่อแสดงให้เห็นความสัมพันธ์ระหว่างเนมสเปซต่าง ๆ รวมถึงความหนาแน่นของความสัมพันธ์ระหว่างเนมสเปซต่าง ๆ ภายในโปรเจค

4.3.3. ส่วนการวิเคราะห์หาวิธีการรีแฟคทอริง

ส่วนการวิเคราะห์หาวิธีการรีแฟคทอริง คือ ส่วนการค้นหาวิธีการรีแฟคทอริงที่เหมาะสมสำหรับเนมสเปซที่ถูกตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยการตรวจหาคอมมิวนิตี เพื่อเสนอแนวทางการแยกแพ็กเกจที่เหมาะสมในกระบวนการรีแฟคทอริง โดยมีส่วนการแสดงผลย่อยดังต่อไปนี้

(1) ส่วนการแสดงผลการตรวจหาคอมมิวนิตี

ในส่วนนี้จะแสดงกราฟความสัมพันธ์ในระดับเนมสเปซ โดยการเปลี่ยนสีของคลาสต่าง ๆ ไปตามคอมมิวนิตีที่ตรวจพบ และสามารถรวมกลุ่มของคลาสและแทนที่การแสดงผลด้วยกลุ่มคอมมิวนิตีที่ตรวจพบ

(2) ส่วนการแสดงวิธีการรีแฟคทอริง

ในส่วนนี้จะเป็นการนำกราฟความสัมพันธ์ในระดับเนมสเปซที่ถูกแทนที่ด้วยกลุ่มคอมมิวนิตี มาแสดงผลการค้าหาวิธีการรีแฟคทอริงที่ให้ผลลัพธ์ดีที่สุด โดยการแสดงผลจะแบ่งแยกกลุ่มคอมมิวนิตีตามสีที่แสดง ในกรณีที่ไม่สามารถค้นหาวิธีการรีแฟคทอริงที่เป็นไปตามข้อกำหนดได้ ในส่วนนี้จะไม่แสดงผลต่อผู้ใช้งาน

4.3.4. ส่วนการออกรายงานผล

ส่วนการออกรายงานผล คือ ส่วนสร้างรายงานผลในรูปแบบไฟล์เอ็กซ์เซล โดยในรายงานจะประกอบด้วยข้อมูลต่าง ๆ ดังนี้

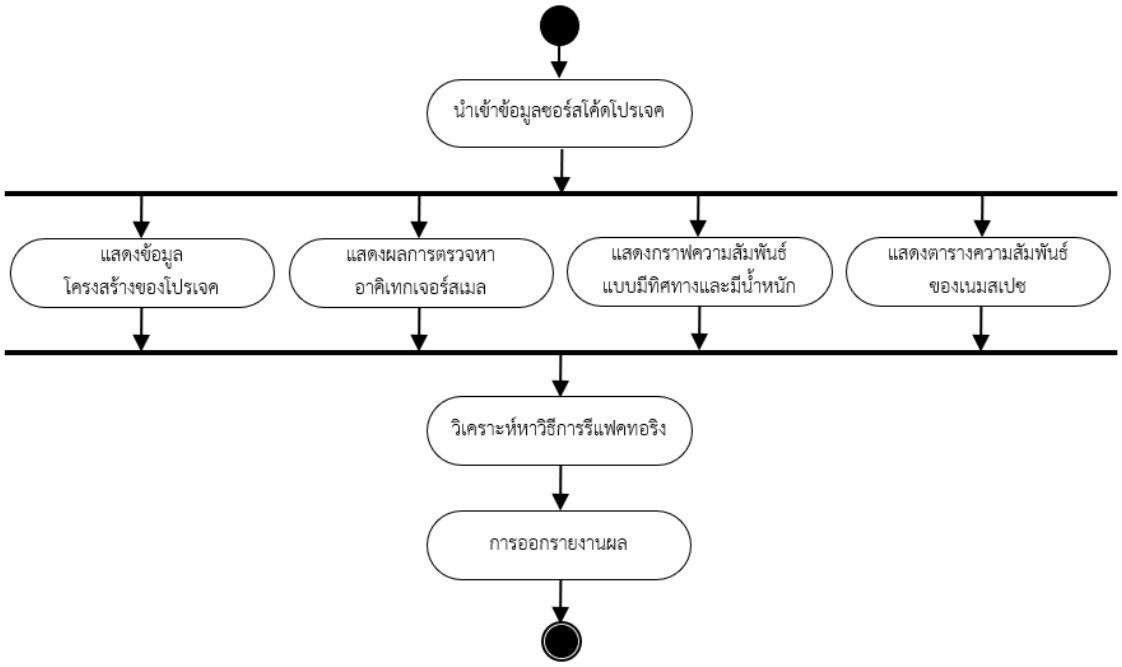
(1) รายงานแสดงข้อมูลโครงสร้างของโปรเจค

(2) รายงานแสดงผลการตรวจหาอาคิเทกเจอร์สเมล

(3) ส่วนแสดงวิธีการรีแฟคทอริงสำหรับเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยนำเสนอกลุ่มของคลาสในการแยกแพ็กเกจ และแสดงผลการเปรียบเทียบค่าตัวชี้วัดต่าง ๆ ก่อนและหลังกระบวนการรีแฟคทอริงที่แนะนำ

4.4.แผนภาพกิจกรรม

แผนภาพกิจกรรมของเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกิน โดยใช้การตรวจหาคอมมิวนิตี แสดงถึงลำดับขั้นตอนของกิจกรรมต่าง ๆ รวมถึงการทำงานของเครื่องมือ โดยมีรายละเอียดดังรูปที่ 4.2

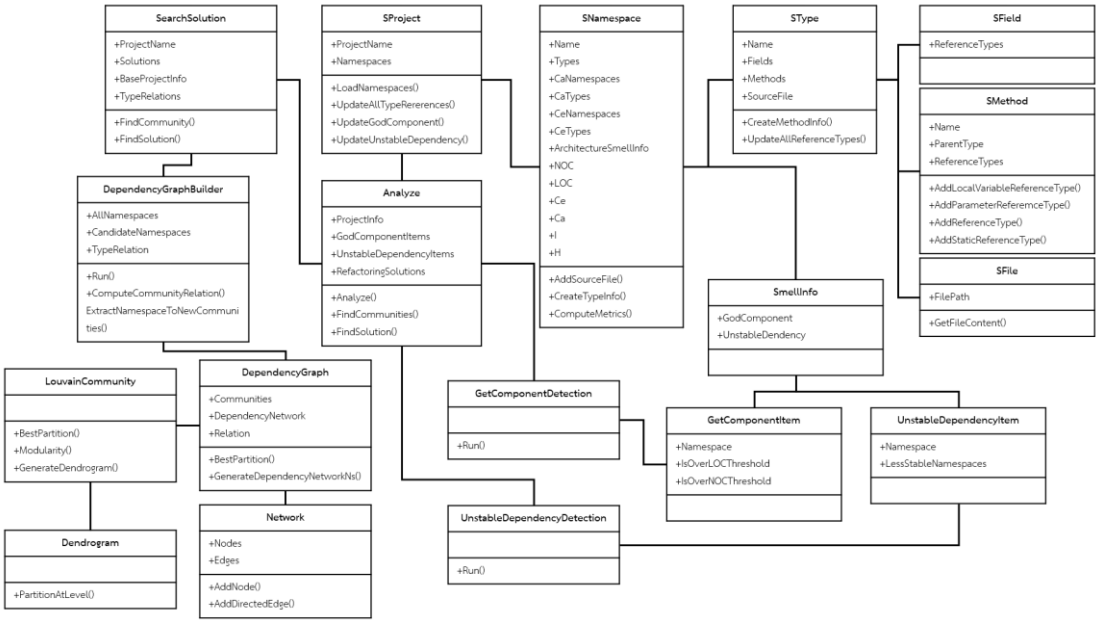


รูปที่ 4.2 แผนภาพกิจกรรมของเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกิน โดยใช้การตรวจหาคอมมิวนิตี

โดยกิจกรรมเริ่มต้นคือการนำข่าข้อมูลซอร์สโค้ดของโปรเจคที่ต้องการวิเคราะห์หาวิธีการรีแฟคทอริง จากนั้นระบบจะทำการวิเคราะห์ข้อมูลโครงสร้างของโปรเจค ความสัมพันธ์ของคลาสต่าง ๆ และตรวจหาอาคิเทกเจอร์สเมลในโปรเจค โดยแสดงข้อมูลต่าง ๆ ให้ผู้ใช้งานประกอบด้วย แสดงข้อมูลโครงสร้างโปรเจค แสดงผลการตรวจหาอาคิเทกเจอร์สเมล แสดงกราฟความสัมพันธ์แบบมีทิศทางและมีน้ำหนัก และแสดงตารางความสัมพันธ์ของเนมสเปซ จากนั้นเป็นกิจกรรมการวิเคราะห์หาวิธีการรีแฟคทอริงสำหรับเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน และส่วนการทำงานของกิจกรรมสุดท้ายคือการออกแบบรายงานผล

4.5.แผนภาพคลาส

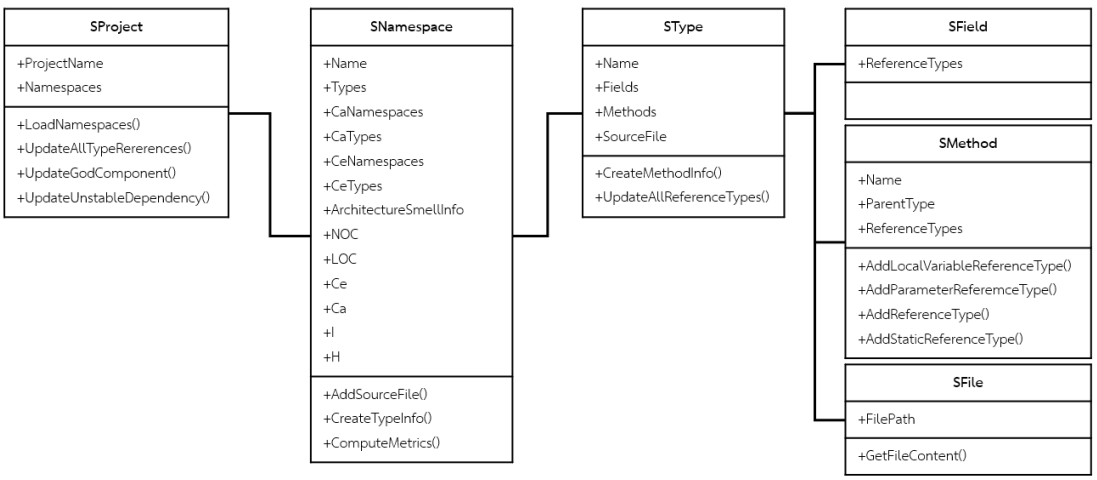
แผนภาพคลาสแสดงภาพรวมการทำงานของเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี แสดงดังรูปที่ 4.3 โดยสามารถจัดกลุ่มคลาสตามประเภทการทำงานต่าง ๆ ประกอบด้วย แผนภาพคลาสของการจัดเก็บข้อมูลโครงสร้างโปรเจค แผนภาพคลาสของการตรวจหาอาคิเทกเจอร์สเมล และแผนภาพคลาสของวิธีการรีแฟคทอริงโดยใช้การตรวจหาคอมมิวนิตี



รูปที่ 4.3 แผนภาพคลาสแสดงภาพรวมการทำงานของเครื่องมือ

4.5.1. แผนภาพคลาสของการจัดเก็บข้อมูลโครงสร้างโปรเจค

เป็นแผนภาพคลาสของการจัดเก็บข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์ระดับคลาส หลังจากผู้ใช้งานนำเข้าข้อมูลซอร์สโค้ดของโปรเจค เครื่องมือที่พัฒนาจะทำการแปลงข้อมูลซอร์สโค้ดจากรูปแบบไฟล์ต่าง ๆ ให้อยู่ในรูปโครงสร้างข้อมูลต่าง ๆ ดังรูปที่ 4.4 โดยมีรายละเอียดดังนี้

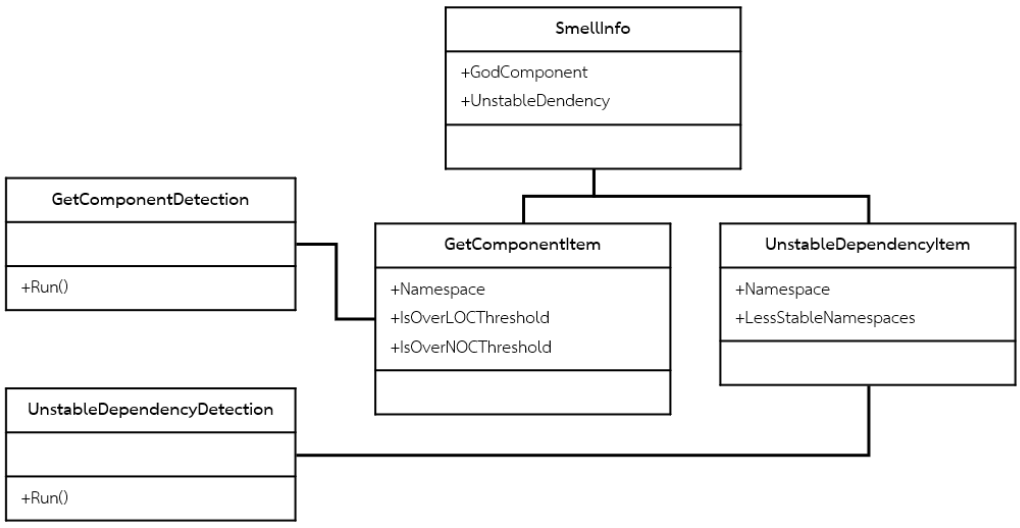


รูปที่ 4.4 แผนภาพคลาสของการจัดเก็บข้อมูลโครงสร้างโปรเจค

- คลาส SProject จัดเก็บข้อมูลในระดับโปรเจค ซึ่งประกอบด้วยข้อมูลเนมสเปซต่าง ๆ
- คลาส SNamespace จัดเก็บข้อมูลระดับเนมสเปซ ซึ่งประกอบด้วยข้อมูลคลาสต่าง ๆ รวมถึงค่าตัวชี้วัดซอฟต์แวร์ในระดับเนมสเปซ และผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี
- คลาส SType จัดเก็บข้อมูลระดับคลาส ซึ่งประกอบด้วยโครงสร้างคลาส อินเตอร์เฟซ สตรัค และอีนัม ซึ่งประกอบด้วยข้อมูลฟิลด์ต่าง ๆ และเมธอดต่าง ๆ ภายในคลาส รวมถึงค่าตัวชี้วัดซอฟต์แวร์ในระดับคลาส นอกจากนี้มีการเก็บข้อมูลความสัมพันธ์ระหว่างคลาส และข้อมูลไฟล์ที่อยู่ของคลาส
- คลาส SField จัดเก็บข้อมูลระดับฟิลด์ ซึ่งเก็บข้อมูลว่าฟิลด์นั้นมีการอ้างอิงถึงคลาสใดในโปรเจค
- คลาส SMethod จัดเก็บข้อมูลในระดับเมธอด ซึ่งเก็บข้อมูลว่าเมธอดนั้นมีการอ้างอิงถึงคลาสใดในโปรเจค
- คลาส SFile จัดเก็บข้อมูลในระดับไฟล์

4.5.2. แผนภาพคลาสของการตรวจหาอาคิเทกเจอร์สเมล

เป็นแผนภาพคลาสของการตรวจหาอาคิเทกเจอร์สเมลและการจัดเก็บผลการตรวจหาอาคิเทกเจอร์สเมลในระดับเนมสเปซ หลังจากได้ข้อมูลโครงสร้างโปรจค เครื่องมือจะทำการตรวจหาอาคิเทกเจอร์สเมลและจัดเก็บผลการตรวจหาอาคิเทกเจอร์สเมลลงในข้อมูลเนมสเปซ ดังรูปที่ 4.5 โดยมีรายละเอียดดังนี้



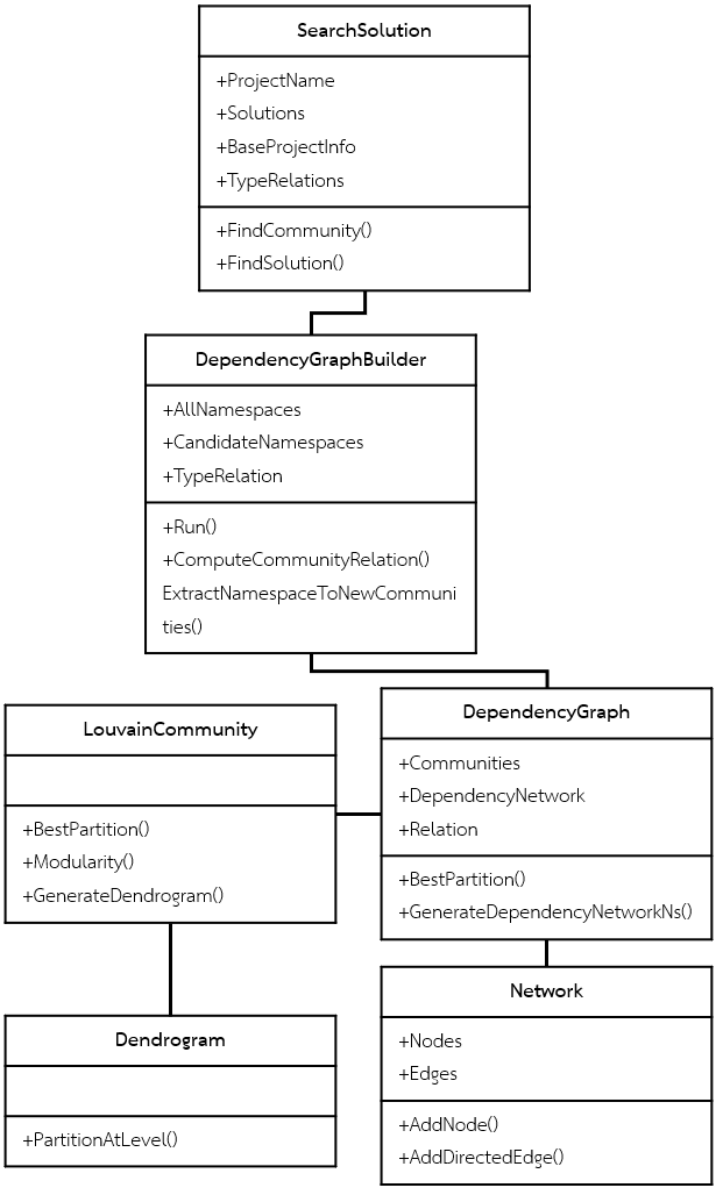
รูปที่ 4.5 แผนภาพคลาสของการตรวจหาอาคิเทกเจอร์สเมล

- คลาส GodComponentDetection เป็นคลาสสำหรับการตรวจหาอาคิเทกเจอร์สเมลในรูปแบบแพ็กเกจขนาดใหญ่เกิน
- คลาส UnstableDependencyDetection เป็นคลาสสำหรับการตรวจหาอาคิเทกเจอร์สเมลในรูปแบบอันสเตเบิลดีเพนเดนซี
- คลาส SmellInfo เป็นคลาสที่เก็บข้อมูลการตรวจหาอาคิเทกเจอร์สเมลทั้ง 2 รูปแบบ ซึ่งเป็นข้อมูลที่ถูกจัดเก็บในระดับเนมสเปซ ในคลาส SNamespace
- คลาส GodComponentItem เป็นคลาสที่สืบทอดจากคลาส ArchitectureSmellItemBase ไว้สำหรับเก็บข้อมูลของเนมสเปซในโปรเจคที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน และค่าตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง
- คลาส UnstableDependencyItem เป็นคลาสที่สืบทอดจากคลาส ArchitectureSmellItemBase ไว้สำหรับเก็บข้อมูลของเนมสเปซในโปรเจคที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี และเนมสเปซที่พึ่งพาที่มีความเสถียรภาพน้อยกว่า

4.5.3. แผนภาพคลาสของวิธีการรีแฟคทอริงโดยใช้การตรวจหาคอมมิวนิตี

เป็นแผนภาพคลาสของวิธีการรีแฟคทอริงโดยการตรวจหาคอมมิวนิตี โดยประกอบด้วย 3 ส่วนหลักดังรูปที่ 4.6 และมีรายละเอียดดังนี้

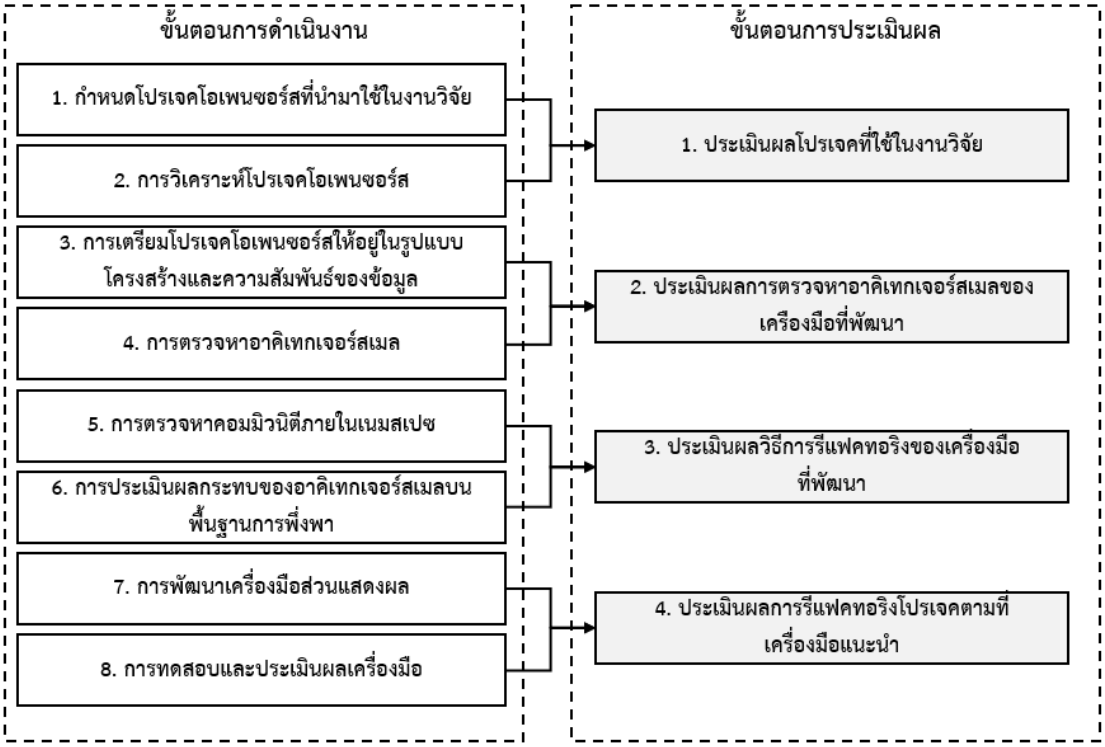
- ส่วนการค้นหาวิธีการรีแฟคทอริงที่เป็นไปได้ที่ดีที่สุด คือ คลาส SearchSolution เป็นคลาสการทำงานหลักสำหรับการค้าหาวิธีการรีแฟคทอริงโดยเริ่มจากการค้นหาคอมมิวนิตีในเนมสเปซ หากลุ่มคอมมิวนิตีที่เป็นไปตามเงื่อนไข แยกเนมสเปซย่อยและเปรียบเทียบค่าตัวชี้วัดเพื่อหาวิธีการรีแฟคทอริงที่ดีที่สุด
- ส่วนการสร้างกราฟมีทิศทางและน้ำหนักจากความสัมพันธ์ระหว่างเนมสเปซ ความสัมพันธ์ระหว่างคลาส และความสัมพันธ์ระหว่างคอมมิวนิตี ประกอบด้วย 2 คลาส คือ คลาส DependencyGraphBuilder สำหรับการสร้างกราฟและคลาส DependencyGraph สำหรับการตรวจหาและเก็บข้อมูลคอมมิวนิตีในกราฟ
- ส่วนการตรวจหาคอมมิวนิตีด้วยอัลกอริทึมลูแวน ประกอบด้วย 3 คลาส คือ คลาส LouvainCommunity เป็น คลาสการทำงานหลักของอัลกอริทึมลูแวน คลาส Dendrogram เป็นคลาสเก็บข้อมูลเดนโดรแกรมของกราฟ และคลาส Network เป็นคลาสการสร้างเครือข่ายสำหรับอัลกอริทึมลูแวน



รูปที่ 4.6 แผนภาพคลาสของวิธีการรีแฟคทอริงโดยใช้การตรวจหาคอมมิวนิตี

บทที่ 5
การทดสอบและประเมินผลความสามารถของเครื่องมือ

ผู้วิจัยได้นำเสนอวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี โดยนำมาพัฒนาเป็นเครื่องมือ ดังนั้นเพื่อที่จะทดสอบความสามารถของเครื่องมือ ผู้วิจัยได้แบ่งการทดสอบและประเมินผลในแต่ละขั้นตอนของวิธีการที่ได้นำเสนอ โดยการประเมินผลความสามารถของเครื่องมือจะประกอบด้วย 4 ขั้นตอน ประกอบด้วย การประเมินผลโปรเจคที่ใช้ในงานวิจัย การประเมินผลการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนา การประเมินผลการรีแฟคทอริงของเครื่องมือที่พัฒนา และการประเมินผลการรีแฟคทอริงโปรเจคตามที่เครื่องมือแนะนำ โดยแสดงภาพรวมขั้นตอนการประเมินผลดังรูปที่ 5.1 และมีรายละเอียดในแต่ละขั้นตอนดังนี้



รูปที่ 5.1 ภาพรวมขั้นตอนการประเมินผล

5.1.ประเมินผลโปรเจคที่ใช้ในงานวิจัย

ในขั้นตอนแรกเป็นการประเมินผลโปรเจคที่เลือกมาใช้ในงานวิจัย โดยขั้นตอนนี้ประกอบด้วยการศึกษารายละเอียดโปรเจคที่ใช้ในงานวิจัย ว่าเป็นไปตามเงื่อนไขข้อกำหนดของงานวิจัย และการตรวจหาอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์เพื่อใช้เป็นข้อมูลพื้นฐาน

5.1.1. รายละเอียดโปรเจคโปรเจคที่ใช้ในงานวิจัย

ในงานวิจัยนี้ได้กำหนดเงื่อนไขในการเลือกโปรเจคโอเพนซอร์สในขั้นตอนการดำเนินงาน โดยโปรเจคที่ใช้ในงานวิจัยนี้เป็นโปรเจคที่พัฒนาด้วยภาษาซีชาร์ปบนพื้นฐานดอตเน็ตเฟรมเวิร์กจากเว็บไซต์กิตฮับจำนวน 3 โปรเจคที่มีขนาดแตกต่างกันประกอบด้วย

(1) โปรเจค Hyperion

โปรเจค Hyperion เป็นระบบโพลีมอร์ฟิกซีเรียลไลเซอร์ (polymorphic serializer) ที่มีประสิทธิภาพสูง สำหรับโปรแกรมที่พัฒนาบนพื้นฐานดอตเน็ตเฟรมเวิร์ก โดยถูกออกแบบมาเพื่อการส่งข้อมูลที่มีความปลอดภัยในระบบกระจายข้อมูล เช่น เซอร์วิสบัส (service bus) โดยมีรายละเอียดโปรเจคเพิ่มเติมดัง ตารางที่ 5.1

ตารางที่ 5.1 รายละเอียดโปรเจค Hyperion

ชื่อโปรเจค	Hyperion
ดอตเน็ตเฟรมเวิร์กเวอร์ชัน	4.5
จำนวนเนมสเปซ	6
ขนาดโปรเจค	โปรเจคขนาดเล็ก
เว็บไซต์	https://github.com/akkadotnet/Hyperion
วันเข้าถึงข้อมูล	1 ตุลาคม 2561

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

(2) โปรเจค Hangfire.Core

โปรเจค Hangfire.Core เป็นระบบที่ช่วยสนับสนุนการทำงานของเอเอสพีดอตเน็ตแอพพลิเคชัน (ASP.NET application) เพื่อให้ใช้งานได้ง่าย สำหรับการทำงานเบื้องหลัง (background process) ที่ใช้ระยะเวลานาน และระยะเวลาสั้น โดยไม่ต้องใช้วินโดว์เซอร์วิส (Windows Service) และ Task Scheduler โดยมีรายละเอียดโปรเจคเพิ่มเติมดัง ตารางที่ 5.2

ตารางที่ 5.2 รายละเอียดโปรเจค Hangfire.Core

ชื่อโปรเจค	Hangfire.Core
ดอตเน็ตเฟรมเวิร์กเวอร์ชัน	4.5
จำนวนเนมสเปซ	16
ขนาดโปรเจค	โปรเจคขนาดกลาง
เว็บไซต์	https://github.com/HangfireIO/Hangfire
วันเข้าถึงข้อมูล	30 กันยายน 2561

(3) โปรเจค Nancy

โปรเจค Nancy เป็นเฟรมเวิร์กสำหรับการพัฒนาเซอร์วิสพื้นฐานบนโปรโตคอลเอชทีทีพี (HTTP) ที่มีน้ำหนักเบาและขั้นตอนน้อยบนดอตเน็ตเฟรมเวิร์ก โดยการเลือกเส้นทางที่ดีที่สุด และรองรับการร้องขอในหลายรูปแบบ ประกอบด้วย GET, HEAD, PUT, POST, DELETE, OPTIONS และ PATCH โดยมีรายละเอียดโปรเจคเพิ่มเติมดัง ตารางที่ 5.3

ตารางที่ 5.3 รายละเอียดโปรเจค Nancy

ชื่อโปรเจค	Nancy
ดอตเน็ตเฟรมเวิร์กเวอร์ชัน	4.5.2
จำนวนเนมสเปซ	37
ขนาดโปรเจค	โปรเจคขนาดใหญ่
เว็บไซต์	https://github.com/NancyFx/Nancy
วันเข้าถึงข้อมูล	30 กันยายน 2561

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.1.2. การตรวจหาอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์

ผลการตรวจหาอาคิเทกเจอร์สเมลในโปรเจคที่ใช้ในงานวิจัยนี้ด้วยเครื่องมือพาณิชย์จะถูกนำมาใช้เป็นข้อมูลพื้นฐานสำหรับการเปรียบเทียบความสามารถในการตรวจหาอาคิเทกเจอร์สเมลในโปรเจคของเครื่องมือที่พัฒนาขึ้น โดยเครื่องมือที่นำมาวิเคราะห์โปรเจคเป็นเครื่องมือพาณิชย์ที่นำเสนอในขั้นตอนที่ 3.2 โดยผลการวิเคราะห์ข้อมูลทั่วไปโปรเจค ซึ่งประกอบด้วย จำนวนเนมสเปซ จำนวนคลาส จำนวนบรรทัด และผลการตรวจหาอาคิเทกเจอร์สเมล แสดงรายละเอียดดังตารางที่ 5.4

ตารางที่ 5.4 ผลการวิเคราะห์และตรวจหาอาคิเทกเจอร์สเมลในโปรเจคด้วยเครื่องมือพาณิชย์

ชื่อโปรเจค	ตัวชี้วัดซอต์แวร์			จำนวนอาคิเทกเจอร์สเมล	
	จำนวนเนมสเปซ	จำนวนคลาส	จำนวนบรรทัด	แพ็กเกจขนาดใหญ่เกิน	อันสเตเบิลดีเพนเดนซี
Hyperion	6	165	6,930	2	4
Hangfire.Core	16	302	28,189	3	7
Nancy	37	479	45,839	2	19

จากตารางที่ 5.4 แสดงผลการวิเคราะห์และตรวจหาอาคิเทกเจอร์สเมลในโปรเจคที่ใช้ในงานวิจัยด้วยเครื่องมือพาณิชย์ แสดงให้เห็นจำนวนเนมสเปซ จำนวนคลาส และจำนวนบรรทัด ของแต่ละโปรเจคที่เลือกมาที่มีขนาดที่แตกต่างกัน และผลการตรวจหาจำนวนอาคิเทกเจอร์สเมลนั้นในทุกโปรเจคพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและอันสเตเบิลดีเพนเดนซี โดยโปรเจคที่มีขนาดใหญ่หรือมีเนมสเปซจำนวนมากมีโอกาสพบอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีในจำนวนมากกว่าโปรเจคที่มีขนาดเล็กกว่า โดยรายละเอียดของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี ได้แสดงรายละเอียดเพิ่มเติมในภาคผนวก ข.1.

5.2.ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

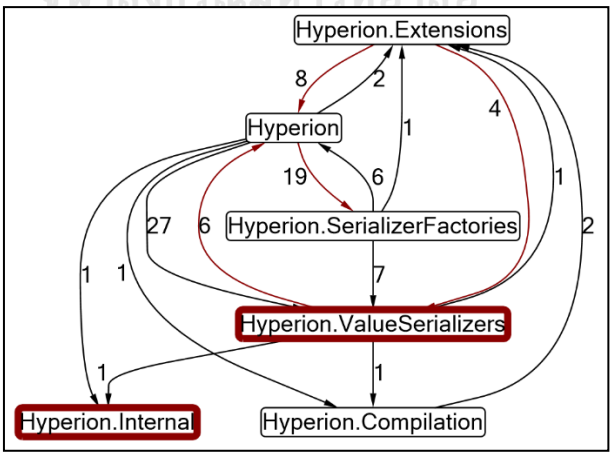
การประเมินผลในขั้นตอนนี้ จะเริ่มจากขั้นตอนการเตรียมโปรเจคโอเพนซอร์สให้อยู่ในรูปแบบโครงสร้างและความสัมพันธ์ของข้อมูล จากข้อมูลโปรเจคไฟล์ที่ใช้ในงานวิจัยเพื่อวิเคราะห์โครงสร้างของโปรเจคและความสัมพันธ์ของข้อมูล จากนั้นในขั้นตอนการตรวจหาอาคิเทกเจอร์สเมลจะนำข้อมูลโครงสร้างของโปรเจคและข้อมูลความสัมพันธ์ของคลาสที่เตรียมไว้ มาตรวจหาอาคิเทกเจอร์สเมลตามนิยาม โดยผลการวิเคราะห์ข้อมูลและผลการตรวจหาอาคิเทกเจอร์สเมลจากเครื่องมือที่พัฒนาขึ้นนั้น แสดงรายละเอียดดังตารางที่ 5.5 โดยแสดงรายละเอียด จำนวนเนมสเปซ จำนวนคลาส จำนวนบรรทัดและจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบของแต่ละโปรเจค

ตารางที่ 5.5 ผลการวิเคราะห์และตรวจหาอาคิเทกเจอร์สเมลในโปรเจคด้วยเครื่องมือที่พัฒนา

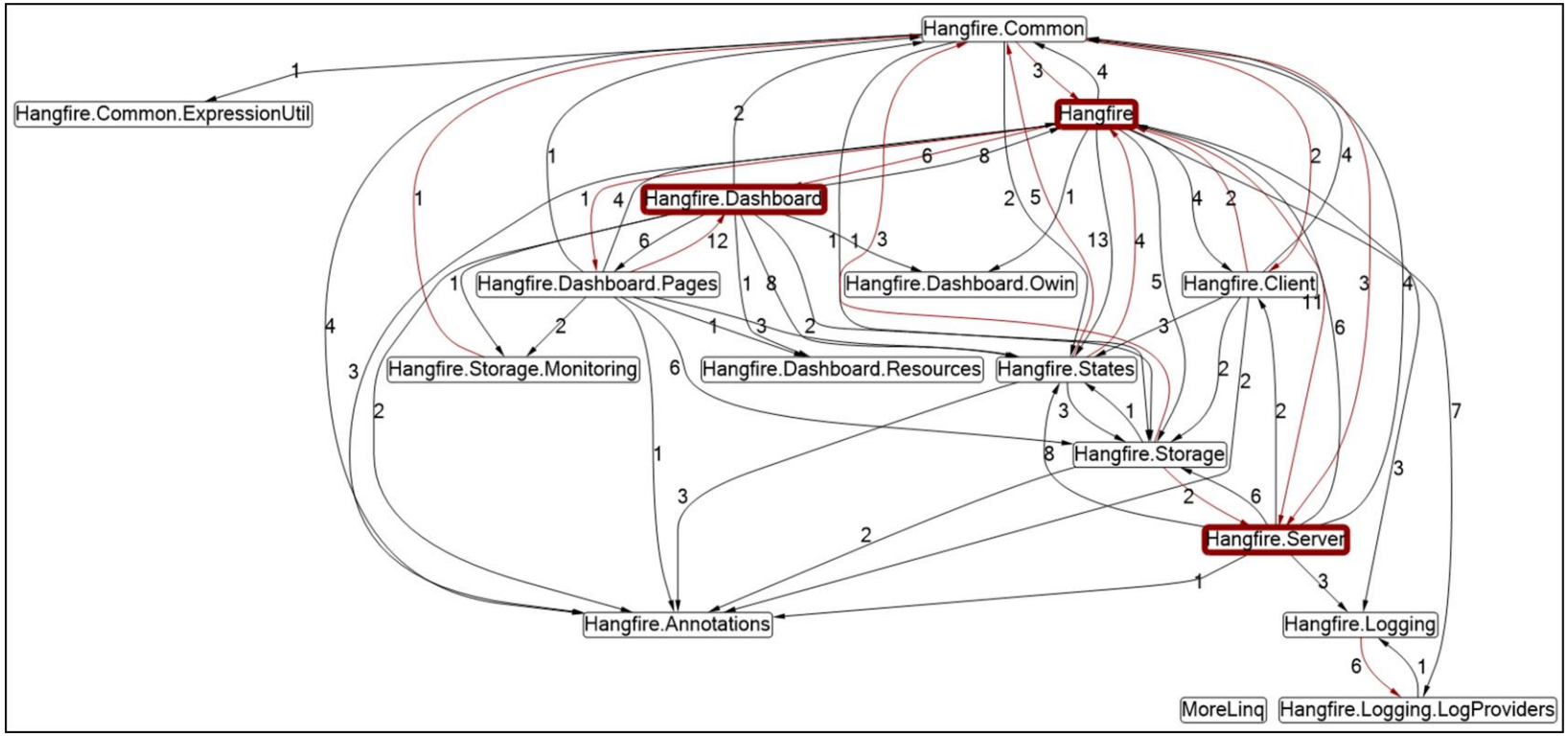
ชื่อโปรเจค	ตัวชี้วัดซอต์แวร์			จำนวนอาคิเทกเจอร์สเมล	
	จำนวนเนมสเปซ	จำนวนคลาส	จำนวนบรรทัด	แพ็กเกจขนาดใหญ่เกิน	อันสเตเบิลดีเพนเดนซี
Hyperion	6	165	5,160	2	3
Hangfire.Core	16	302	26,258	3	8
Nancy	37	479	38,554	2	18

นอกจากนี้เครื่องมือที่พัฒนาขึ้นสามารถวิเคราะห์ข้อมูลระดับเนมสเปซของแต่ละโปรเจคเพื่อแสดงค่าตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้องในระดับเนมสเปซ แสดงความสัมพันธ์ของข้อมูลในระดับเนมสเปซในรูปแบบเมทริกซ์การพึ่งพา (dependency matrix) และรายละเอียดของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี โดยแสดงรายละเอียดเพิ่มเติมในภาคผนวก ข.2.1. - ข.2.4.

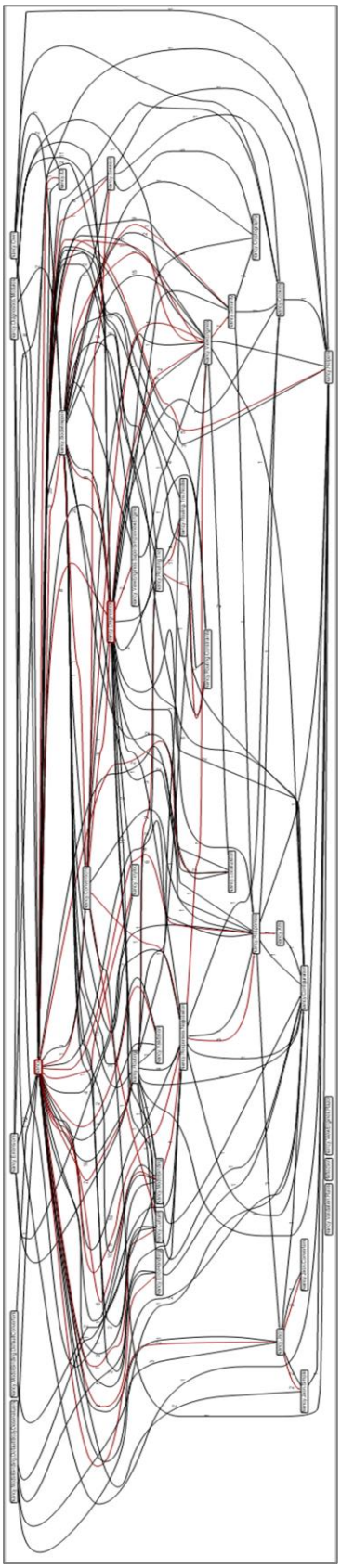
โดยจากผลการวิเคราะห์ของเครื่องมือสามารถนำมาสร้างกราฟการพึ่งพาระหว่างเนมสเปซในโปรเจค ประกอบด้วย กราฟการพึ่งพาของเนมสเปซในโปรเจค Hyperion แสดงดังรูปที่ 5.2 กราฟการพึ่งพาของเนมสเปซในโปรเจค Hangfire.Core แสดงดังรูปที่ 5.3 กราฟการพึ่งพาของเนมสเปซในโปรเจค Nancy แสดงดังรูปที่ 5.4 โดยเนมสเปซที่มีกรอบสีแดงแสดงถึงเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน และเส้นเชื่อมสีแสดงแสดงถึงการพึ่งพาเนมสเปซที่มีความเสถียรภาพน้อยกว่า



รูปที่ 5.2 กราฟการพึ่งพาของเนมสเปซในโปรเจค Hyperion จากเครื่องมือที่พัฒนา



รูปที่ 5.3 กราฟการพึ่งพาของเนมสเปซในโปรเจค Hangfire.Core จากเครื่องมือที่พัฒนา



รูปที่ 5.4 กราฟการพึ่งพาของเมตเปโตในโปรเจค Nancy จากเครื่องมือที่พัฒนา

การเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมลระหว่างเครื่องมือที่พัฒนาและเครื่องมือเชิงพาณิชย์เพื่อประเมินประสิทธิภาพความสามารถของเครื่องมือที่พัฒนาขึ้น โดยผลการเปรียบเทียบประสิทธิภาพการตรวจหาอาคิเทกเจอร์สเมลมีรายละเอียดดังนี้

(1) ผลการเปรียบเทียบจำนวนเนมสเปซและจำนวนคลาสที่ตรวจพบ

จากตารางที่ 5.4 และตารางที่ 5.5 แสดงจำนวนเนมสเปซ จำนวนคลาส และจำนวนบรรทัดที่ตรวจพบด้วยเครื่องมือพาณิชย์และเครื่องมือที่พัฒนา โดยแสดงผลการเปรียบเทียบ ดังตารางที่ 5.6

ตารางที่ 5.6 ผลการเปรียบเทียบจำนวนเนมสเปซและจำนวนคลาสที่ตรวจพบ

ชื่อโปรเจค	จำนวนเนมสเปซ		จำนวนคลาส		จำนวนบรรทัด	
	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา
Hyperion	6	6	165	165	6,930	5,160
Hangfire.Core	16	16	302	302	28,189	26,258
Nancy	37	37	479	479	45,839	38,554

จากตารางที่ 5.6 แสดงผลการเปรียบเทียบตัวชี้วัดซอฟต์แวร์ระดับโปรเจคที่ตรวจพบด้วยเครื่องมือพาณิชย์และเครื่องมือที่พัฒนาขึ้น โดยจำนวนเนมสเปซและจำนวนคลาสที่พบในแต่ละโปรเจคมีจำนวนที่เท่ากัน แต่จำนวนบรรทัดของเครื่องมือที่พัฒนาขึ้นให้จำนวนบรรทัดที่น้อยกว่าในแต่ละโปรเจค โดยเหตุผลมาจากการนับจำนวนบรรทัดของเครื่องมือที่พัฒนาขึ้นนั้นจะไม่นับบรรทัดที่ไม่มีข้อมูล แต่ในส่วนของเครื่องมือพาณิชย์นับทุกบรรทัดที่มีในโปรเจค ดังนั้นในส่วนนี้ทำให้ค่าจำนวนบรรทัดมีจำนวนที่แตกต่างกัน

(2) ผลการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบ

จากตารางที่ 5.4 และตารางที่ 5.5 แสดงจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบด้วยเครื่องมือพาณิชย์และเครื่องมือที่พัฒนา โดยแสดงผลการเปรียบเทียบ ดังตารางที่ 5.7

ตารางที่ 5.7 ผลการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบ

ชื่อโปรเจค	จำนวนอาคิเทกเจอร์สเมล			
	แพ็กเกจขนาดใหญ่เกิน		อันสเตเบิลดีเพนเดนซี	
	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา
Hyperion	2	2	4	3
Hangfire.Core	3	3	7	8
Nancy	2	2	19	18

จากตารางที่ 5.7 แสดงผลการเปรียบเทียบการตรวจหาจำนวนอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์และเครื่องมือที่พัฒนาขึ้น โดยจำนวนแพ็กเกจขนาดใหญ่เกินที่ตรวจพบมีจำนวนเท่ากัน แต่จำนวนอันสเตเบิลดีเพนเดนซี ที่ตรวจพบจากทั้ง 2 เครื่องมือมีปริมาณที่แตกต่างกัน โดยสาเหตุมาจากการพิจารณาการพึ่งพาที่แตกต่างกันส่งผลต่อการคำนวณค่าตัวชี้วัดซอฟต์แวร์ต่าง ๆ รวมถึงค่าความไม่แน่นอน ดังนั้นผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีจึงมีความคลาดเคลื่อนกันเล็กน้อย โดยรายละเอียดการเปรียบเทียบการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซีในระดับเนมสเปซ แสดงรายละเอียดเพิ่มเติมในภาคผนวก ข.2.5. - ข.2.6.

การประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซีสามารถทำได้โดยการเปรียบเทียบผลการตรวจหาอาคิเทกเจอร์สเมลแต่ละรูปแบบในแต่ละเนมสเปซที่พบระหว่างเครื่องมือพาณิชย์และเครื่องมือที่พัฒนาขึ้นด้วยการคำนวณค่าอินเตอร์เซกชันโอเวอร์ยูเนียน โดยที่แสดงผลการประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินดังตารางที่ 5.8 และแสดงผลการประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีดังตารางที่ 5.9

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 5.8 แสดงผลการประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน

ชื่อโปรเจค	ค่าอินเตอร์เซกชันโอเวอร์ยูเนียน
Hyperion	1.00
Hangfire.Core	1.00
Nancy	1.00
เฉลี่ย	**1.00**

ตารางที่ 5.9 แสดงผลการประเมินประสิทธิภาพเครื่องมือสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

ชื่อโปรเจค	ค่าอินเตอร์เซกชันโอเวอร์ยูเนียน
Hyperion	0.75
Hangfire.Core	0.67
Nancy	0.68
เฉลี่ย	**0.69**

จากตารางที่ 5.8 แสดงประสิทธิภาพของเครื่องมือที่พัฒนาเมื่อเปรียบเทียบกับเครื่องมือพาณิชย์ในการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินได้อย่างมีประสิทธิภาพ โดยมีค่าอินเตอร์เซกชันโอเวอร์ยูเนียนอยู่ที่ 1.00 ซึ่งบ่งบอกถึงเครื่องมือที่พัฒนาขึ้นสามารถตรวจพบอาคิเทกเจอร์สเมลรูปแบบนี้ได้เทียบเท่ากับเครื่องมือพาณิชย์ และจากตารางที่ 5.9 แสดงประสิทธิภาพของเครื่องมือที่พัฒนาเมื่อเปรียบเทียบกับเครื่องมือพาณิชย์ในการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีได้อย่างดี โดยมีค่าอินเตอร์เซกชันโอเวอร์ยูเนียนอยู่ที่ 0.69 บ่งบอกถึงเครื่องมือที่พัฒนามีสามารถตรวจพบอาคิเทกเจอร์สเมลรูปแบบนี้ในทิศทางเดียวกัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.3.ประเมินผลวิธีการรีแฟคทอริงของเครื่องมือที่พัฒนา

ในขั้นตอนนี้ประกอบด้วยการประเมินผลขั้นตอนการตรวจหาคอมมิวนิตีภายในเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน และขั้นตอนการแยกเนมสเปซย่อย โดยรายละเอียดการตรวจหาคอมมิวนิตีของแต่ละโปรเจคแสดงรายละเอียดเพิ่มเติมในภาคผนวก ข.3.1.

โดยขั้นตอนประเมินผลการแยกเนมสเปซย่อย เป็นการค้นหาวิธีการรีแฟคทอริงที่เป็นไปได้และให้ผลลัพธ์ดีที่สุดตามขั้นตอนการพิจารณาจากกลุ่มคอมมิวนิตีที่ได้ โดยกลุ่มคอมมิวนิตีที่ให้ผลลัพธ์การรีแฟคทอริงที่ดีที่สุดแสดงรายละเอียดเพิ่มเติมในภาคผนวก ข.3.2.

การประเมินผลวิธีการรีแฟคทอริงของเครื่องมือที่พัฒนาในงานวิจัยนี้ว่ามีประสิทธิภาพสามารถทำได้โดยเปรียบเทียบค่าตัวชี้วัดซอฟต์แวร์และจำนวนอาคิเทกเจอร์สเมลที่ตรวจพบของเนมสเปซก่อนการรีแฟคทอริงและหลังการรีแฟคทอริง โดยตารางที่ 5.10 แสดงการเปรียบเทียบค่าตัวชี้วัดพื้นฐานก่อนและหลังกระบวนการรีแฟคทอริง และตารางที่ 5.11 แสดงการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลและค่าตัวชี้วัดก่อนและหลังกระบวนการรีแฟคทอริง โดยตัวย่อในตารางมีความหมายดังนี้

- ก่อน หมายถึง ก่อนกระบวนการรีแฟคทอริง
- หลัง หมายถึง หลังกระบวนการรีแฟคทอริง
- แยก หมายถึง เนมสเปซที่แนะนำให้แยกออกมา
- คงเหลือ หมายถึง เนมสเปซที่คงเหลือในเนมสเปซเดิม

ตารางที่ 5.10 แสดงการเปรียบเทียบค่าตัวชี้วัดพื้นฐานก่อนและหลังกระบวนการรีแฟคทอริง

ชื่อเนมสเปซ	จำนวนคลาส			จำนวนบรรทัด		
	ก่อน	หลัง		ก่อน	หลัง	
		แยก	คงเหลือ		แยก	คงเหลือ
Hyperion.Internal	74	4	70	539	70	469
Hyperion.ValueSerializers	32	29	3	1154	1043	111
Hangfire	47	27	17	4122	1805	2144
		3			173	
Hangfire.Dashboard	49	-	-	2993	-	-
Hangfire.Server	35	29	6	1843	1535	308
Nancy	96	29	67	10980	3571	7409
Nancy.Diagnostics	37	16	21	1691	930	761

จากตารางที่ 5.10 แสดงการเปรียบเทียบค่าตัวชี้วัดพื้นฐานก่อนและหลังกระบวนการรีแฟคทอริงของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยจาก 3 โปรเจคที่นำมาใช้ในการทดลองตรวจพบอาคิเทกเจอร์สเมลจำนวน 7 เนมสเปซ โดยในตารางแสดงการ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

เปรียบเทียบจำนวนคลาสและจำนวนบรรทัด ของเนมสเปซก่อนกระบวนการรีแฟคทอริง และเนมสเปซหลังกระบวนการรีแฟคทอริงที่ถูกแยกเนมสเปซใหม่ออกมา และคลาสคงเหลือในเนมสเปซเดิม จากตารางนี้จะพบว่าเนมสเปซ Hangfire.Dashboard เป็นเนมสเปซที่ไมสามารถค้นหาวิธีการรีแฟคทอริงที่ให้ผลลัพธ์ที่ดีขึ้นได้ ดังนั้นเครื่องมือที่พัฒนาจะไม่แนะนำให้แยกเนมสเปซย่อย

จากตารางที่ 5.11 แสดงการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลและค่าตัวชี้วัดก่อนและหลังกระบวนการรีแฟคทอริง ซึ่งพบว่าเมื่อพิจารณาที่จำนวนอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินในเนมสเปซ Hyperion.ValueSerializers Hangfire Hangfire.Server และ Nancy.Diagnostics มีจำนวนที่ลดลง แต่เนมสเปซ Hyperion.Internal และ Nancy มีจำนวนไม่ลดลง เนื่องจากเครื่องมือไม่สามารถหาวิธีที่แยกเนมสเปซเพิ่มแล้วยังให้ผลลัพธ์ที่ดีขึ้น

เมื่อพิจารณาที่จำนวนอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี พบว่าเนมสเปซเกือบทั้งหมดสามารถหลีกเลี่ยงการเพิ่มจำนวนอาคิเทกเจอร์สเมลรูปแบบนี้ขึ้นได้ ยกเว้นเนมสเปซ Hyperion.ValueSerializers มีการเพิ่มจำนวนอาคิเทกเจอร์สเมลรูปแบบนี้ขึ้น แต่เมื่อวิเคราะห์การตรวจหาอาคิเทกเจอร์สเมลรูปแบบนี้ในระดับเนมสเปซพบว่าจำนวนที่เพิ่มขึ้นมาจากเนมสเปซที่แยกมาใหม่ แสดงว่าถ้าเนมสเปซก่อนการรีแฟคทอริงมีการตรวจพบอาคิเทกเจอร์สเมลรูปแบบนี้จะมีโอกาสที่หลังการรีแฟคทอริงจะทำให้เนมสเปซที่แยกมาใหม่ตรวจพบด้วยทำให้จำนวนอาคิเทกเจอร์สเมลรูปแบบนี้เพิ่มมากขึ้น แต่ในบางกรณีเช่นเนมสเปซ Nancy และ Nancy.Diagnostics เครื่องมือสามารถหลีกเลี่ยงการเพิ่มจำนวนอาคิเทกเจอร์สเมลรูปแบบนี้ได้

เมื่อพิจารณาที่ค่าตัวชี้วัดค่าความเชื่อมแน่น พบว่าเนมสเปซที่แยกมาใหม่ให้ค่าความเชื่อมแน่นที่สูงกว่าเนมสเปซก่อนการรีแฟคทอริง ซึ่งบ่งบอกถึงคลาสภายในเนมสเปซย่อยนี้มีความสัมพันธ์กันอย่างหนาแน่นขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เมื่อพิจารณาค่าตัวชี้วัดค่าการคู่ควบ พบว่าเนมสเปซแยกมาใหม่ให้ค่าการคู่ควบที่ไม่เพิ่มมากขึ้นกว่าเนมสเปซก่อนการรีแฟคทอริง ซึ่งบ่งบองถึงเนมสเปซที่แยกออกมาไม่ได้มีความสัมพันธ์กับเนมสเปซรอบข้างเพิ่มมากขึ้น

ตารางที่ 5.11 แสดงการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลและค่าตัวชี้วัดก่อนและหลังกระบวนการรีแฟคทอริง

ชื่อเนมสเปซ	จำนวนอาคิเทกเจอร์สเมล				ค่าความเชื่อมแน่น			ค่าการคู่ควบ		
	แพ็กเกจขนาดใหญ่เกิน		อันสเตเบิลดีเพนเดนซี		ก่อน	หลัง		ก่อน	หลัง	
	ก่อน	หลัง	ก่อน	หลัง		แยก	คงเหลือ		แยก	คงเหลือ
Hyperion.Internal	2	2	3	**3**	0.53	**1.50**	0.49	0	**0**	**0**
Hyperion.ValueSerializers	2	**1**	3	4	0.38	**0.41**	0.33	4	**4**	**3**
Hangfire	3	**2**	8	**8**	1.28	**1.30**	0.71	11	**11**	**7**
						1.00			**7**	
Hangfire.Dashboard	3	-	8	-	1.82	-	-	9	-	-
Hangfire.Server	3	**2**	8	**8**	1.49	**1.66**	0.67	7	**6**	8
Nancy	2	2	18	**17**	1.39	**1.55**	0.79	19	**13**	**15**
Nancy.Diagnostics	2	**1**	18	**18**	1.00	**1.13**	0.67	18	**15**	**12**

5.4.ประเมินผลการรีแฟคทอริงโปรเจคตามที่เครื่องมือแนะนำ

ในส่วนการประเมินผลขั้นตอนนี้จะนำวิธีการรีแฟคทอริงไปแก้ไขโครงสร้างของโปรเจคตามวิธีการรีแฟคทอริงที่เครื่องมือแนะนำ เพื่อประเมินผลว่าหลังจากการนำไปแก้ไขโครงสร้างจริงแล้ว ได้ผลลัพธ์ดังที่เครื่องมือได้นำเสนอ โดยมีรายละเอียดการเปรียบเทียบจำนวนอาคิเทกเจอร์สเมลหลังการรีแฟคทอริงที่เครื่องมือแนะนำ และผลการรีแฟคทอริงจริงและตรวจหาจำนวนอาคิเทกเจอร์สเมลด้วยเครื่องมือที่พัฒนาขึ้นดังตารางที่ 5.12 โดยตัวย่อในตารางมีความหมายดังนี้

- หลัง หมายถึง ผลลัพธ์หลังจากกระบวนการรีแฟคทอริงที่เครื่องมือที่พัฒนาแนะนำ
- แก้ไข หมายถึง ผลลัพธ์การแก้ไขโครงสร้างจริงตามเครื่องมือที่พัฒนาแนะนำด้วยเครื่องมือไมโครซอฟวิชวลสตูดิโอในการแยกคลาสไปยังเนมสเปซใหม่ และตรวจผลด้วยเครื่องมือที่พัฒนาอีกครั้ง

ตารางที่ 5.12 ผลการเปรียบเทียบการตรวจหาจำนวนอาคิเทกเจอร์สเมลหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริง

ชื่อโปรเจค	ชื่อเนมสเปซ	จำนวนอาคิเทกเจอร์สเมล			
		แพ็กเกจขนาดใหญ่เกิน		อันสเตเบิลดีเพนเดนซี	
		หลัง	แก้ไข	หลัง	แก้ไข
Hyperion	Hyperion.Internal	2	2	3	3
Hyperion	Hyperion.ValueSerializers	1	1	4	4
Hangfire.Core	Hangfire	2	2	9	9
Hangfire.Core	Hangfire.Server	2	2	8	8
Nancy	Nancy	2	2	17	17
Nancy	Nancy.Diagnostics	1	1	18	18

จากตารางที่ 5.12 แสดงผลการเปรียบเทียบการตรวจหาจำนวนอาคิเทกเจอร์สเมลหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริง โดยให้ผลลัพธ์การตรวจหาจำนวนอาคิเทกเจอร์สเมลที่มีค่าตรงกันในแต่ละโปรเจค ซึ่งบ่งบองถึงความน่าเชื่อถือของเครื่องมือในการแนะนำวิธีการรีแฟคทอริงว่าผลลัพธ์ที่ได้จากเครื่องมือมีความน่าเชื่อถือเมื่อเปรียบเทียบกับการทดสอบแก้ไขโครงสร้างจริง โดยสามารถนำผลลัพธ์มาคำนวณค่าอินเตอร์เซกชันโอเวอร์ยูเนียน โดยให้ผลลัพธ์ดังตารางที่ 5.13

สำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน และตารางที่ 5.14 สำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

ตารางที่ 5.13 แสดงผลการประเมินประสิทธิภาพเครื่องมือหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริงสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน

ชื่อโปรเจค	ค่าอินเตอร์เซกชันโอเวอร์ยูเนียน
Hyperion	1.00
Hangfire.Core	1.00
Nancy	1.00
เฉลี่ย	**1.00**

ตารางที่ 5.14 แสดงผลการประเมินประสิทธิภาพเครื่องมือหลังการรีแฟคทอริงและการแก้ไขโครงสร้างโปรเจคจริงสำหรับการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

ชื่อโปรเจค	ค่าอินเตอร์เซกชันโอเวอร์ยูเนียน
Hyperion	1.00
Hangfire.Core	1.00
Nancy	1.00
เฉลี่ย	**1.00**

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6
บทสรุปและข้อเสนอแนะ

6.1.สรุปผลการวิจัย

งานวิจัยนี้นำเสนอวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี เพื่อลดผลกระทบจากอาคิเทคเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินให้มีขนาดเล็กลงด้วยวิธีการเอ็กซ์แทรกต์แพ็กเกจ และพิจารณาผลกระทบเพื่อหลีกเลี่ยงการเกิดอาคิเทกเจอร์สเมลบนพื้นฐานการพึ่งพาขึ้นมาใหม่ โดยมุ่งเน้นที่อาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี จากวิธีการที่นำเสนอนี้ ได้นำมาพัฒนาเป็นเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี ด้วยภาษาซีชาร์ป โดยเครื่องมือนี้สามารถวิเคราะห์ข้อมูลโครงสร้างโปรเจค ตรวจหาอาคิเทกเจอร์สเมล และแนะนำวิธีการรีแฟคทอริงที่เหมาะสมเพื่อลดผลกระทบจากอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินที่ตรวจพบ

วิธีการที่นำเสนอจะเริ่มจากการแปลงข้อมูลไฟล์ของโปรเจคให้อยู่ในรูปแบบข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์ข้อมูล จากนั้นเครื่องมือจะตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซีที่มีในระบบก่อนการรีแฟคทอริง จากนั้นจะนำเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินมาสร้างกราฟการพึ่งพาจากคลาสภายในเนมสเปซ เนมสเปซรอบข้าง และความสัมพันธ์ต่าง ๆ หลังจากได้กราฟการพึ่งพาที่มีน้ำหนักและมีทิศทาง จะนำไปหาคอมมิวนิตีภายในเนมสเปซ โดยประยุกต์ใช้อัลกอริทึมลูแวน เมื่อได้คอมมิวนิตีภายในต่าง ๆ เครื่องมือจะค้นหาวิธีการรีแฟคทอริงที่ให้ผลลัพธ์ดีที่สุดเพื่อที่จะแยกกลุ่มของคลาสที่มีความสัมพันธ์ที่แน่นหนาออกมาเป็นแนมสเปซใหม่ โดยให้ค่าความเชื่อมแน่นที่สูงขึ้นและค่าการคู่ควบที่ไม่มากกว่าเนมสเปซก่อนการรีแฟคทอริง โดยค่าตัวชี้วัดซอฟต์แวร์ทั้งสองค่านี้สามารถบ่งชี้ถึงการออกแบบที่ดี โดยผลลัพธ์ที่ได้จากเครื่องมือจะแสดงผลออกมาในรูปแบบกราฟการพึ่งพาหลังกระบวนการรีแฟคทอริง รวมถึงรายงานกลุ่มของคลาสที่แนะนำให้แยกไปยังเนมสเปซใหม่

ในการทดลองความสามารถเครื่องมือในงานวิจัยนี้ นำโปรเจคโอเพนซอร์สจำนวน 3 โปรเจคที่มีขนาดแตกต่างกันมาใช้ในการทดลอง โดยเครื่องมือตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินจำนวน 7 เนมสเปซ จากทั้งหมด 59 เนมสเปซใน 3 โปรเจค โดยเครื่องมือสามารถหาวิธีการรีแฟคทอริงที่ให้ผลลัพธ์ที่ดีขึ้นกับ 6 เนมสเปซ โดยเนมสเปซที่ไม่สามารถหาวิธีการรีแฟคทอริงที่ดีขึ้นได้เนื่องจากเนมสเปซดังกล่าวมีค่าความเชื่อมแน่นที่สูงหรือบ่งบอกถึงคลาสภายในเนมสเปซนี้มีความสัมพันธ์ที่แน่นหนา โดยวิธีการรีแฟคทอริงที่เครื่องมือแนะนำสำหรับ 6 เนมสเปซนั้นให้ค่าความเชื่อมแน่นที่สูงขึ้นและค่าการคู่ควบที่ไม่มากกว่าเดิมสำหรับเนมสเปซที่แยกออกมาเปรียบเทียบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กับเนมสเปซก่อนการรีแฟคทอริง โดยผลการตรวจหาจำนวนอาคิเทกเจอร์สเมลนั้น 4 ใน 6 วิธีที่นำเสนอสามารถลดจำนวนอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินลง โดย 2 วิธีที่ไม่สามารถกำจัดอาคิเทกเจอร์สเมลรูปแบบนี้ได้เนื่องจากจำนวนคลาสภายในเนมสเปซมีจำนวนมากเกินไปและเมื่อเครื่องมือพยายามค้นหาวิธีการแยกเนมสเปซเพิ่มเติมก็ไม่พบวิธีการที่ให้ผลลัพธ์ที่ดีขึ้น และ 5 ใน 6 วิธีที่นำเสนอสามารถหลีกเลี่ยงการเกิดอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีขึ้นมาใหม่ อาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีสามารถเกิดขึ้นใหม่ได้ง่ายเมื่อมีการเปลี่ยนแปลงความสัมพันธ์ระหว่างเนมสเปซ โดยที่เนมสเปซก่อนกระบวนการรีแฟคทอริงตรวจพบอาคิเทกเจอร์สเมลนี้ จะมีโอกาสตรวจพบอาคิเทกเจอร์สเมลรูปแบบนี้เกิดขึ้นใหม่ในเนมสเปซที่แยกออกมา ดังนั้นจากผลการทดสอบความสามารถของเครื่องมือพบว่าเครื่องมือที่พัฒนาขึ้นจากวิธีที่นำเสนอสามารถตรวจหาจำนวนอาคิเทกเจอร์สเมลที่เกิดขึ้นในโปรเจค และสามารถเสนอแนะวิธีการรีแฟคทอริงที่เหมาะสมได้ โดยให้ผลการทดลองโดยการเปรียบเทียบก่อนและหลังการรีแฟคทอริงไปในทางที่ดีขึ้น

6.2.ข้อจำกัดของงานวิจัย

1. การแปลงข้อมูลไฟล์ของโปรเจคให้อยู่ในรูปแบบข้อมูลโครงสร้างของโปรเจคและความสัมพันธ์นั้นจะพิจารณาเฉพาะความสัมพันธ์ของคลาสต่าง ๆ ภายในโปรเจคที่พิจารณา โดยไม่พิจารณาถึงคลาสในโปรเจคอื่น ๆ ที่มีความสัมพันธ์กับโปรเจคที่กำลังพิจารณา
2. ความสัมพันธ์ของคลาสต่าง ๆ ภายในโปรเจค จะพิจารณาจากการอ้างถึงคลาสนั้น จากตัวแปรระดับฟิลด์ และระดับพร็อพเพอร์ตี้ ในคลาส อินเตอร์เฟซ, สตรัค และอีนัม และจากตัวแปรภายในเมธอทเท่านั้น
3. เนมสเปซที่มีความสัมพันธ์ของคลาสภายในที่แน่นหนาหรือมีค่าความเชื่อมแน่นสูง ทำให้เครื่องมือไม่สามารถหาวิธีการรีแฟคทอริงที่ให้ผลดีขึ้นได้ ในกรณีดังกล่าวจำเป็นต้องใช้นักพัฒนาที่มีความเชี่ยวชาญในการวิเคราะห์โครงสร้างดังกล่าวว่าก่อนเพื่อลดความซับซ้อนของความสัมพันธ์ของคลาสภายในเนมสเปซลง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

6.3.แนวทางการพัฒนางานวิจัยต่อ

1. ศึกษาการจัดการคลาสคงเหลือในเนมสเปซหลังจากกระบวนการรีแฟคทอริง ด้วยเทคนิคการรีแฟคทอริงอื่น ๆ เพิ่มเติม เช่น การย้ายคลาส เพื่อเพิ่มคุณภาพซอฟต์แวร์โดยรวมของโปรเจคให้ดีขึ้น
2. ศึกษาถึงผลกระทบของอาคิเทกเจอร์สเมลรูปแบบอื่น ๆ ที่มีโอกาสเกิดขึ้นหลังจากกระบวนการรีแฟคทอริง
3. ศึกษาวิธีการรีแฟคทอริงในอาคิเทกเจอร์สเมลรูปแบบอื่น ๆ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บรรณานุกรม

1. Fowler, M., *Refactoring: Improving the Design of Existing Code*. 1999, Boston, MA, USA: Addison-Wesley.
2. Lippert, M. and S. Roock, *Refactoring in Large Software Projects: Performing Complex Restructurings Successfully*. 2006: Wiley.
3. Garcia, J., et al., *Toward a Catalogue of Architectural Bad Smells*, in *Proceedings of the 5th International Conference on the Quality of Software Architectures: Architectures for Adaptive Software Systems*. 2009, Springer-Verlag: East Stroudsburg, PA, USA. p. 146-162.
4. Arcelli Fontana, F., et al., *Automatic Detection of Instability Architectural Smells*. 2016. 433-437.
5. Sharma, T. and D. Spinellis, *A survey on software smells.* Journal of Systems and Software, 2018. **138**: p. 158-173.
6. Garcia, J., et al., *Identifying Architectural Bad Smells*, in *Proceedings of the 2009 European Conference on Software Maintenance and Reengineering*. 2009, IEEE Computer Society. p. 255-258.
7. Mo, R., et al., *Hotspot Patterns: The Formal Definition and Automatic Detection of Architecture Smells*, in *Proceedings of the 2015 12th Working IEEE/IFIP Conference on Software Architecture*. 2015, IEEE Computer Society. p. 51-60.
8. Le, D., et al., *Toward a Classification Framework for Software Architectural Smells.* Technical Report csse. usc. edu, 2017.
9. Martin, R.C. and M. Martin, *Agile Principles, Patterns, and Practices in C\# (Robert C. Martin)*. 2006: Prentice Hall PTR.
10. Samarthyam, G., G. Suryanarayana, and T. Sharma, *Refactoring for software architecture smells*, in *Proceedings of the 1st International Workshop on Software Refactoring*. 2016, ACM: Singapore, Singapore. p. 1-4.
11. Bavota, G. *Using structural and semantic information to support software refactoring*. in *2012 34th International Conference on Software Engineering (ICSE)*. 2012.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

12. Malliaros, F. and M. Vazirgiannis, *Clustering and Community Detection in Directed Networks: A Survey*. Vol. 533. 2013.
13. Fortunato, S. and D. Hric, *Community detection in networks: A user guide.* CoRR, 2016. **abs/1608.00163**.
14. Blondel, V.D., et al., *Fast unfolding of communities in large networks.* Journal of Statistical Mechanics: Theory and Experiment, 2008. **2008**(10): p. P10008.
15. Neo4j, I. *5.1. The Louvain algorithm – Chapter 5. Community detection algorithms*. [cited 2019 February 3, 2019]; Available from: https://neo4j.com/docs/graph-algorithms/current/algorithms/louvain.
16. Rizzi, L., F.A. Fontana, and R. Roveda, *Support for architectural smell refactoring*, in *Proceedings of the 2nd International Workshop on Refactoring*. 2018, ACM: Montpellier, France. p. 7-10.
17. Pan, W., B. Jiang, and B. Li, *Refactoring Software Packages via Community Detection in Complex Software Networks*. Vol. 10. 2013.
18. Designite. *Designite – Reduce Technical Debt of your Software*. [cited 2019 February 3, 2019]; Available from: http://www.designite-tools.com.

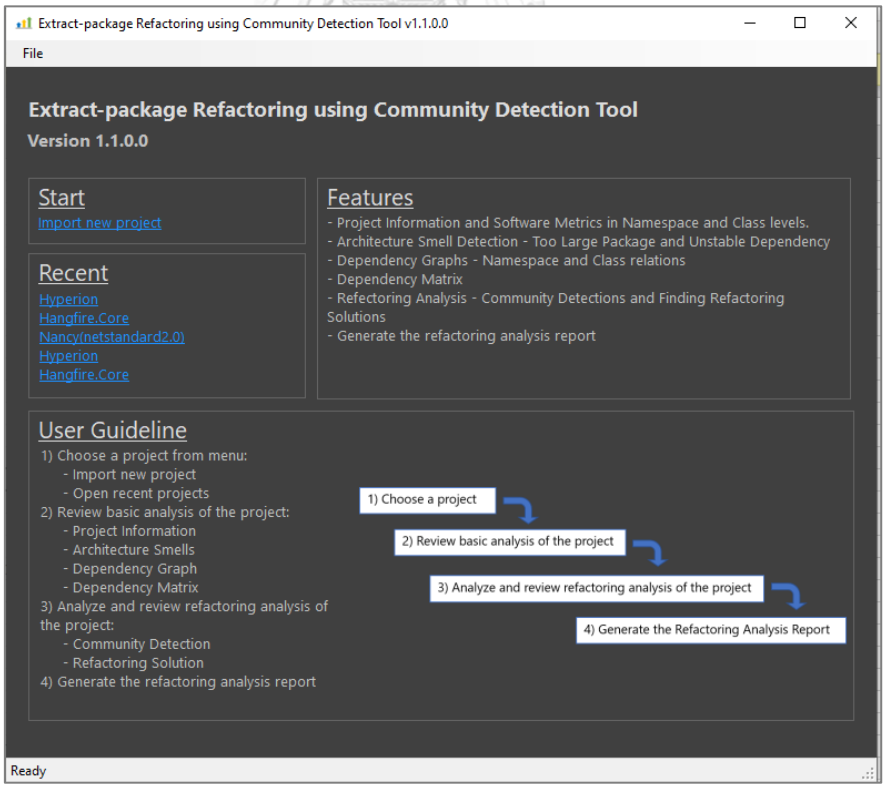
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก
คู่มือการใช้งานเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี

ในภาคผนวกนี้จะอธิบายการใช้งานเครื่องมือสนับสนุนวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกินโดยใช้การตรวจหาคอมมิวนิตี ในส่วนของการแสดงผล โดยจะแบ่งการแสดงผลออกเป็น 4 ส่วนหลัก คือ การนำข้อมูลโปรเจคการทำงานหลักเข้าระบบ การแสดงผลวิเคราะห์โปรเจคขั้นพื้นฐานและการตรวจหาอาคิเทกเจอร์สเมล การตรวจหาและแสดงผลวิธีการรีแฟคทอริง และการออกรายงานวิธีการรีแฟคทอริง

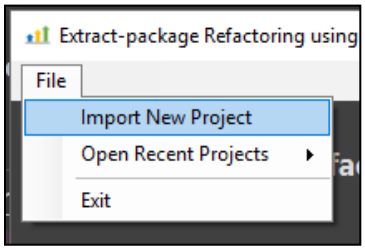
ก.1.การนำข้อมูลโปรเจคการทำงานหลักเข้าระบบ

เมื่อผู้ใช้งานเปิดระบบขึ้นมาแล้วจะแสดงหน้าจอเริ่มต้นสำหรับผู้ใช้งานดังรูปที่ ก.1 เพื่อให้ผู้ใช้งานเลือกโปรเจคที่ต้องการตรวจหา

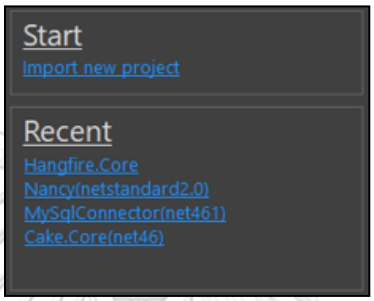


รูปที่ ก.1 หน้าจอเริ่มต้นสำหรับผู้ใช้งาน

ผู้ใช้งานสามารถเลือกนำเข้าข้อมูลโปรเจคใหม่ หรือเลือกโปรเจคที่เคยเปิดใช้งานล่าสุด โดยสามารถเลือกจากเมนูด้านบน ดังรูปที่ ก.2 หรือเลือกจากเมนูเส้นทางลัดในหน้าจอเริ่มต้น ดังรูปที่ ก.3

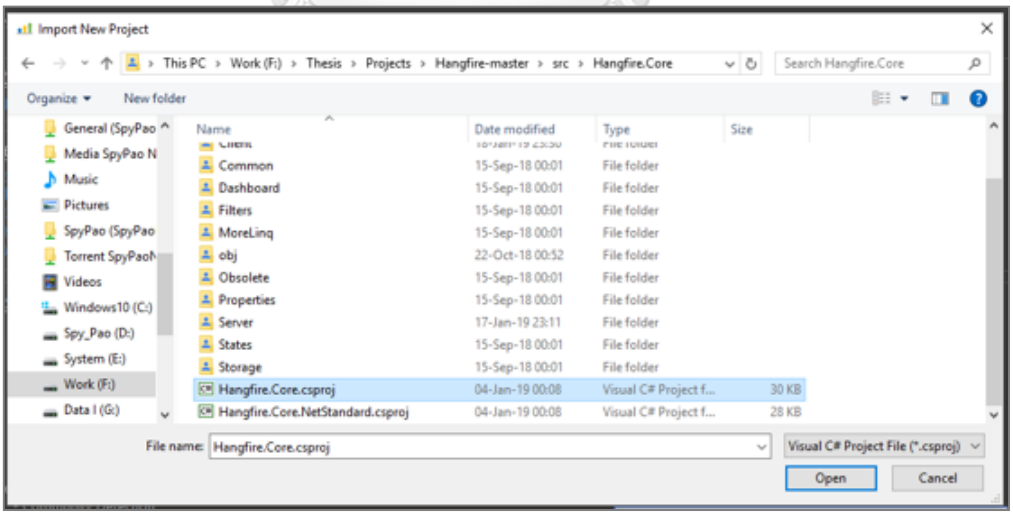


รูปที่ ก.2 เมนูสำหรับนำเข้าข้อมูลโปรเจคใหม่หรือเลือกโปรเจคล่าสุด



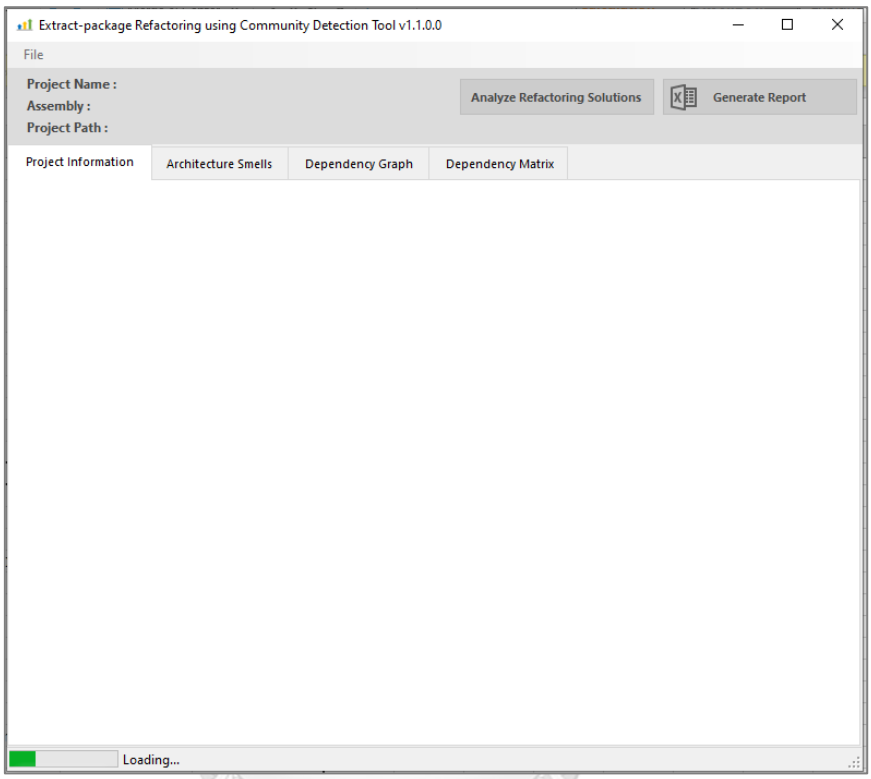
รูปที่ ก.3 เมนูเส้นทางลัดในหน้าจอเริ่มต้นสำหรับนำเข้าข้อมูลโปรเจคใหม่หรือเลือกโปรเจคล่าสุด

เมื่อผู้ใช้งานเลือกนำเข้าข้อมูลโปรเจคใหม่จะแสดงหน้าจอให้ผู้ใช้งานเลือกไฟล์โปรเจค โดยไฟล์โปรเจคจะมีนามสกุลคือ csproj โดยเลือกไฟล์และคลิกปุ่ม Open ดังรูปที่ ก.4



รูปที่ ก.4 หน้าต่างสำหรับผู้ใช้งานเลือกนำเข้าข้อมูลโปรเจคใหม่

หลังจากผู้ใช้งานเลือกนำเข้าข้อมูลโปรเจคใหม่ หรือเลือกโปรเจคล่าสุดแล้ว ระบบจะทำการอ่านข้อมูลโปรเจคเพื่อเตรียมผลวิเคราะห์พื้นฐานและตรวจหาอาคิเทกเจอร์สเมล โดยแสดงสถานะการทำงานในแถบสถานะในส่วนด้านล่างของเครื่องมือดังรูปที่ ก.5



รูปที่ ก.5 เครื่องมือแสดงสถานะการทำงานในแถบสถานะระหว่างการประมวลผล

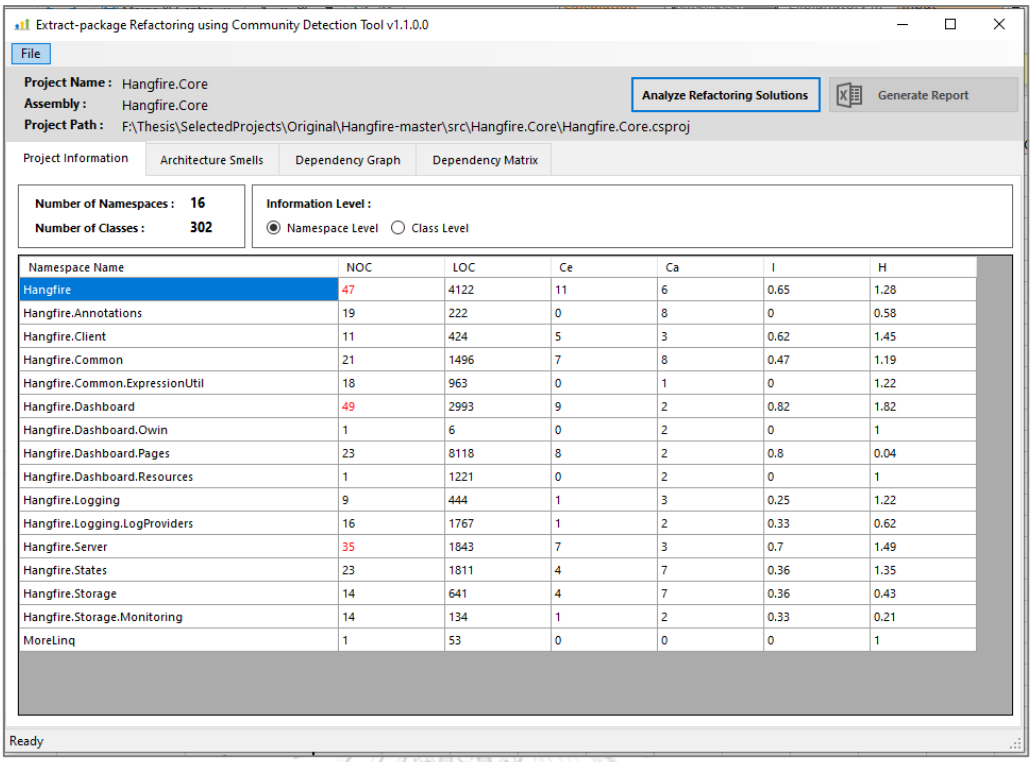
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ก.2.การแสดงผลวิเคราะห์โปรเจคขั้นพื้นฐานและการตรวจหาอาคิเทกเจอร์สเมล

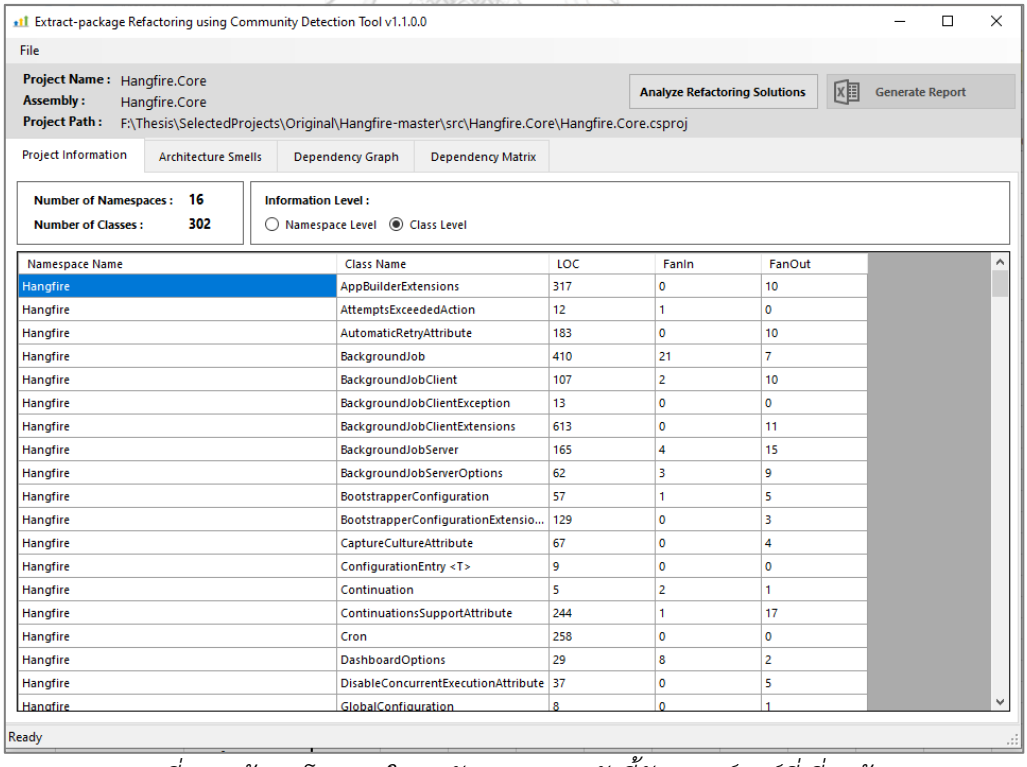
หลังจากการขั้นตอนการนำข้อมูลโปรเจคการทำงานหลักเข้าระบบ จะแสดงผลวิเคราะห์โปรเจคขั้นพื้นฐานและผลการตรวจหาอาคิเทกเจอร์สเมล โดยแบ่งออกเป็น 4 ส่วน ประกอบด้วย

(1) แถบข้อมูลโปรเจค (Project Information)

ในส่วนนี้จะแสดงข้อมูลโปรเจคทั่วไปประกอบด้วยจำนวนเนมสเปซในโปรเจคและจำนวนคลาสในโปรเจค นอกจากนี้จะแสดงผลในรายละเอียดย่อย 2 ส่วน ประกอบด้วย ข้อมูลในระดับเนมสเปซและค่าตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง ดังรูปที่ ก.6 และข้อมูลในระดับคลาสและค่าตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง ดังรูปที่ ก.7



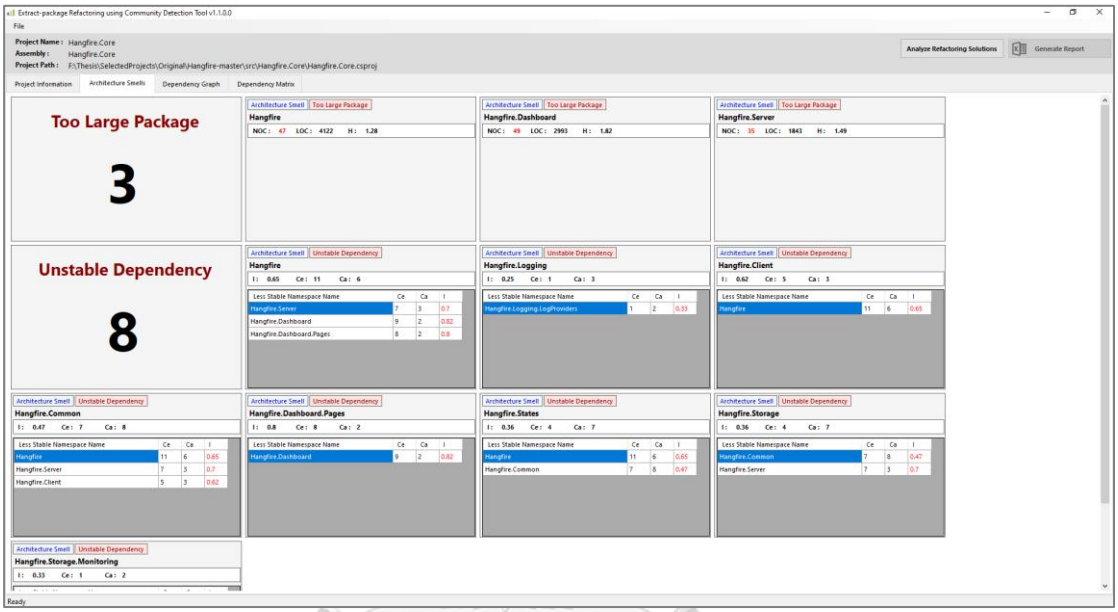
รูปที่ ก.6 ข้อมูลโปรเจคในระดับเนมสเปซและตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง



รูปที่ ก.7 ข้อมูลโปรเจคในระดับคลาสและตัวชี้วัดซอฟต์แวร์ที่เกี่ยวข้อง

(2) แถบผลการตรวจหาอาคิเทกเจอร์สเมล (Architecture Smells)

ในส่วนนี้จะแสดงผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี ในรูปแบบรายงานแดชบอร์ด (dashboard) ดังรูปที่ ก.8 โดยแสดงรายละเอียดจำนวนของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลดังกล่าว และค่าตัวชี้วัดซอฟต์แวร์ รวมถึงรายละเอียดที่อธิบายเหตุผลการเกิดอาคิเทกเจอร์สเมล

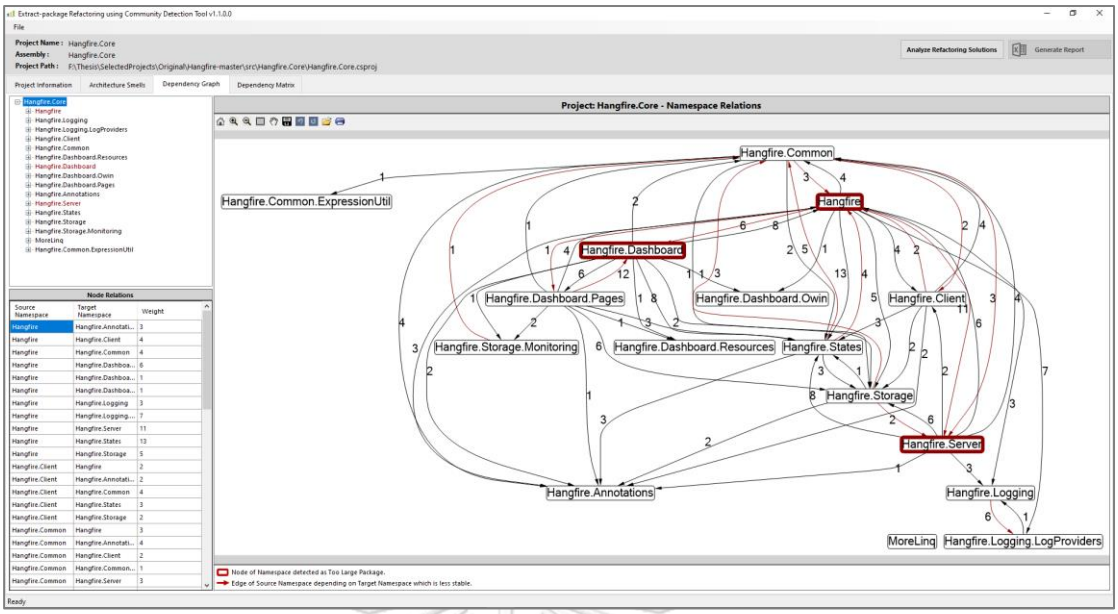


รูปที่ ก.8 ผลการตรวจหาอาคิเทกเจอร์สเมลในโปรเจค

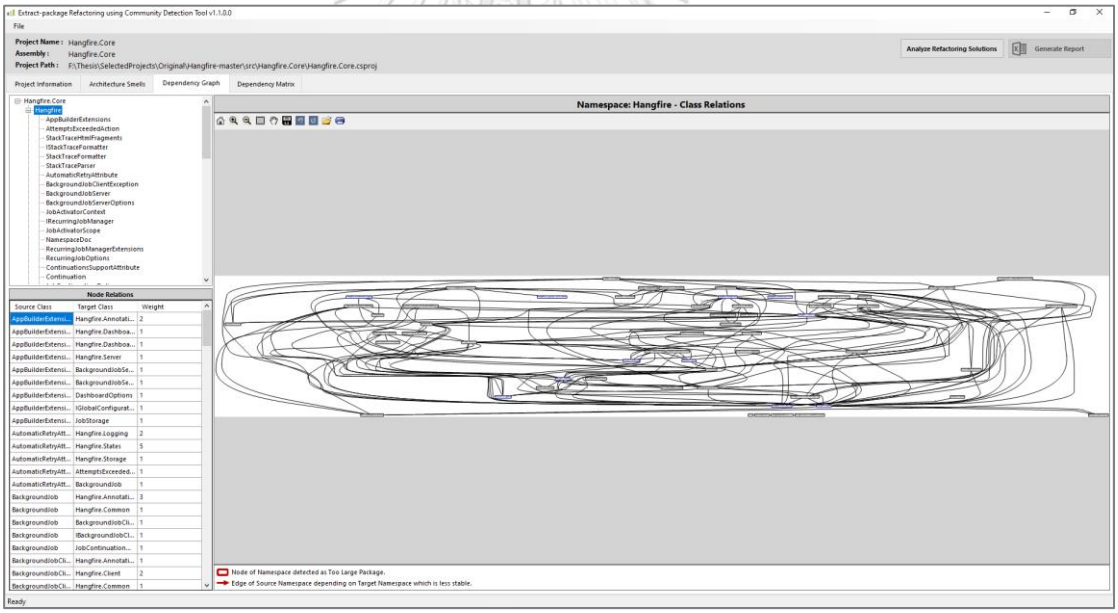
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

(3) แถบแสดงกราฟการพึ่งพา (Dependency Graph)

ในส่วนนี้จะแสดงกราฟการพึ่งพาและข้อมูลความสัมพันธ์ โดยจะแบ่งออกเป็น 2 ส่วน ประกอบด้วย กราฟการพึ่งพาระหว่างเนมสเปซภายในโปรเจค ดังรูปที่ ก.9 โดยสามารถเลือกที่ชื่อโปรเจค และกราฟการพึ่งพาระหว่างคลาสภายในโปรเจครวมถึงความสัมพันธ์กับเนมสเปซข้างเคียง ดังรูปที่ ก.10 โดยเลือกที่ชื่อเนมสเปซ



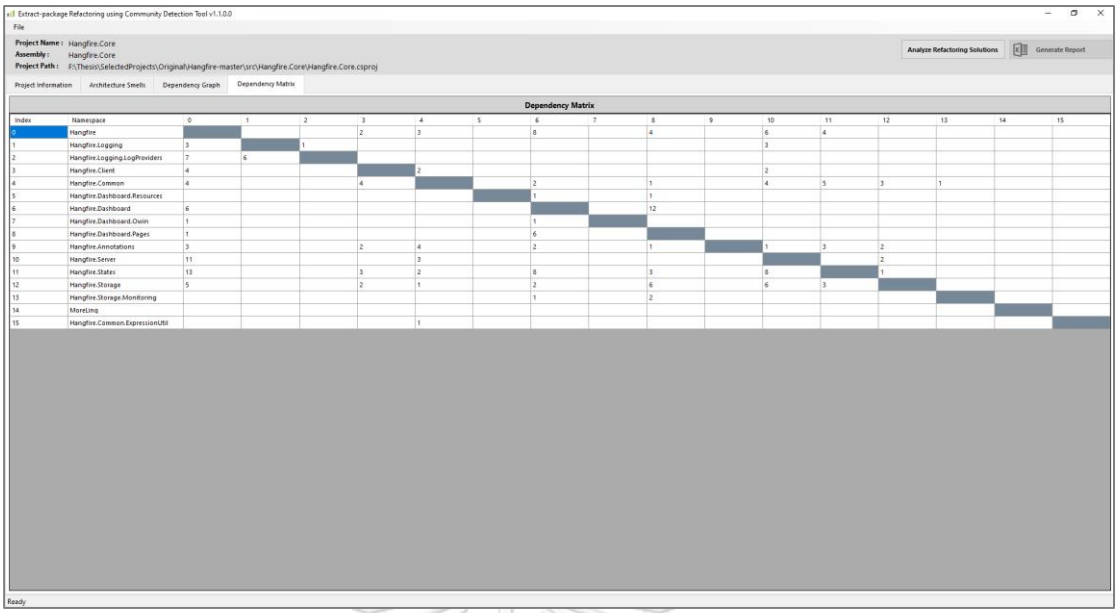
รูปที่ ก.9 กราฟการพึ่งพาระหว่างเนมสเปซภายในโปรเจค



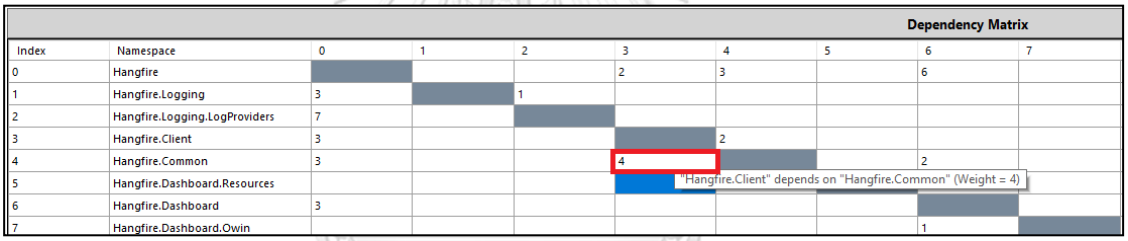
รูปที่ ก.10 กราฟการพึ่งพาระหว่างคลาสภายในโปรเจครวมถึงความสัมพันธ์กับเนมสเปซข้างเคียง

(4) แถบแสดงเมทริกซ์การพึ่งพา (Dependency Matrix)

ในส่วนนี้จะแสดงข้อมูลความสัมพันธ์ของเนมสเปซภายในโปรเจคในรูปแบบของเมทริกซ์ ดังรูปที่ ก.11 โดยค่าที่แสดงในตารางบ่งบองถึงทิศทางและน้ำหนักของความสัมพันธ์ดังรูปที่ ก.12



รูปที่ ก.11 เมทริกซ์การพึ่งพาของเนมสเปซภายในโปรเจค



Dependency Matrix

Index	Namespace	0	1	2	3	4	5	6	7
0	Hangfire				2	3		6	
1	Hangfire.Logging	3		1					
2	Hangfire.Logging.LogProviders	7							
3	Hangfire.Client	3				2			
4	Hangfire.Common	3			4			2	
5	Hangfire.Dashboard.Resources								
6	Hangfire.Dashboard	3							
7	Hangfire.Dashboard.Owin							1	

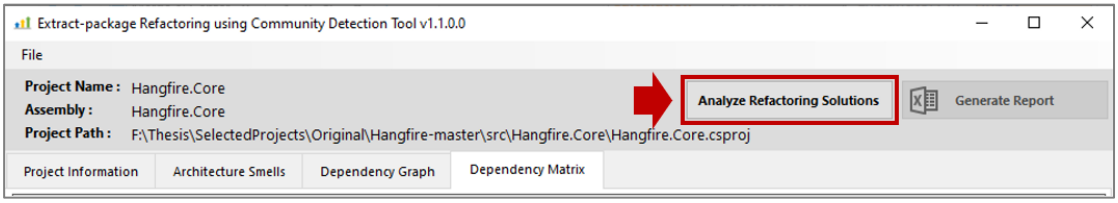
"Hangfire.Client" depends on "Hangfire.Common" (Weight = 4)

รูปที่ ก.12 แสดงความหมายของความสัมพันธ์ระหว่างเนมสเปซในเมทริกซ์การพึ่งพา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

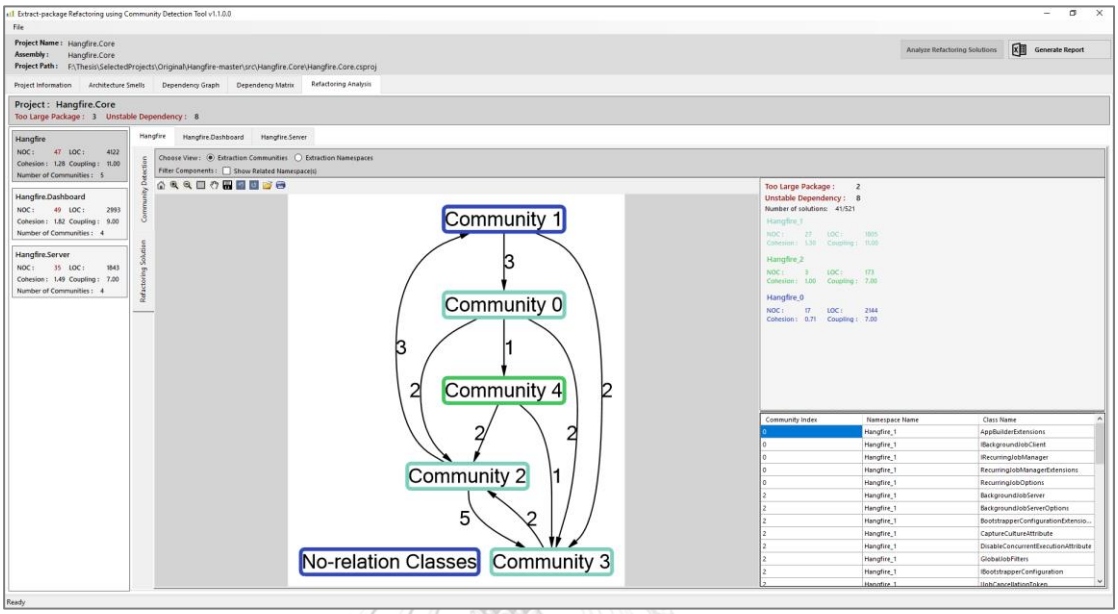
ก.3.การตรวจหาและแสดงผลวิธีการรีแฟคทอริง

ผู้ใช้งานสามารถตรวจหาวิธีการรีแฟคทอริงโดยสามารถคลิกปุ่ม Analyze Refactoring Solutions ดังรูปที่ ก.13 เพื่อให้ระบบเริ่มทำการตรวจหาวิธีการรีแฟคทอริงสำหรับแพ็กเกจขนาดใหญ่เกิน



รูปที่ ก.13 แสดงปุ่มสำหรับการเริ่มตรวจหาวิธีการรีแฟคทอริง

หลังจากระบบตรวจหาวิธีการรีแฟคทอริงในเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินเสร็จสิ้นจะแสดงแถบผลวิเคราะห์การรีแฟคทอริง (Refactoring Analysis) ดังรูปที่ ก.14

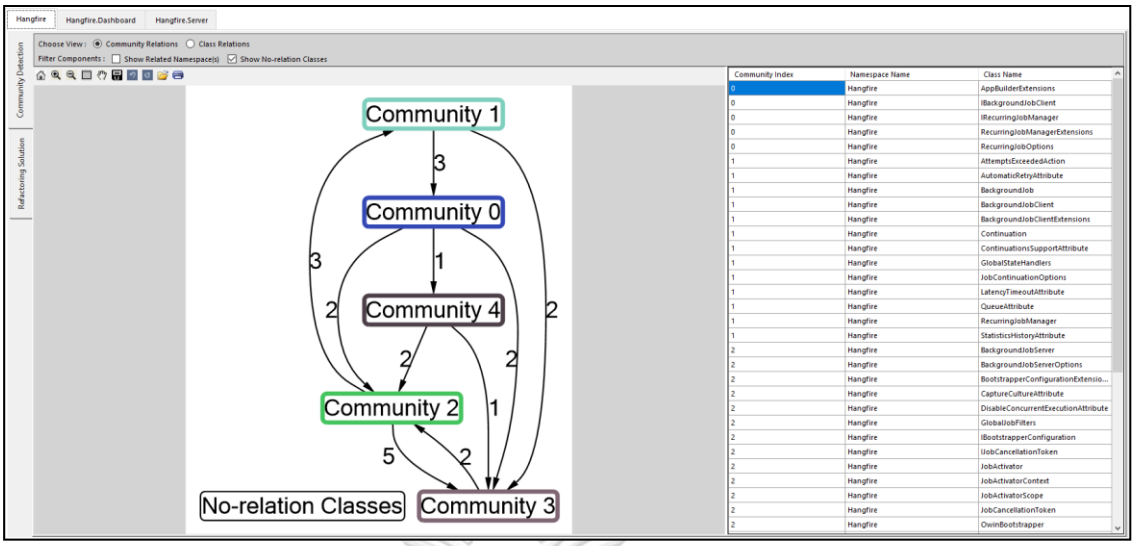


รูปที่ ก.14 แสดงแถบผลวิเคราะห์การรีแฟคทอริง

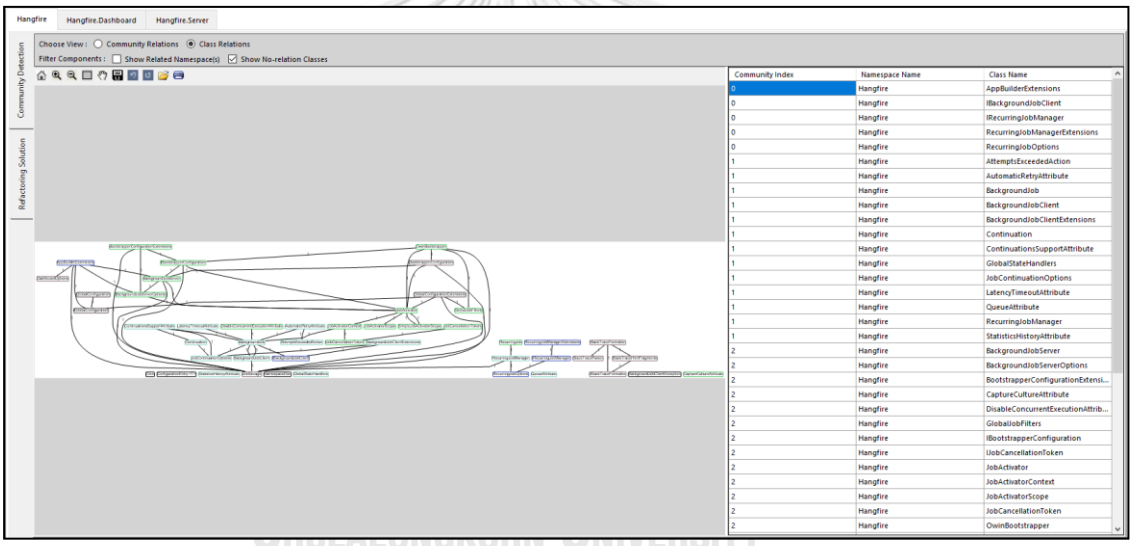
โดยแถบผลวิเคราะห์การรีแฟคทอริงนั้นจะแสดงรายการของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลในรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยเมื่อผู้ใช้งานเลือกที่เนมสเปซจะแสดงข้อมูลของเนมสเปซดังกล่าวโดยประกอบด้วย 2 แถบย่อย คือ แถบผลการตรวจหาคอมมิวนิตี (Community Detection) และแถบแสดงวิธีการรีแฟคทอริง (Refactoring Solution) ถ้าเนมสเปซดังกล่าวสามารถตรวจหาวิธีการรีแฟคทอริงที่ไม่ทำให้เกิดอาคิเทกเจอร์สเมลในรูปแบบแพ็กเกจขนาดใหญ่เกินและอันสเตเบิลดีเพนเดนซีเพิ่มมากขึ้น รวมถึงมีค่าความเชื่อมแน่นที่สูงขึ้น ในกรณีตรวจไม่พบวิธีการรีแฟคทอริงจะไม่แสดงแถบย่อยในส่วนนี้

แถบผลการตรวจหาคอมมิวนิตีจะแสดงกราฟการพึ่งพาของคอมมิวนิตีต่าง ๆ ที่เป็นตัวแทนของคลาสต่าง ๆ ในคอมมิวนิตี โดยแสดงรายละเอียดคลาสในคอมมิวนิตีต่าง ๆ ในตารางด้านขวามือ โดยผู้ใช้งานสามารถปรับเปลี่ยนมุมมองของกราฟการพึ่งพานี้ได้ โดยแสดงความสัมพันธ์ของคอมมิวนิตี ดังรูปที่ ก.15 หรือความสัมพันธ์ของคลาสโดยแต่ละคลาสจะแสดงสีตามคอมมิวนิตีที่ถูกจัดกลุ่มไว้ดังรูปที่ ก.16 นอกจากนี้ยังสามารถเลือกแสดงหรือไม่แสดงเนมสเปซข้างเคียง และคลาสที่ไม่มีความสัมพันธ์ใดๆ ในกราฟการพึ่งพาได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

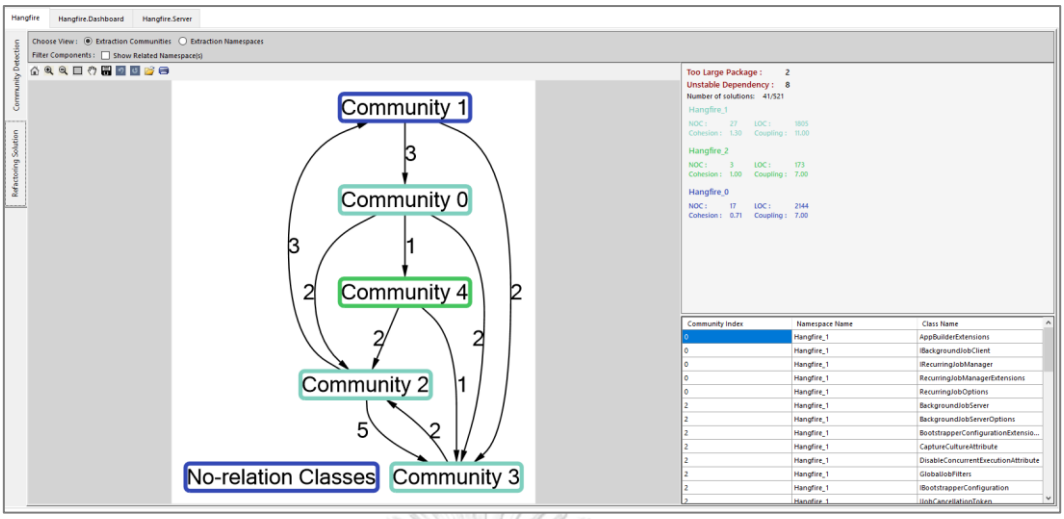


รูปที่ ก.15 แถบผลการตรวจหาคอมมิวนิตีโดยแสดงความสัมพันธ์ของคอมมิวนิตี

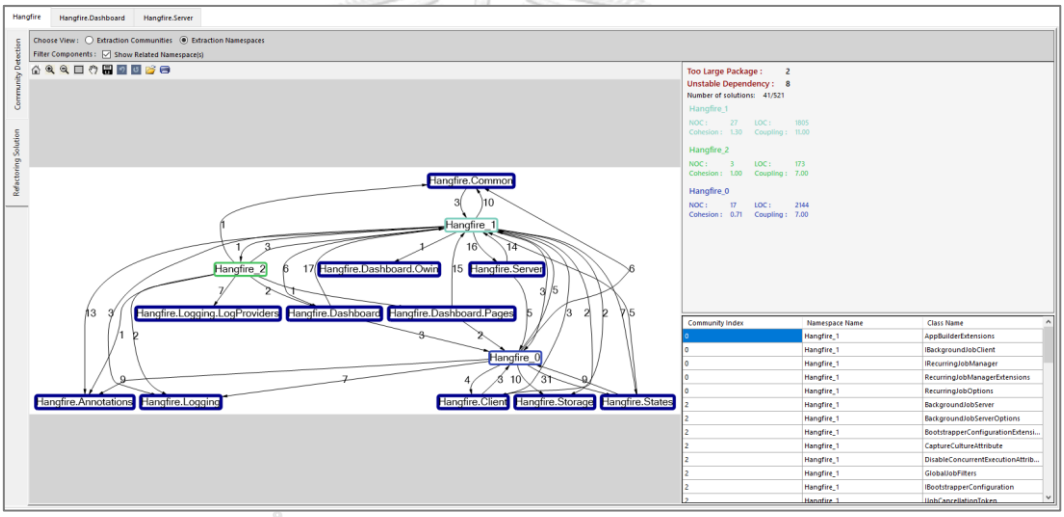


รูปที่ ก.16 แถบผลการตรวจหาคอมมิวนิตีโดยแสดงความสัมพันธ์ของคลาส

แถบแสดงวิธีการรีแฟคทอริงจะแสดงความสัมพันธ์ของคอมมิวนิตีต่าง ๆ โดยแสดงสีที่บ่งบองถึงเนมสเปซที่ถูกแยกออกมา ดังรูปที่ ก.17 หรือแสดงความสัมพันธ์ของเนมสเปซที่ถูกแยกออกมา ดังรูปที่ ก.18 โดยรายละเอียดการแยกเนมสเปซของคอมมิวนิตีต่าง ๆ รวมถึงในระดับคลาส ถูกแสดงรายละเอียดในส่วนขวามือ นอกจากนี้แสดงค่าตัวชี้วัดที่เกี่ยวข้องของเนมสเปซที่ถูกแยกออกมา รวมถึงผลการตรวจหาอาคิเทกเจอร์สเมลในรูปแบบแพ็กเกจขนาดใหญ่เกินและอันสเตเบิลดีเพนเดนซีในโปรเจคหลังจากการแยกเนมสเปซใหม่เกิดขึ้น ในส่วนขวามือดังรูปที่ ก.19



รูปที่ ก.17 แถบแสดงวิธีการรีแฟคทอริงโดยแสดงความสัมพันธ์ของคอมมิวนิตีในเนมสเปซใหม่

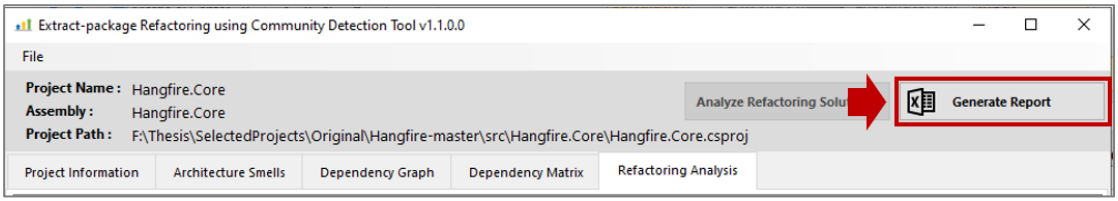


รูปที่ ก.18 แถบแสดงวิธีการรีแฟคทอริงโดยแสดงความสัมพันธ์ของเนมสเปซที่ถูกแยกออกมาใหม่

รูปที่ ก.19 แสดงผลการตรวจหาอาคิเทกเจอร์สเมลและค่าตัวชี้วัดหลังจากการแยกเนมสเปซ

ก.4.การออกรายงานวิธีการรีแฟคทอริง

ผู้ใช้งานสามารถออกรายงานวิธีการรีแฟคทอริงได้หลังจากขั้นตอนการตรวจหาและแสดงผลวิธีการรีแฟคทอริง โดยกดที่ปุ่ม Generate Report ดังรูปที่ ก.20 จากนั้นระบบจะถามตำแหน่งที่ต้องการจะบันทึกรายงานนี้



รูปที่ ก.20 การออกรายงานวิธีการรีแฟคทอริง

หลังจากระบบสร้างรายงานสำเร็จจะสอบถามผู้ใช้งานว่าต้องการแสดงรายงานดังกล่าวเลยหรือไม่ หากผู้ใช้ตอบตกลงจะเปิดรายงานในรูปแบบของไฟล์เอ็กซ์เซล โดยรายละเอียดของรายงานประกอบด้วย 3 ส่วนดังนี้

(1) ส่วนข้อมูลโปรเจค (Project Information)

ในส่วนนี้จะแสดงข้อมูลโปรเจคเหมือนกับแถบข้อมูลโปรเจค โดยแสดงตัวอย่างดังรูปที่ ก.21

	A	B	C	D	E	F	G
1	**Project Name :**	Hangfire.Core					
2	**Assembly Name :**	Hangfire.Core					
3	**Project Path :**	F:\Thesis\Projects\Hangfire-master\src\Hangfire.Core\Hangfire.Core.csproj					
4							
5	**Number of Namespaces :**	16					
6	**Number of Classes :**	302					
7							
8	**Namespace Information**						
9	**Namespace**	**NOC**	**LOC**	**Ce**	**Ca**	**I**	**H**
10	Hangfire	47	4122	11	6	0.65	1.28
11	Hangfire.Annotations	19	222	0	8	0	0.58
12	Hangfire.Client	11	424	5	3	0.62	1.45
13	Hangfire.Common	21	1496	7	8	0.47	1.19
14	Hangfire.Common.ExpressionUtil	18	963	0	1	0	1.22
15	Hangfire.Dashboard	49	2993	9	2	0.82	1.82
16	Hangfire.Dashboard.Owin	1	6	0	2	0	1
17	Hangfire.Dashboard.Pages	23	8118	8	2	0.8	0.04
18	Hangfire.Dashboard.Resources	1	1221	0	2	0	1
19	Hangfire.Logging	9	444	1	3	0.25	1.22
20	Hangfire.Logging.LogProviders	16	1767	1	2	0.33	0.62
21	Hangfire.Server	35	1843	7	3	0.7	1.49
22	Hangfire.States	23	1811	4	7	0.36	1.35
23	Hangfire.Storage	14	641	4	7	0.36	0.43
24	Hangfire.Storage.Monitoring	14	134	1	2	0.33	0.21
25	MoreLinq	1	53	0	0	0	1
26							
27	**Class Information**						
28	**Namespace**	**Class Name**	**LOC**	**Fan In**	**Fan Out**		
29	Hangfire	AppBuilderExtensions	317	0	10		
30	Hangfire	AttemptsExceededAction	12	1	0		
31	Hangfire	AutomaticRetryAttribute	183	0	10		
32	Hangfire	BackgroundJob	410	21	7		
33	Hangfire	BackgroundJobClient	107	2	10		
34	Hangfire	BackgroundJobClientException	13	0	0		
35	Hangfire	BackgroundJobClientExtensions	613	0	11		
36	Hangfire	BackgroundJobServer	165	4	15		
37	Hangfire	BackgroundJobServerOptions	62	3	9		
38	Hangfire	BootstrapperConfiguration	57	1	5		

Project Information | Architecture Smells | Hangfire | Hangfire.Dashboard | Hangfire.Server

รูปที่ ก.21 รายงานส่วนข้อมูลโปรเจค

(2) ส่วนผลการตรวจหาอาคิเทกเจอร์สเมล (Architecture Smells)

ในส่วนนี้จะแสดงผลการตรวจหาอาคิเทกเจอร์สเมลเหมือนกับแถบผลการตรวจหาอาคิเทกเจอร์สเมล โดยแสดงตัวอย่างดังรูปที่ ก.22

	A	B	C	D	E
1	**Project Name :**	Hangfire.Core			
2	**Assembly Name :**	Hangfire.Core			
3	**Project Path :**	F:\Thesis\Projects\Hangfire-master\src\Hangfire.Core\Hangfire.Core.csproj			
4					
5	**Number of Too Large Package :**	3			
6	**Number of Unstable Dependency :**	8			
7					
8	**Too Large Package**				
9	**Namespace**	**NOC**	**LOC**	**H**	
10	Hangfire	47	4122	1.28	
11	Hangfire.Dashboard	49	2993	1.82	
12	Hangfire.Server	35	1843	1.49	
13					
14	**Unstable Dependency**				
15	**Namespace**	**I**	**Ce**	**Ca**	**Less Stable Namespace**
16	Hangfire	0.65	11	6	Hangfire.Server, Hangfire.Dashboard, Hangfire.Dashboard.Pages
17	Hangfire.Logging	0.25	1	3	Hangfire.Logging.LogProviders
18	Hangfire.Client	0.62	5	3	Hangfire
19	Hangfire.Common	0.47	7	8	Hangfire, Hangfire.Server, Hangfire.Client
20	Hangfire.Dashboard.Pages	0.8	8	2	Hangfire.Dashboard
21	Hangfire.States	0.36	4	7	Hangfire, Hangfire.Common
22	Hangfire.Storage	0.36	4	7	Hangfire.Common, Hangfire.Server
23	Hangfire.Storage.Monitoring	0.33	1	2	Hangfire.Common
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					

Project Information | Architecture Smells | Hangfire | Hangfire.Dashboard | Hangfire.Server

รูปที่ ก.22 รายงานส่วนผลการตรวจหาอาคิเทกเจอร์สเมล

(3) ส่วนแสดงวิธีการรีแฟคทอริงของเนมสเปซต่าง ๆ

ในส่วนนี้จะแสดงรายละเอียดการรีแฟคทอริงของเนมสเปซต่าง ๆ โดยจะแสดงแยกรายละเอียดของแต่ละเนมสเปซ โดยเปรียบเทียบก่อนการเปลี่ยนแปลง หลังจากการรีแฟคทอริง โดยแสดงตัวอย่างดังรูปที่ ก.23

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

	A	B	C	D	E
1	**Project :**	Hangfire.Core			
2	**Namespace :**	Hangfire			
3	**Number of Communities :**	5			
4					
5		**Original**	**Solution**		
6	**Number of Too Large Package :**	3		2	
7	**Number of Unstable Dependency :**	8		8	
8					
9		**Original**	**Solution**		
10		Hangfire	Hangfire_1	Hangfire_2	Hangfire_0
11	**NOC :**	47	27	3	17
12	**LOC :**	4122	1805	173	2144
13	**H :**	1.28	1.3	1	0.71
14					
15	**Namespace**	**Class Name**	**Community Index**	**Solution**	
16	Hangfire	AppBuilderExtensions	0	Hangfire_1	
17	Hangfire	IBackgroundJobClient	0	Hangfire_1	
18	Hangfire	IRecurringJobManager	0	Hangfire_1	
19	Hangfire	RecurringJobManagerExtensions	0	Hangfire_1	
20	Hangfire	RecurringJobOptions	0	Hangfire_1	
21	Hangfire	AttemptsExceededAction	1	Hangfire_0	
22	Hangfire	AutomaticRetryAttribute	1	Hangfire_0	
23	Hangfire	BackgroundJob	1	Hangfire_0	
24	Hangfire	BackgroundJobClient	1	Hangfire_0	
25	Hangfire	BackgroundJobClientExtensions	1	Hangfire_0	
26	Hangfire	Continuation	1	Hangfire_0	
27	Hangfire	ContinuationsSupportAttribute	1	Hangfire_0	
28	Hangfire	GlobalStateHandlers	1	Hangfire_0	
29	Hangfire	JobContinuationOptions	1	Hangfire_0	
30	Hangfire	LatencyTimeoutAttribute	1	Hangfire_0	
31	Hangfire	QueueAttribute	1	Hangfire_0	
32	Hangfire	RecurringJobManager	1	Hangfire_0	
33	Hangfire	StatisticsHistoryAttribute	1	Hangfire_0	
34	Hangfire	BackgroundJobServer	2	Hangfire_1	
35	Hangfire	BackgroundJobServerOptions	2	Hangfire_1	
36	Hangfire	BootstrapperConfigurationExtensions	2	Hangfire_1	
37	Hangfire	CaptureCultureAttribute	2	Hangfire_1	

Project Information | Architecture Smells | **Hangfire** | Hangfire.Dashboard | Hangfire.Server

CHULALONGKORN UNIVERSITY

รูปที่ ก.23 รายงานส่วนแสดงวิธีการรีแฟคทอริงของเนมสเปซ

ภาคผนวก ข
การทดสอบและประเมินผลความสามารถของเครื่องมือเพิ่มเติม

ในภาคผนวนนี้จะแสดงรายละเอียดเพิ่มเติมจาก บทที่ 5 การทดสอบและประเมินผลความสามารถของเครื่องมือ ในรายละเอียดเชิงลึก โดยมีรายละเอียดต่าง ๆ ดังนี้

ข.1. ผลการตรวจหาอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์เพิ่มเติม

จากหัวข้อ 5.1.2. การตรวจหาอาคิเทกเจอร์สเมลด้วยเครื่องมือพาณิชย์ ในโปรเจคที่เลือกใช้ในงานวิจัย โดยในส่วนนี้จะแสดงรายละเอียดของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน โดยแสดงจำนวนคลาสที่มีในเนมสเปซ ดังตารางที่ ข.1 และรายละเอียดเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี โดยแสดงเนมสเปซที่พึ่งพาและมีความเสถียรภาพน้อยกว่า ดังตารางที่ ข.2

ตารางที่ ข.1 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินด้วยเครื่องมือพาณิชย์

ชื่อโปรเจค	ชื่อเนมสเปซ	จำนวนคลาส
Hyperion	Hyperion.Internal	74
Hyperion	Hyperion.ValueSerializers	32
Hangfire.Core	Hangfire	47
Hangfire.Core	Hangfire.Dashboard	49
Hangfire.Core	Hangfire.Server	35
Nancy	Nancy	96
Nancy	Nancy.Diagnostics	37

ตารางที่ ข.2 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีด้วยเครื่องมือพาณิชย์

ชื่อโปรเจค	ชื่อเนมสเปซ	เนมสเปซที่พึ่งพา ที่เสถียรภาพน้อยกว่า
Hyperion	Hyperion	Hyperion.SerializerFactories
Hyperion	Hyperion.Compilation	Hyperion.Extensions
Hyperion	Hyperion.Extensions	Hyperion

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Hyperion	Hyperion.ValueSerializers	Hyperion Hyperion.Extensions Hyperion.Compilation
Hangfire.Core	Hangfire	Hangfire.Server Hangfire.Dashboard.Pages
Hangfire.Core	Hangfire.Logging	Hangfire.Logging.LogProviders
Hangfire.Core	Hangfire.Client	Hangfire
Hangfire.Core	Hangfire.Common	Hangfire Hangfire.Server Hangfire.Storage Hangfire.Client Hangfire.States
Hangfire.Core	Hangfire.Dashboard	Hangfire Hangfire.Dashboard.Pages
Hangfire.Core	Hangfire.States	Hangfire Hangfire.Storage Hangfire.Server
Hangfire.Core	Hangfire.Storage	Hangfire.Server
Nancy	Nancy	Nancy.Bootstrapper Nancy.Diagnostics Nancy.Tinyloc Nancy.Culture Nancy.Responses.Negotiation Nancy.Conventions Nancy.Responses Nancy.ModelBinding Nancy.Routing Nancy.ViewEngines Nancy.Json Nancy.ErrorHandling

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

		Nancy.Session
Nancy	Nancy.Configuration	Nancy
Nancy	Nancy.Conventions	Nancy.Session Nancy.Bootstrapper Nancy.Diagnostics
Nancy	Nancy.Cookies	Nancy.Helpers
Nancy	Nancy.Culture	Nancy.Conventions
Nancy	Nancy.Diagnostics	Nancy.Bootstrapper Nancy.ViewEngines.SuperSimple ViewEngine
Nancy	Nancy.Helpers	Nancy
Nancy	Nancy.IO	Nancy
Nancy	Nancy.Json	Nancy.Json.Simple Nancy.Json.Converters
Nancy	Nancy.Localization	Nancy
Nancy	Nancy.ModelBinding	Nancy.ModelBinding.DefaultBody Deserializers Nancy.ModelBinding.DefaultConverters Nancy.Responses.Negotiation
Nancy	Nancy.Responses.Negotiation	Nancy.Conventions Nancy.Responses Nancy.ViewEngines
Nancy	Nancy.Routing.Constraints	Nancy.Routing.Trie
Nancy	Nancy.Routing	Nancy.Diagnostics Nancy.ErrorHandling
Nancy	Nancy.Routing.Trie	Nancy.Routing Nancy.Routing.Trie.Nodes
Nancy	Nancy.Security	Nancy.Bootstrapper
Nancy	Nancy.Session	Nancy.Bootstrapper
Nancy	Nancy.Validation	Nancy

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Nancy	Nancy.ViewEngines	Nancy.Security Nancy.Conventions Nancy.Bootstrapper

ข.2. ผลการประเมินผลการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนาเพิ่มเติม

จากหัวข้อ 5.2 ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลของเครื่องมือที่พัฒนา โดยในส่วนนี้จะแสดงรายละเอียดเพิ่มเติม ประกอบด้วย การประเมินผลการเตรียมโครงสร้างข้อมูลของโปรเจค การประเมินผลการเตรียมความสัมพันธ์ของข้อมูลระหว่างคลาส การประเมินผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี และการผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินและรูปแบบอันสเตเบิลดีเพนเดนซี

ข.2.1. ประเมินผลการเตรียมโครงสร้างข้อมูลโปรเจค

โดยผลการประเมินโครงสร้างข้อมูลโปรเจคจากเครื่องมือที่พัฒนา โดยนำมาวิเคราะห์กับโปรเจคที่ใช้ทดลองในงานวิจัยโดยผลการวิเคราะห์ข้อมูลระดับเนมสเปซของแต่ละโปรเจคแสดงดังตารางที่ ข.3 ตารางที่ ข.4 และตารางที่ ข.5 โดยในแต่ละตารางแสดงค่าตัวชี้วัดซอฟต์แวร์ในระดับเนมสเปซ โดยประกอบด้วย จำนวนคลาส (NOC) จำนวนบรรทัด (LOC) ค่าเอฟเฟอเรนท์คัปปลิง (Ce) ค่าแอฟเฟอเรนท์คัปปลิง (Ca) ค่าความไม่แน่นอน (I - Instability) และค่าความเชื่อมแน่น (H – Cohesion)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.3 ผลการวิเคราะห์ข้อมูลระดับเนมสเปซของโปรเจค Hyperion จากเครื่องมือที่พัฒนา

ชื่อเนมสเปซ	NOC	LOC	Ce	Ca	I	H
Hyperion	13	934	5	3	0.62	1.46
Hyperion.Compilation	21	861	1	2	0.33	2.52
Hyperion.Extensions	6	555	2	4	0.33	0.83
Hyperion.Internal	**74**	539	0	2	0	0.53
Hyperion.SerializerFactories	19	1117	3	1	0.75	0.05
Hyperion.ValueSerializers	**32**	1154	4	3	0.57	0.38

ตารางที่ ข.4 ผลการวิเคราะห์ข้อมูลระดับเนมสเปซของโปรเจค Hangfire.Core จากเครื่องมือที่พัฒนา

ชื่อเนมสเปซ	NOC	LOC	Ce	Ca	I	H
Hangfire	**47**	4122	11	6	0.65	1.28
Hangfire.Annotations	19	222	0	8	0	0.58
Hangfire.Client	11	424	5	3	0.62	1.45
Hangfire.Common	21	1496	7	8	0.47	1.19
Hangfire.Common.ExpressionUtil	18	963	0	1	0	1.22
Hangfire.Dashboard	**49**	2993	9	2	0.82	1.82
Hangfire.Dashboard.Owin	1	6	0	2	0	1
Hangfire.Dashboard.Pages	23	8118	8	2	0.8	0.04
Hangfire.Dashboard.Resources	1	1221	0	2	0	1
Hangfire.Logging	9	444	1	3	0.25	1.22
Hangfire.Logging.LogProviders	16	1767	1	2	0.33	0.62
Hangfire.Server	**35**	1843	7	3	0.7	1.49
Hangfire.States	23	1811	4	7	0.36	1.35
Hangfire.Storage	14	641	4	7	0.36	0.43
Hangfire.Storage.Monitoring	14	134	1	2	0.33	0.21
MoreLinq	1	53	0	0	0	1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ ข.5 ผลการวิเคราะห์ข้อมูลระดับเนมสเปซของโปรเจค Nancy จากเครื่องมือที่พัฒนา

ชื่อเนมสเปซ	NOC	LOC	Ce	Ca	I	H
Nancy	**96**	10980	19	27	0.41	1.39
Nancy.Bootstrapper	20	1906	17	9	0.65	1.95
Nancy.Configuration	9	303	0	14	0	0.78
Nancy.Conventions	18	1010	7	6	0.54	1.17
Nancy.Cookies	2	184	1	7	0.12	0.5
Nancy.Cryptography	11	314	0	4	0	0.82
Nancy.Culture	2	42	3	4	0.43	0.5
Nancy.Diagnostics	**37**	1691	18	6	0.75	1
Nancy.Diagnostics.Modules	6	325	7	0	1	0.33
Nancy.ErrorHandling	4	176	3	3	0.5	0.5
Nancy.Extensions	10	736	5	0	1	0.1
Nancy.Helpers	8	1816	1	12	0.08	0.38
Nancy.IO	2	741	1	2	0.33	0.5
Nancy.Json	8	339	4	5	0.44	1.5
Nancy.Json.Converters	2	117	1	1	0.5	0.5
Nancy.Json.Simple	11	2207	3	1	0.75	1.18
Nancy.Localization	4	254	1	4	0.2	0.75
Nancy.ModelBinding	18	1360	3	6	0.33	1.44
Nancy.ModelBinding.DefaultBodyDeserializers	2	87	5	0	1	0.5
Nancy.ModelBinding.DefaultConverters	4	192	2	0	1	0.25
Nancy.Owin	5	304	5	0	1	1.2
Nancy.Responses	15	681	7	8	0.47	0.2
Nancy.Responses.Negotiation	15	1189	7	9	0.44	2.07
Nancy.Routing	29	1083	10	6	0.62	1.14
Nancy.Routing.Constraints	18	606	1	4	0.2	0.06
Nancy.Routing.Trie	8	350	4	5	0.44	0.62
Nancy.Routing.Trie.Nodes	11	837	4	1	0.8	1
Nancy.Security	12	733	5	3	0.62	1

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Nancy.Session	5	459	5	2	0.71	0.8
Nancy.Tinyloc	29	4082	1	2	0.33	2.52
Nancy.Validation	11	419	1	3	0.25	1.18
Nancy.Validation.Rules	6	120	0	0	0	0.33
Nancy.ViewEngines	28	1442	8	8	0.5	1.43
Nancy.ViewEngines.Razor	1	18	0	0	0	1
Nancy.ViewEngines.SuperSimpleViewEngine	6	825	4	1	0.8	1.33
Nancy.Xml	3	63	1	1	0.5	0.67
Reflection	3	563	0	0	0	0.33

ข.2.2. ประเมินผลการเตรียมความสัมพันธ์ของข้อมูลระหว่างคลาส

โดยผลการประเมินผลการเตรียมความสัมพันธ์ของข้อมูลระหว่างคลาส โดยเครื่องมือที่พัฒนานำความสัมพันธ์ในระดับคลาสมาสร้างความสัมพันธ์ในระดับเนมสเปซ โดยการจัดเก็บโครงสร้างความสัมพันธ์ในระดับสามารถแสดงจัดเก็บในรูปแบบเมทริกซ์การพึ่งพา (dependency matrix) ดังแสดงตัวอย่างในตารางที่ ข.6 และตารางที่ ข.7 และตารางที่ ข.8 โดยในแต่ละตารางจะแสดงความสัมพันธ์ของแต่ละเนมสเปซในโปรเจค โดยแสดงทิศทางความสัมพันธ์และปริมาณความสัมพันธ์ระหว่างคลาสในแต่ละเนมสเปซที่มีการพึ่งพากัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.6 เมทริกซ์การพึ่งพาของเนมสเปซในโปรเจค Hyperion จากเครื่องมือที่พัฒนา

#	ชื่อเนมสเปซ	0	1	2	3	4	5
0	Hyperion	-		8		6	6
1	Hyperion.Compilation	1	-				1
2	Hyperion.Extensions	2	2	-		1	1
3	Hyperion.Internal	1			-		1
4	Hyperion.SerializerFactories	19				-	
5	Hyperion.ValueSerializers	27		4		7	-

ตารางที่ ข.7 เมทริกซ์การพึ่งพาของเนมสเปซในโปรเจค Hangfire.Core จากเครื่องมือที่พัฒนา

#	ชื่อเนมสเปซ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	Hangfire	-			2	3		8		4		6	4				
1	Hangfire.Logging	3	-	1								3					
2	Hangfire.Logging.LogProviders	7	6	-													
3	Hangfire.Client	4			-	2						2					
4	Hangfire.Common	4			4	-		2		1		4	5	3	1		
5	Hangfire.Dashboard.Resources						-	1		1							
6	Hangfire.Dashboard	6						-		12							
7	Hangfire.Dashboard.Owin	1						1	-								
8	Hangfire.Dashboard.Pages	1						6		-							
9	Hangfire.Annotations	3			2	4		2		1	-	1	3	2			
10	Hangfire.Server	11				3						-		2			
11	Hangfire.States	13			3	2		8		3		8	-	1			
12	Hangfire.Storage	5			2	1		2		6		6	3	-			
13	Hangfire.Storage.Monitoring							1		2					-		
14	MoreLinq															-	
15	Hangfire.Common.ExpressionUtil					1											-

ตารางที่ ข.8 เมทริกซ์การพึ่งพาของเนมสเปซในโปรเจค Nancy จากเครื่องมือที่พัฒนา

#	ชื่อเนมสเปซ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	Nancy	-	35		9			2	20	11	5	12	5	1		1		2	2	6	1	1	8	8	10		16	2	3	9	8	1	2		5	2		
1	Nancy.Bootstrapper	9	-		1				2	2													2							1	1				1	3		
2	Nancy.Configuration	2	6	-				1	2	1	1					1				1	1			1	1		1								1			1
3	Nancy.Conventions	1	1		-			1	1																1										1			
4	Nancy.Cookies	2				-			1														1	1	1					1	1							
5	Nancy.Cryptography		1				-		5																					2	5							
6	Nancy.Culture	1	2					-	1																		1											
7	Nancy.Diagnostics	8	8		3				-	11		2																							2			
8	Nancy.Diagnostics.Modules									-																												
9	Nancy.ErrorHandling	2	2								-																1											
10	Nancy.Extensions											-																										
11	Nancy.Helpers	3			1	1				1		1	-					1					1	1	1		1				1				1			
12	Nancy.IO	1												-									1															
13	Nancy.Json.Converters														-	2																						
14	Nancy.Json	2													1	-		2			3			3														
15	Reflection																-																					
16	Nancy.Json.Simple															2		-																				
17	Nancy.Localization	1	2						1										-																			
18	Nancy.ModelBinding	1	10						1											-	1	2					1											
19	Nancy.ModelBinding.DefaultBodyDeserializers																				-																	
20	Nancy.ModelBinding.DefaultConverters																					-																
21	Nancy.Owin																						-															
22	Nancy.Responses	6	2		1				1			1												-	3					1					1			
23	Nancy.Responses.Negotiation	5	3						3	1	1									1	1			1	-		4											
24	Nancy.Routing.Constraints		1						1																	-		1	1									
25	Nancy.Routing	2	17						11			1															-	2	1									
26	Nancy.Routing.Trie		4						2																	1	2	-	4									
27	Nancy.Routing.Trie.Nodes																											11	-									
28	Nancy.Security		2						1																					-					1			
29	Nancy.Session	1			2																										-							
30	Nancy.Tinyloc	4							2																							-						
31	Nancy.Validation	3	3																								1						-					
32	Nancy.Validation.Rules																																	-				
33	Nancy.ViewEngines	1	15		1				6	1															2		1								-	4		
34	Nancy.ViewEngines.SuperSimpleViewEngine								2																											-		
35	Nancy.ViewEngines.Razor																																				-	
36	Nancy.Xml																						1															-

ข.2.3. ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน

ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินด้วยเครื่องมือที่พัฒนาขึ้น โดยนำข้อมูลโครงสร้างของโปรเจคเป็นข้อมูลในการวิเคราะห์ตรวจหา โดยได้ผลการตรวจหาแสดงดังตารางที่ ข.9

ตารางที่ ข.9 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินด้วยเครื่องมือที่พัฒนา

ชื่อโปรเจค	ชื่อเนมสเปซ	จำนวนคลาส	จำนวนบรรทัด
Hyperion	Hyperion.Internal	74	539
Hyperion	Hyperion.ValueSerializers	32	1,154
Hangfire.Core	Hangfire	47	4,122
Hangfire.Core	Hangfire.Dashboard	49	2,993
Hangfire.Core	Hangfire.Server	35	1,843
Nancy	Nancy	96	10,980
Nancy	Nancy.Diagnostics	37	1,691

ข.2.4. ประเมินผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีด้วยเครื่องมือที่พัฒนาขึ้น โดยนำข้อมูลโครงสร้างของโปรเจคและข้อมูลความสัมพันธ์เป็นข้อมูลในการวิเคราะห์ตรวจหา โดยได้ผลการตรวจหาแสดงดังตารางที่ ข.10

ตารางที่ ข.10 ผลการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีด้วยเครื่องมือที่พัฒนา

ชื่อโปรเจค	ชื่อเนมสเปซ	เนมสเปซที่พึ่งพาที่เสถียรภาพน้อยกว่า
Hyperion	Hyperion	Hyperion.SerializerFactories
Hyperion	Hyperion.Extensions	Hyperion Hyperion.ValueSerializers
Hyperion	Hyperion.ValueSerializers	Hyperion

Hangfire.Core	Hangfire	Hangfire.Server Hangfire.Dashboard Hangfire.Dashboard.Pages
Hangfire.Core	Hangfire.Logging	Hangfire.Logging.LogProviders
Hangfire.Core	Hangfire.Client	Hangfire
Hangfire.Core	Hangfire.Common	Hangfire Hangfire.Server Hangfire.Client
Hangfire.Core	Hangfire.Dashboard.Pages	Hangfire.Dashboard
Hangfire.Core	Hangfire.States	Hangfire Hangfire.Common
Hangfire.Core	Hangfire.Storage	Hangfire.Common Hangfire.Server
Hangfire.Core	Hangfire.Storage.Monitoring	Hangfire.Common
Nancy	Nancy	Nancy.Bootstrapper Nancy.Diagnostics Nancy.Culture Nancy.Responses.Negotiation Nancy.Conventions Nancy.Responses Nancy.ViewEngines Nancy.Routing Nancy.Json Nancy.ErrorHandling Nancy.Session
Nancy	Nancy.Bootstrapper	Nancy.Diagnostics
Nancy	Nancy.Conventions	Nancy.Session Nancy.Bootstrapper Nancy.Diagnostics
Nancy	Nancy.Culture	Nancy.Conventions

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Nancy	Nancy.Diagnostics	Nancy.ViewEngines.SuperSimpleViewEngine
Nancy	Nancy.Helpers	Nancy
Nancy	Nancy.IO	Nancy
Nancy	Nancy.Json	Nancy.Json.Simple Nancy.Json.Converters
Nancy	Nancy.Localization	Nancy
Nancy	Nancy.ModelBinding	Nancy Nancy.Responses.Negotiation
Nancy	Nancy.Responses	Nancy.Xml
Nancy	Nancy.Responses.Negotiation	Nancy.Conventions Nancy.Responses Nancy.ViewEngines
Nancy	Nancy.Routing.Constraints	Nancy.Routing.Trie
Nancy	Nancy.Security	Nancy.Bootstrapper
Nancy	Nancy.Tinyloc	Nancy
Nancy	Nancy.Validation	Nancy
Nancy	Nancy.ViewEngines	Nancy.Security Nancy.Conventions Nancy.Diagnostics Nancy.Bootstrapper

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ข.2.5. ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน

จากตารางที่ 5.7 แสดงผลการเปรียบเทียบการตรวจหาจำนวนอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินด้วยเครื่องมือพาณิชย์และเครื่องมือที่พัฒนา โดยรายละเอียดการตรวจหาในระดับเนมสเปซ แสดงดังตารางที่ ข.11

ตารางที่ ข.11 ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกิน

ชื่อโปรเจค	ชื่อเนมสเปซ	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา
Hyperion	Hyperion	-	-
Hyperion	Hyperion.Compilation	-	-
Hyperion	Hyperion.Extensions	-	-
Hyperion	Hyperion.Internal	พบ	พบ
Hyperion	Hyperion.SerializerFactories	-	-
Hyperion	Hyperion.ValueSerializers	พบ	พบ
Hangfire.Core	Hangfire	พบ	พบ
Hangfire.Core	Hangfire.Annotations	-	-
Hangfire.Core	Hangfire.Client	-	-
Hangfire.Core	Hangfire.Common	-	-
Hangfire.Core	Hangfire.Common.ExpressionUtil	-	-
Hangfire.Core	Hangfire.Dashboard	พบ	พบ
Hangfire.Core	Hangfire.Dashboard.Owin	-	-
Hangfire.Core	Hangfire.Dashboard.Pages	-	-
Hangfire.Core	Hangfire.Dashboard.Resources	-	-
Hangfire.Core	Hangfire.Logging	-	-
Hangfire.Core	Hangfire.Logging.LogProviders	-	-
Hangfire.Core	Hangfire.Server	พบ	พบ
Hangfire.Core	Hangfire.States	-	-
Hangfire.Core	Hangfire.Storage	-	-
Hangfire.Core	Hangfire.Storage.Monitoring	-	-
Hangfire.Core	MoreLinq	-	-
Nancy	Nancy	พบ	พบ
Nancy	Nancy.Bootstrapper	-	-
Nancy	Nancy.Configuration	-	-
Nancy	Nancy.Conventions	-	-
Nancy	Nancy.Cookies	-	-

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Nancy	Nancy.Cryptography	-	-
Nancy	Nancy.Culture	-	-
Nancy	Nancy.Diagnostics	พบ	พบ
Nancy	Nancy.Diagnostics.Modules	-	-
Nancy	Nancy.ErrorHandling	-	-
Nancy	Nancy.Extensions	-	-
Nancy	Nancy.Helpers	-	-
Nancy	Nancy.IO	-	-
Nancy	Nancy.Json	-	-
Nancy	Nancy.Json.Converters	-	-
Nancy	Nancy.Json.Simple	-	-
Nancy	Nancy.Localization	-	-
Nancy	Nancy.ModelBinding	-	-
Nancy	Nancy.ModelBinding.DefaultBodyDeserializers	-	-
Nancy	Nancy.ModelBinding.DefaultConverters	-	-
Nancy	Nancy.Owin	-	-
Nancy	Nancy.Responses	-	-
Nancy	Nancy.Responses.Negotiation	-	-
Nancy	Nancy.Routing	-	-
Nancy	Nancy.Routing.Constraints	-	-
Nancy	Nancy.Routing.Trie	-	-
Nancy	Nancy.Routing.Trie.Nodes	-	-
Nancy	Nancy.Security	-	-
Nancy	Nancy.Session	-	-
Nancy	Nancy.TinyIoc	-	-
Nancy	Nancy.Validation	-	-
Nancy	Nancy.Validation.Rules	-	-
Nancy	Nancy.ViewEngines	-	-
Nancy	Nancy.ViewEngines.Razor	-	-
Nancy	Nancy.ViewEngines.SuperSimpleViewEngine	-	-

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Nancy	Nancy.Xml	-	-
Nancy	Reflection	-	-

จากตารางที่ ข.11 สามารถสรุปผลการเปรียบเทียบการตรวจหาอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินในรูปแบบคอนฟิวชันแมทริกซ์ (Confusion Matrix) ได้ดังตารางที่ ข.12 สำหรับโปรเจค Hyperion ตารางที่ ข.13 สำหรับโปรเจค Hangfire.Core และตารางที่ ข.14 สำหรับโปรเจค Nancy

ตารางที่ ข.12 ผลการเปรียบเทียบจำนวนแพ็กเกจขนาดใหญ่เกินรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hyperion

เครื่องมือพาณิชย์	**เครื่องมือที่พัฒนา**	
	พบ	**ไม่พบ**
พบ	2	0
ไม่พบ	0	4

ตารางที่ ข.13 ผลการเปรียบเทียบจำนวนแพ็กเกจขนาดใหญ่เกินรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hangfire.Core

เครื่องมือพาณิชย์	**เครื่องมือที่พัฒนา**	
	พบ	**ไม่พบ**
พบ	3	0
ไม่พบ	0	13

ตารางที่ ข.14 ผลการเปรียบเทียบจำนวนแพ็กเกจขนาดใหญ่เกินรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Nancy

เครื่องมือพาณิชย์	**เครื่องมือที่พัฒนา**	
	พบ	**ไม่พบ**
พบ	2	0
ไม่พบ	0	35

ข.2.6. ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

จากตารางที่ 5.7 แสดงผลการเปรียบเทียบการตรวจหาจำนวนอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีด้วยเครื่องมือพาณิชย์และเครื่องมือที่พัฒนา โดยรายละเอียดการตรวจหาในระดับเนมสเปซ แสดงดังตารางที่ ข.15

ตารางที่ ข.15 ผลการเปรียบเทียบการตรวจอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซี

ชื่อโปรเจค	ชื่อเนมสเปซ	เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา
Hyperion	Hyperion	พบ	พบ
Hyperion	Hyperion.Compilation	พบ	-
Hyperion	Hyperion.Extensions	พบ	พบ
Hyperion	Hyperion.Internal	-	-
Hyperion	Hyperion.SerializerFactories	-	-
Hyperion	Hyperion.ValueSerializers	พบ	พบ
Hangfire.Core	Hangfire	พบ	พบ
Hangfire.Core	Hangfire.Annotations	-	-
Hangfire.Core	Hangfire.Client	พบ	พบ
Hangfire.Core	Hangfire.Common	พบ	พบ
Hangfire.Core	Hangfire.Common.ExpressionUtil	-	-
Hangfire.Core	Hangfire.Dashboard	พบ	-
Hangfire.Core	Hangfire.Dashboard.Owin	-	-
Hangfire.Core	Hangfire.Dashboard.Pages	-	พบ
Hangfire.Core	Hangfire.Dashboard.Resources	-	-
Hangfire.Core	Hangfire.Logging	พบ	พบ
Hangfire.Core	Hangfire.Logging.LogProviders	-	-
Hangfire.Core	Hangfire.Server	-	-
Hangfire.Core	Hangfire.States	พบ	พบ
Hangfire.Core	Hangfire.Storage	พบ	พบ
Hangfire.Core	Hangfire.Storage.Monitoring	-	พบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Hangfire.Core	MoreLinq	-	-
Nancy	Nancy	พบ	พบ
Nancy	Nancy.Bootstrapper	-	พบ
Nancy	Nancy.Configuration	พบ	-
Nancy	Nancy.Conventions	พบ	พบ
Nancy	Nancy.Cookies	พบ	-
Nancy	Nancy.Cryptography	-	-
Nancy	Nancy.Culture	พบ	พบ
Nancy	Nancy.Diagnostics	พบ	พบ
Nancy	Nancy.Diagnostics.Modules	-	-
Nancy	Nancy.ErrorHandling	-	-
Nancy	Nancy.Extensions	-	-
Nancy	Nancy.Helpers	พบ	พบ
Nancy	Nancy.IO	พบ	พบ
Nancy	Nancy.Json	พบ	พบ
Nancy	Nancy.Json.Converters	-	-
Nancy	Nancy.Json.Simple	-	-
Nancy	Nancy.Localization	พบ	พบ
Nancy	Nancy.ModelBinding	พบ	พบ
Nancy	Nancy.ModelBinding.DefaultBodyDeserializers	-	-
Nancy	Nancy.ModelBinding.DefaultConverters	-	-
Nancy	Nancy.Owin	-	-
Nancy	Nancy.Responses	-	พบ
Nancy	Nancy.Responses.Negotiation	พบ	พบ
Nancy	Nancy.Routing	พบ	-
Nancy	Nancy.Routing.Constraints	พบ	พบ
Nancy	Nancy.Routing.Trie	พบ	พบ
Nancy	Nancy.Routing.Trie.Nodes	-	-
Nancy	Nancy.Security	พบ	พบ
Nancy	Nancy.Session	พบ	-

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Nancy	Nancy.Tinyloc	-	พบ
Nancy	Nancy.Validation	พบ	พบ
Nancy	Nancy.Validation.Rules	-	-
Nancy	Nancy.ViewEngines	พบ	พบ
Nancy	Nancy.ViewEngines.Razor	-	-
Nancy	Nancy.ViewEngines.SuperSimpleViewEngine	-	-
Nancy	Nancy.Xml	-	-
Nancy	Reflection	-	-

จากตารางที่ ข.15 สามารถสรุปผลการเปรียบเทียบการตรวจหาอาคิเทกเจอร์สเมลรูปแบบอันสเตเบิลดีเพนเดนซีในรูปแบบคอนฟิวชันแมทริกซ์ ได้ดังตารางที่ ข.16 สำหรับโปรเจค Hyperion ตารางที่ ข.17 สำหรับโปรเจค Hangfire.Core และตารางที่ ข.18 สำหรับโปรเจค Nancy

ตารางที่ ข.16 ผลการเปรียบเทียบจำนวนอันสเตเบิลดีเพนเดนซีรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hyperion

เครื่องมือพาณิชย์	**เครื่องมือที่พัฒนา**	
	พบ	**ไม่พบ**
พบ	3	1
ไม่พบ	0	2

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.17 ผลการเปรียบเทียบจำนวนอันสเตเบิลดีเพนเดนซีรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Hangfire.Core

เครื่องมือพาณิชย์	**เครื่องมือที่พัฒนา**	
	พบ	**ไม่พบ**
พบ	6	1
ไม่พบ	2	7

ตารางที่ ข.18 ผลการเปรียบเทียบจำนวนอันสเตเบิลดีเพนเดนซีรูปแบบคอนฟิวชันแมทริกซ์ของโปรเจค Nancy

เครื่องมือพาณิชย์	เครื่องมือที่พัฒนา	
	พบ	ไม่พบ
พบ	15	4
ไม่พบ	3	15

ข.3. ผลการประเมินวิธีการรีแฟคทอริงของเครื่องมือที่พัฒนาเพิ่มเติม

จากหัวข้อ 5.3 ประเมินวิธีการรีแฟคทอริงของเครื่องมือที่พัฒนา โดยในส่วนนี้จะแสดงรายละเอียดเพิ่มเติม ประกอบด้วย การประเมินผลการตรวจหาคอมมิวนิตีภายในเนมสเปซ และประเมินผลขั้นตอนการแยกเนมสเปซย่อย

ข.3.1. ประเมินผลขั้นตอนการตรวจหาคอมมิวนิตีภายในเนมสเปซ

โดยผลในขั้นตอนนี้เป็นการตรวจหาคอมมิวนิตีภายในเนมสเปซจากการแปลงข้อมูลโครงสร้างโปรเจคและความสัมพันธ์ของเนมสเปซที่ตรวจพบอาคิเทกเจอร์สเมลรูปแบบแพ็กเกจขนาดใหญ่เกินให้อยู่ในรูปแบบกราฟมีทิศทางและมีน้ำหนัก จากนั้นตรวจหาคอมมิวนิตีภายในเนมสเปซด้วยอัลกอริทึมลูแวน โดยผลการตรวจหาคอมมิวนิตีในรูปแบบกราฟมีทิศทางและมีน้ำหนักมีดังนี้

(1) เนมสเปซ Hyperion.Internal ในโปรเจค Hyperion

เนมสเปซ Hyperion.Internal ประกอบด้วย 74 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 5 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.19 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.1

ตารางที่ ข.19 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hyperion.Internal

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Hyperion	Hyperion.Internal	0	16
Hyperion	Hyperion.Internal	1	9
Hyperion	Hyperion.Internal	2	2
Hyperion	Hyperion.Internal	3	2
Hyperion	Hyperion.Internal	4	4
Hyperion	Hyperion.Internal	กลุ่มไม่มีความสัมพันธ์	41

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ ข.1 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hyperion.Internal

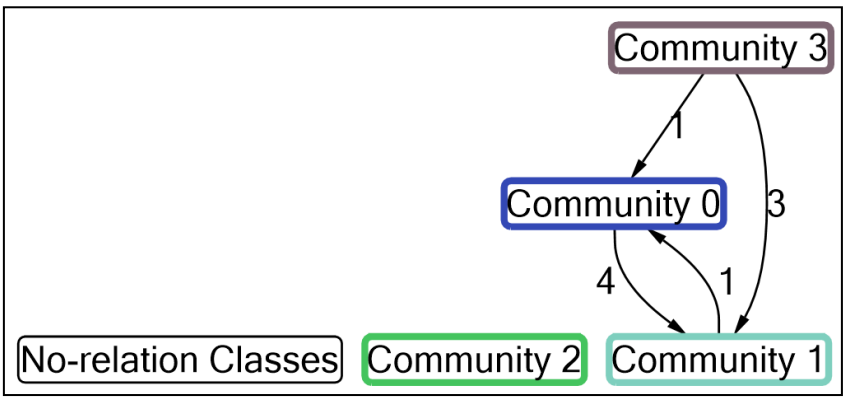
(2) เนมสเปซ Hyperion.ValueSerializers ในโปรเจค Hyperion

เนมสเปซ Hyperion.ValueSerializers ประกอบด้วย 32 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 4 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.20 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.2

ตารางที่ ข.20 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hyperion.ValueSerializers

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Hyperion	Hyperion.ValueSerializers	0	22
Hyperion	Hyperion.ValueSerializers	1	5
Hyperion	Hyperion.ValueSerializers	2	2
Hyperion	Hyperion.ValueSerializers	3	2
Hyperion	Hyperion.ValueSerializers	กลุ่มไม่มีความสัมพันธ์	1

จุฬาลงกรณ์มหาวิทยาลัย



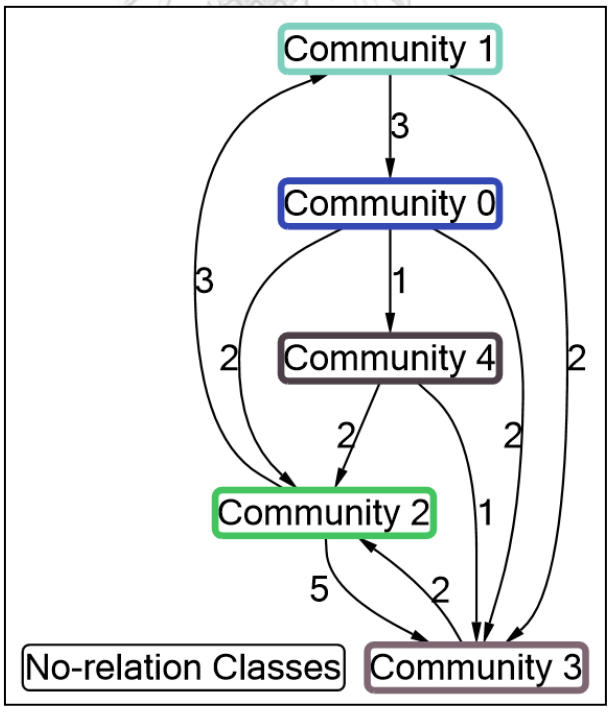
รูปที่ ข.2 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hyperion.ValueSerializers

(3) เนมสเปซ Hangfire ในโปรเจค Hangfire.Core

เนมสเปซ Hangfire ประกอบด้วย 47 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 5 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.21 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.3

ตารางที่ ข.21 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hangfire

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Hangfire.Core	Hangfire	0	5
Hangfire.Core	Hangfire	1	13
Hangfire.Core	Hangfire	2	15
Hangfire.Core	Hangfire	3	7
Hangfire.Core	Hangfire	4	3
Hangfire.Core	Hangfire	กลุ่มไม่มีความสัมพันธ์	4



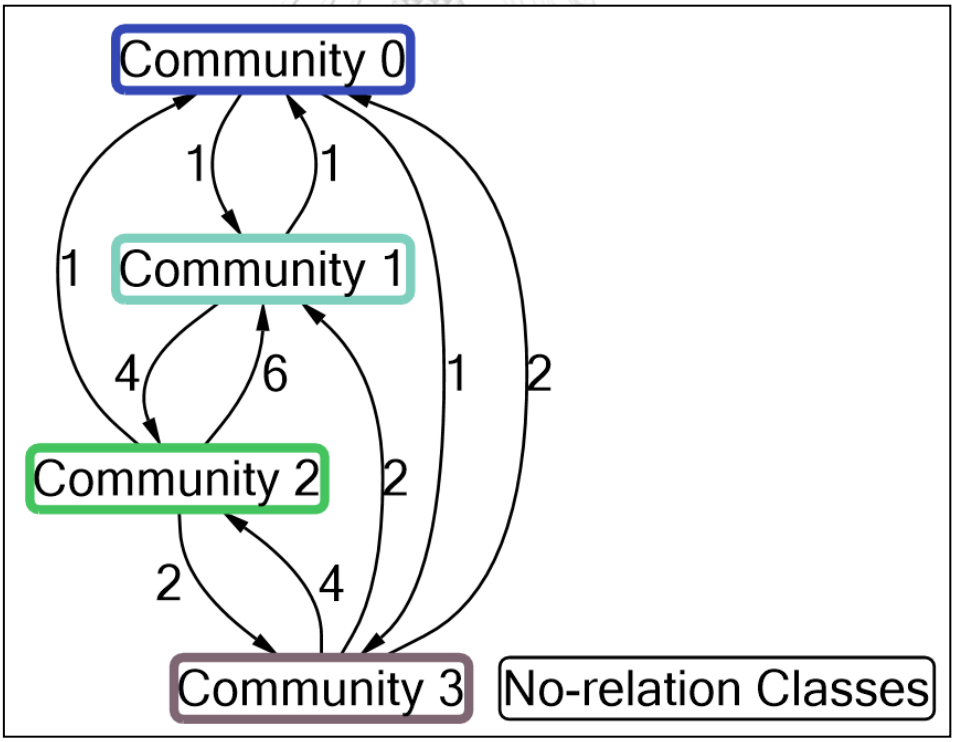
รูปที่ ข.3 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hangfire

(4) เนมสเปซ Hangfire.Dashboard ในโปรเจค Hangfire.Core

เนมสเปซ Hangfire.Dashboard ประกอบด้วย 49 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 4 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.22 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.4

ตารางที่ ข.22 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hangfire.Dashboard

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Hangfire.Core	Hangfire.Dashboard	0	7
Hangfire.Core	Hangfire.Dashboard	1	20
Hangfire.Core	Hangfire.Dashboard	2	15
Hangfire.Core	Hangfire.Dashboard	3	5
Hangfire.Core	Hangfire.Dashboard	กลุ่มไม่มีความสัมพันธ์	2



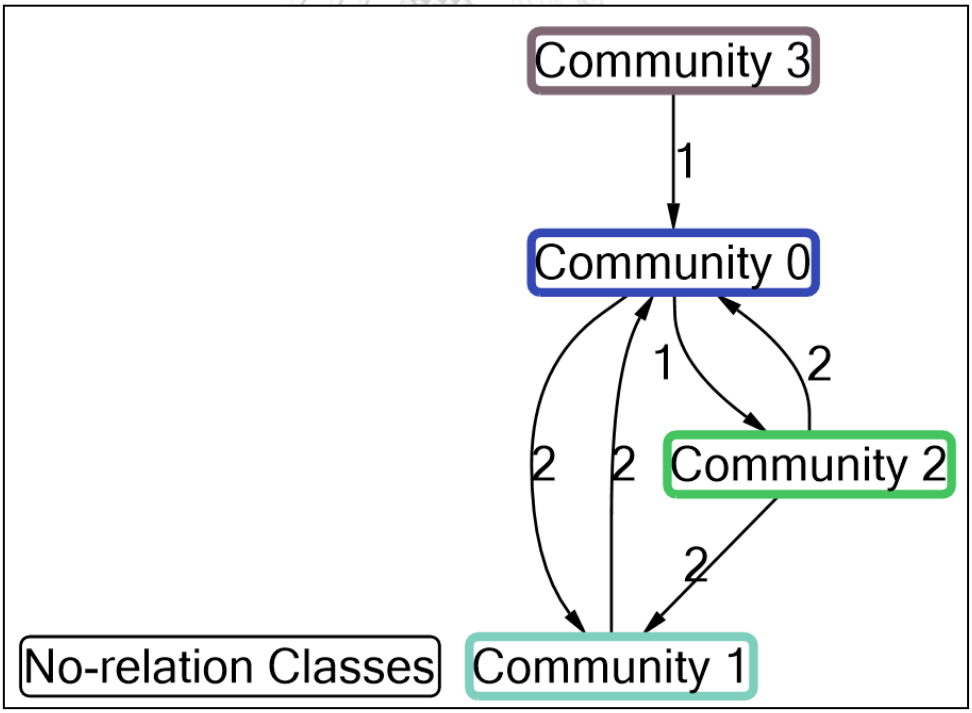
รูปที่ ข.4 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hangfire.Dashboard

(5) เนมสเปซ Hangfire.Server ในโปรเจค Hangfire.Core

เนมสเปซ Hangfire.Server ประกอบด้วย 35 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 4 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.23 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.5

ตารางที่ ข.23 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Hangfire.Server

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Hangfire.Core	Hangfire.Server	0	10
Hangfire.Core	Hangfire.Server	1	14
Hangfire.Core	Hangfire.Server	2	5
Hangfire.Core	Hangfire.Server	3	4
Hangfire.Core	Hangfire.Server	กลุ่มไม่มีความสัมพันธ์	2



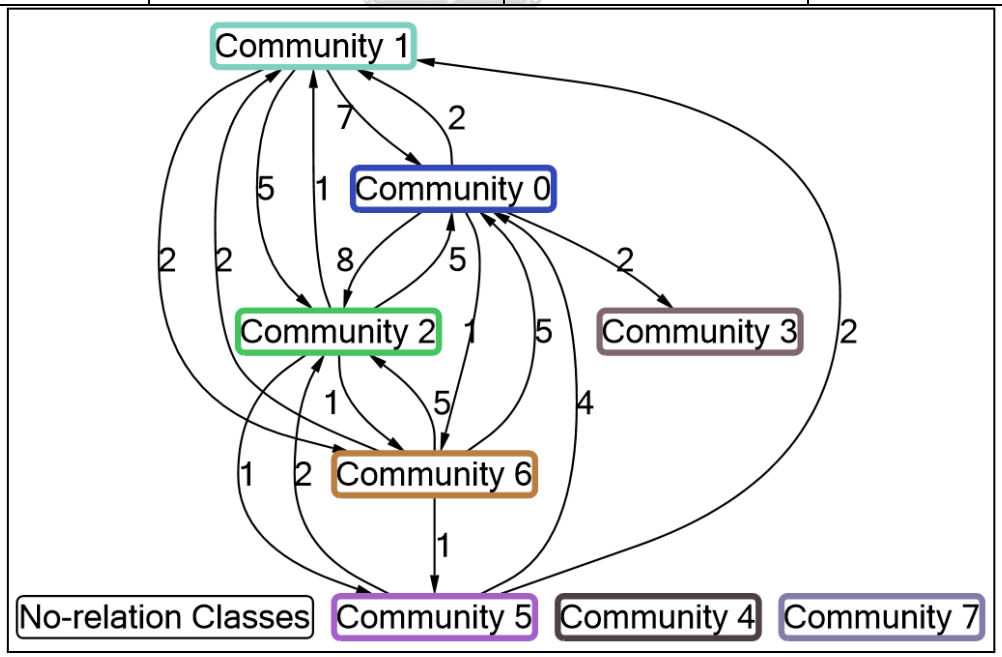
รูปที่ ข.5 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Hangfire.Server

(6) เนมสเปซ Nancy ในโปรเจค Nancy

เนมสเปซ Nancy ประกอบด้วย 96 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 8 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.24 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.6

ตารางที่ ข.24 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Nancy

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Nancy	Nancy	0	14
Nancy	Nancy	1	17
Nancy	Nancy	2	15
Nancy	Nancy	3	3
Nancy	Nancy	4	8
Nancy	Nancy	5	14
Nancy	Nancy	6	14
Nancy	Nancy	7	3
Nancy	Nancy	กลุ่มไม่มีความสัมพันธ์	8



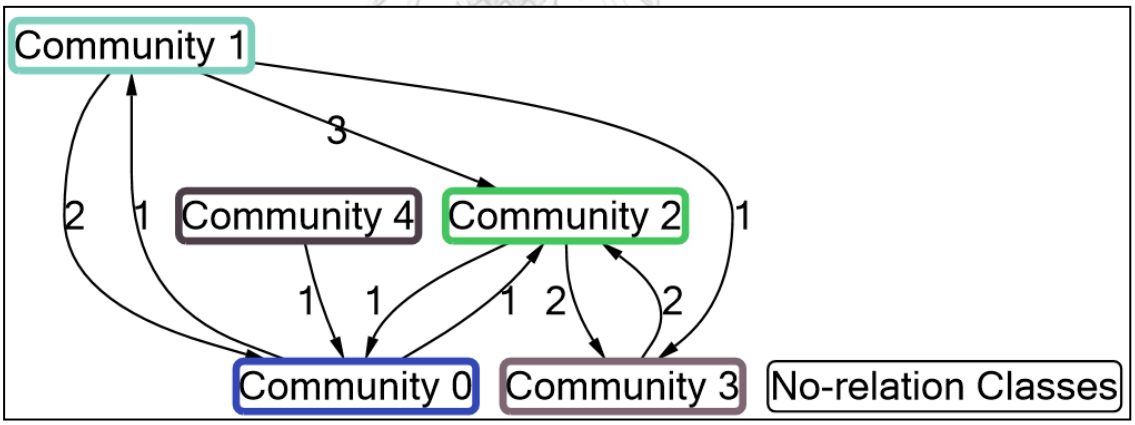
รูปที่ ข.6 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Nancy

(7) เนมสเปซ Nancy.Diagnostics ในโปรเจค Nancy

เนมสเปซ Nancy.Diagnostics ประกอบด้วย 37 คลาส โดยตรวจพบคอมมิวนิตีภาพใน 5 คอมมิวนิตี และกลุ่มของคลาสที่ไม่มีความสัมพันธ์ โดยมีรายละเอียดดังตารางที่ ข.25 และแสดงกราฟการพึ่งพาระหว่างคอมมิวนิตีดังรูปที่ ข.7

ตารางที่ ข.25 ผลการตรวจหาคอมมิวนิตีในเนมสเปซ Nancy.Diagnostics

ชื่อโปรเจค	ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส
Nancy	Nancy.Diagnostics	0	5
Nancy	Nancy.Diagnostics	1	4
Nancy	Nancy.Diagnostics	2	10
Nancy	Nancy.Diagnostics	3	8
Nancy	Nancy.Diagnostics	4	7
Nancy	Nancy.Diagnostics	กลุ่มไม่มีความสัมพันธ์	3



รูปที่ ข.7 กราฟการพึ่งพาระหว่างคอมมิวนิตีของเนมสเปซ Nancy.Diagnostics

ข.3.2. ประเมินผลขั้นตอนการแยกเนมสเปซย่อย

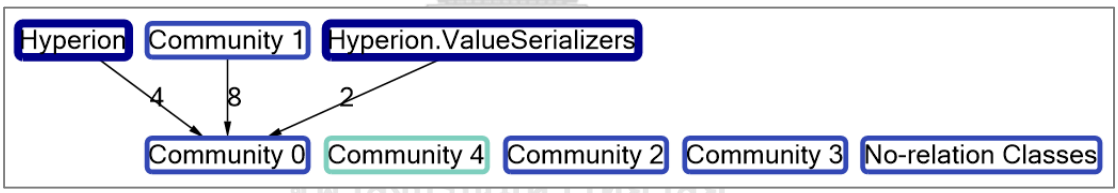
โดยผลในขั้นตอนนี้เป็นผลลัพธ์การแยกเนมสเปซย่อยหลังจากระบวนการรีแฟคทอริง โดยเครื่องมือแนะนำกลุ่มคอมมิวนิตีที่ควรแยกไปยังเนมสเปซย่อยใหม่ นอกจากนี้ยังแสดงกราฟความสัมพันธ์ของกลุ่มคอมมิวนิตี และเนมสเปซย่อยกับเนมสเปซรอบข้าง โดยมีรายละเอียดในแต่ละโปรเจคดังนี้

(1) เนมสเปซ Hyperion.Internal ในโปรเจค Hyperion

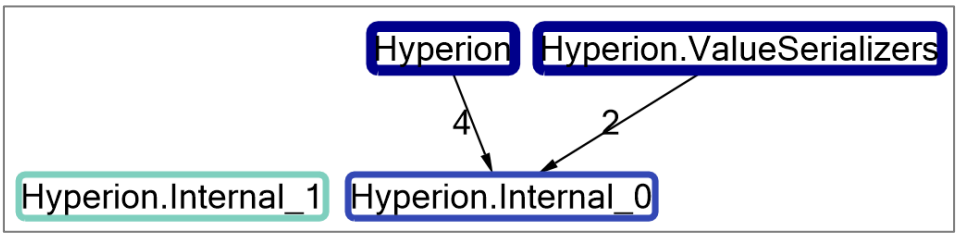
จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Hyperion.Internal พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น โดยการแยกคอมมิวนิตี 4 ออกมาเป็นเนมสเปซใหม่ โดยแสดงผลการจัดเนมสเปซใหม่ของคอมมิวนิตีภายในเนมสเปซ ดังตารางที่ ข.26 และแสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง ดังรูปที่ ข.8 และกราฟการพึ่งพาของเนมสเปซใหม่กับเนมสเปซรอบข้าง ดังรูปที่ ข.9

ตารางที่ ข.26 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hyperion.Internal

ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส	ชื่อเนมสเปซใหม่
Hyperion.Internal	0	16	Hyperion.Internal_0
Hyperion.Internal	1	9	Hyperion.Internal_0
Hyperion.Internal	2	2	Hyperion.Internal_0
Hyperion.Internal	3	2	Hyperion.Internal_0
Hyperion.Internal	4	4	Hyperion.Internal_1
Hyperion.Internal	กลุ่มไม่มีความสัมพันธ์	41	Hyperion.Internal_0



รูปที่ ข.8 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hyperion.Internal กับเนมสเปซรอบข้าง



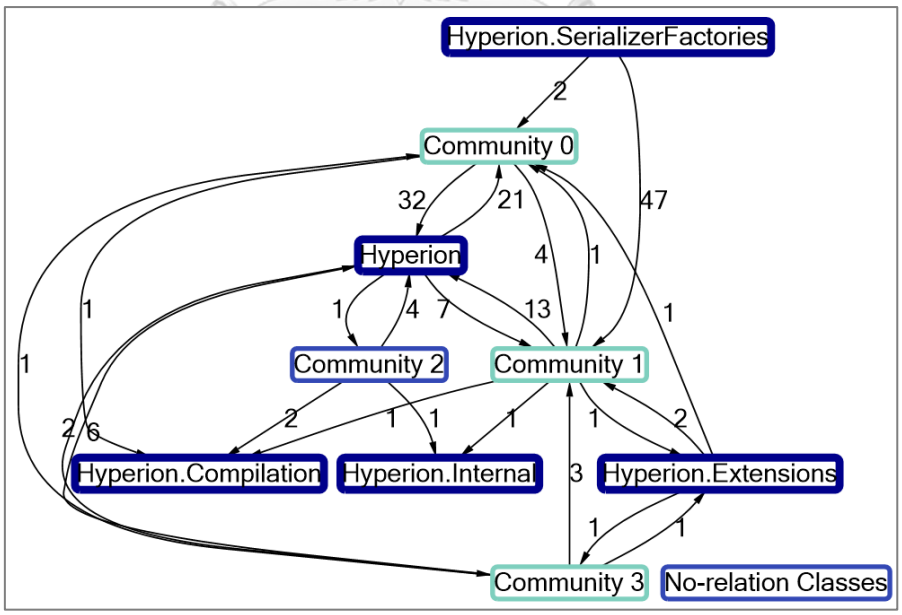
รูปที่ ข.9 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hyperion.Internal กับเนมสเปซรอบข้าง

(2) เนมสเปซ Hyperion.ValueSerializers ในโปรเจค Hyperion

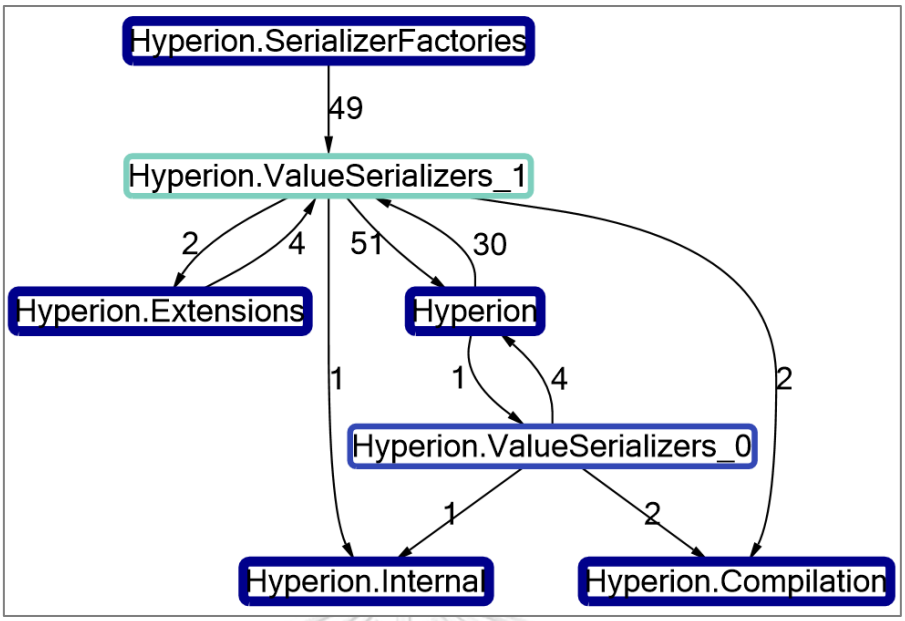
จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Hyperion.ValueSerializers พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น โดยการแยกคอมมิวนิตี 0 คอมมิวนิตี 1 และคอมมิวนิตี 3 ออกมาเป็นเนมสเปซใหม่ โดยแสดงผลการจัดเนมสเปซใหม่ของคอมมิวนิตีภายในเนมสเปซ ดังตารางที่ ข.27 และแสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง ดังรูปที่ ข.10 และกราฟการพึ่งพาของเนมสเปซใหม่กับเนมสเปซรอบข้าง ดังรูปที่ ข.11

ตารางที่ ข.27 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hyperion.ValueSerializers

ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส	ชื่อเนมสเปซใหม่
Hyperion.ValueSerializers	0	22	Hyperion.ValueSerializers_1
Hyperion.ValueSerializers	1	5	Hyperion.ValueSerializers_1
Hyperion.ValueSerializers	2	2	Hyperion.ValueSerializers_0
Hyperion.ValueSerializers	3	2	Hyperion.ValueSerializers_1
Hyperion.ValueSerializers	กลุ่มไม่มีความสัมพันธ์	1	Hyperion.ValueSerializers_0



รูปที่ ข.10 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hyperion.ValueSerializers กับเนมสเปซรอบข้าง



รูปที่ ข.11 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hyperion.ValueSerializers กับเนมสเปซรอบข้าง

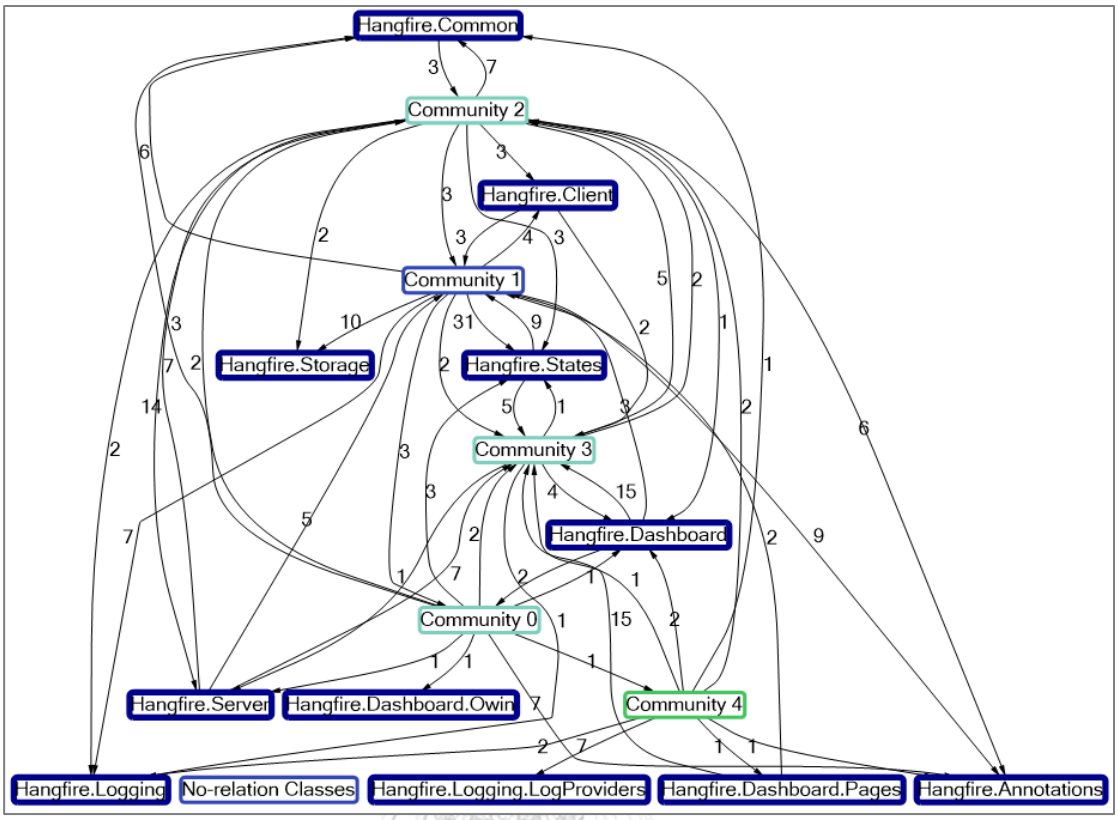
(3) เนมสเปซ Hangfire ในโปรเจค Hangfire.Core

จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Hangfire พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น โดยการแยก 2 เนมสเปซย่อยออกมา ประกอบด้วย เนมสเปซย่อยแรก คือ คอมมิวนิตี 0 คอมมิวนิตี 2 และคอมมิวนิตี 3 และเนมสเปซย่อยที่สอง คือ คอมมิวนิตีที่ 4 โดยแสดงผลการจัดเนมสเปซใหม่ของคอมมิวนิตีภายในเนมสเปซ ดังตารางที่ ข.28 และแสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง ดังรูปที่ ข.12 และกราฟการพึ่งพาของเนมสเปซใหม่กับเนมสเปซรอบข้าง ดังรูปที่ ข.13

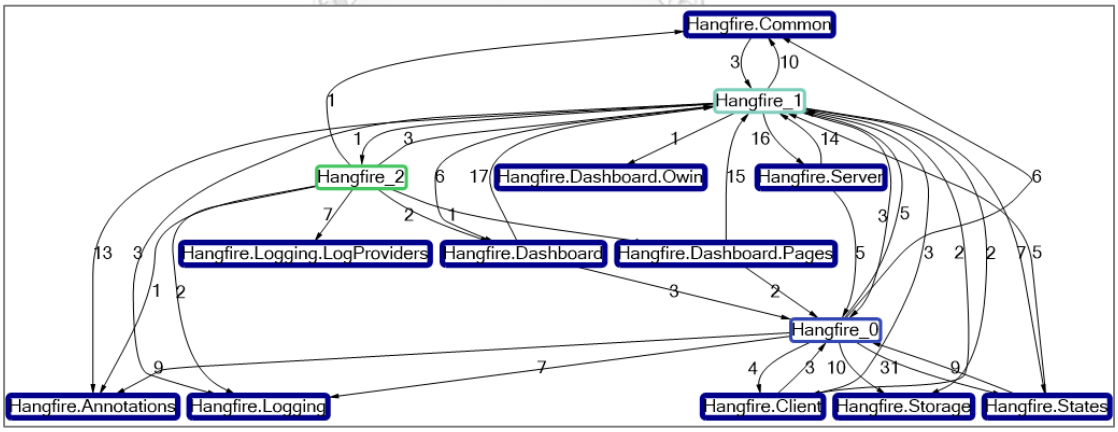
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.28 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hangfire

ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส	ชื่อเนมสเปซใหม่
Hangfire	0	5	Hangfire_1
Hangfire	1	13	Hangfire_0
Hangfire	2	15	Hangfire_1
Hangfire	3	7	Hangfire_1
Hangfire	4	3	Hangfire_2
Hangfire	กลุ่มไม่มีความสัมพันธ์	4	Hangfire_0



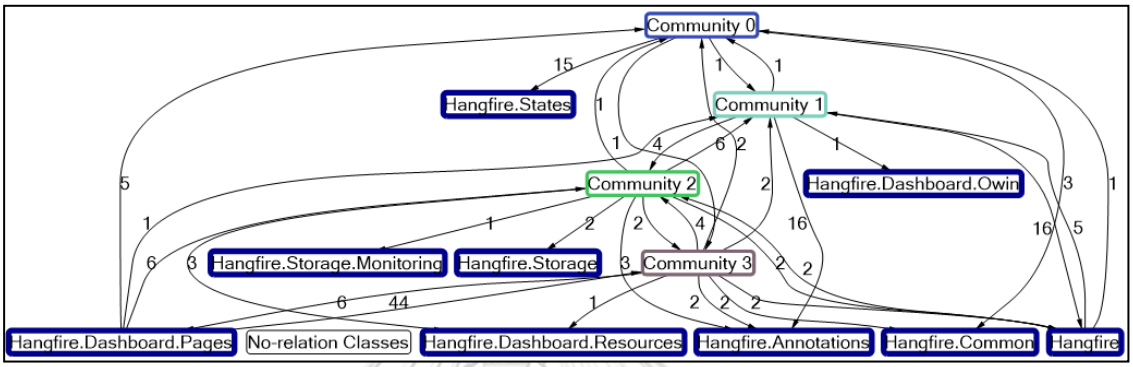
รูปที่ ข.12 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hangfire กับเนมสเปซรอบข้าง



รูปที่ ข.13 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hangfire กับเนมสเปซรอบข้าง

(4) เนมสเปซ Hangfire.Dashboard ในโปรเจค Hangfire.Core

จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Hangfire.Dashboard ไม่พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น เนื่องจากแต่ละคอมมิวนิตีมีความการพึ่งพากันไปมาระหว่างคอมมิวนิตีภายในเนมสเปซจำนวนมาก ทำให้ค่าความเชื่อมแน่นสูงถึง 1.82 ดังนั้นจึงส่งผลให้ไม่สามารถหาวิธีการรีแฟคทอริงเนมสเปซนี้และให้ผลที่ดีขึ้น โดยรูปที่ ข.14 แสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง



รูปที่ ข.14 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hangfire.Dashboard กับเนมสเปซรอบข้าง

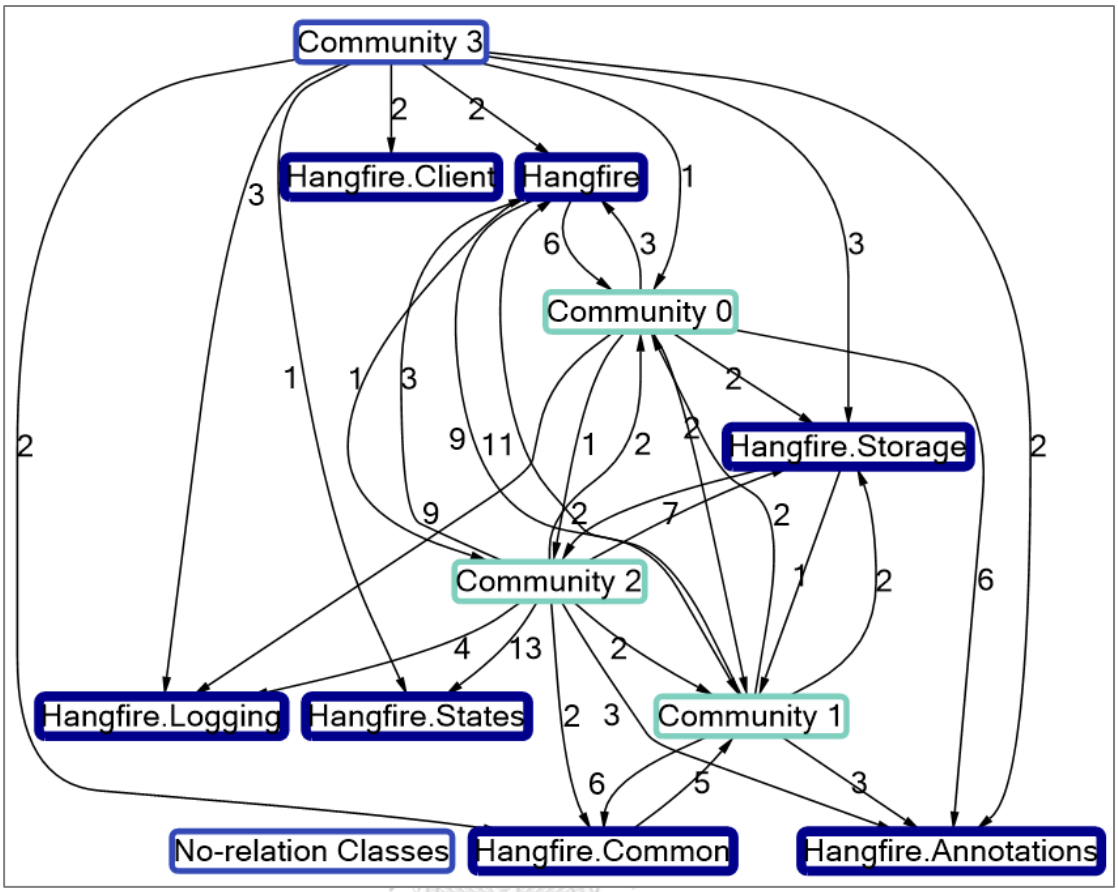
(5) เนมสเปซ Hangfire.Server ในโปรเจค Hangfire.Core

จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Hangfire.Server พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น โดยการแยกคอมมิวนิตี 0 คอมมิวนิตี 1 และคอมมิวนิตี 2 ออกมาเป็นเนมสเปซใหม่ โดยแสดงผลการจัดเนมสเปซใหม่ของคอมมิวนิตีภายในเนมสเปซ ดังตารางที่ ข.29 และแสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง ดังรูปที่ ข.15 และกราฟการพึ่งพาของเนมสเปซใหม่กับเนมสเปซรอบข้าง ดังรูปที่ ข.16

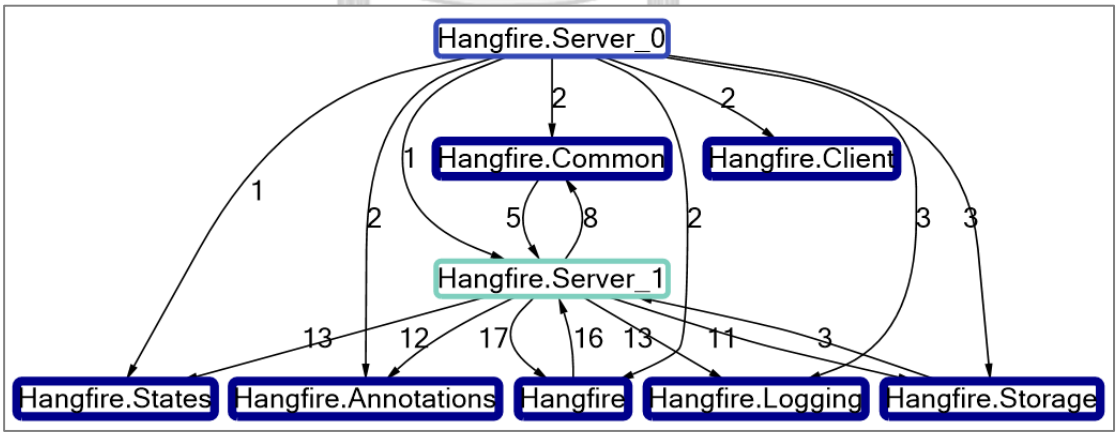
ตารางที่ ข.29 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Hangfire.Server

ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส	ชื่อเนมสเปซใหม่
Hangfire.Server	0	10	Hangfire.Server_1
Hangfire.Server	1	14	Hangfire.Server_1
Hangfire.Server	2	5	Hangfire.Server_1
Hangfire.Server	3	4	Hangfire.Server_0
Hangfire.Server	กลุ่มไม่มีความสัมพันธ์	2	Hangfire.Server_0

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ ข.15 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Hangfire.Server กับเนมสเปซรอบข้าง



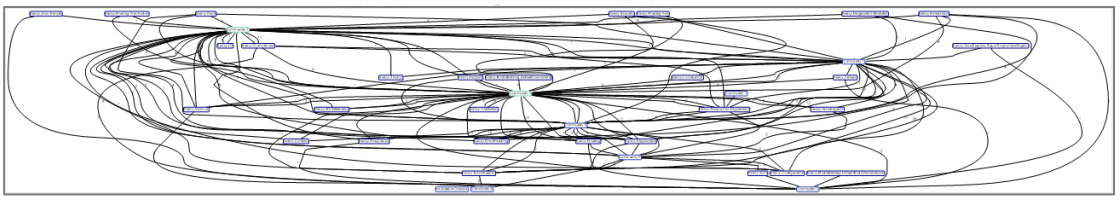
รูปที่ ข.16 กราฟการพึ่งพาของเนมสเปซใหม่ของ Hangfire.Server กับเนมสเปซรอบข้าง

(6) เนมสเปซ Nancy ในโปรเจค Nancy

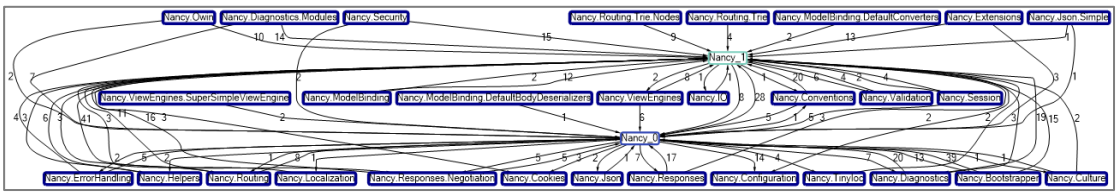
จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Nancy พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น โดยการแยกคอมมิวนิตี 0 และคอมมิวนิตี 2 ออกมาเป็นเนมสเปซใหม่ โดยแสดงผลการจัดเนมสเปซใหม่ของคอมมิวนิตีภายในเนมสเปซ ดังตารางที่ ข.30 และแสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง ดังรูปที่ ข.17 และกราฟการพึ่งพาของเนมสเปซใหม่กับเนมสเปซรอบข้าง ดังรูปที่ ข.18

ตารางที่ ข.30 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Nancy

ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส	ชื่อเนมสเปซใหม่
Nancy	0	14	Nancy_1
Nancy	1	17	Nancy_0
Nancy	2	15	Nancy_1
Nancy	3	3	Nancy_0
Nancy	4	8	Nancy_0
Nancy	5	14	Nancy_0
Nancy	6	14	Nancy_0
Nancy	7	3	Nancy_0
Nancy	กลุ่มไม่มีความสัมพันธ์	8	Nancy_0



รูปที่ ข.17 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Nancy กับเนมสเปซรอบข้าง



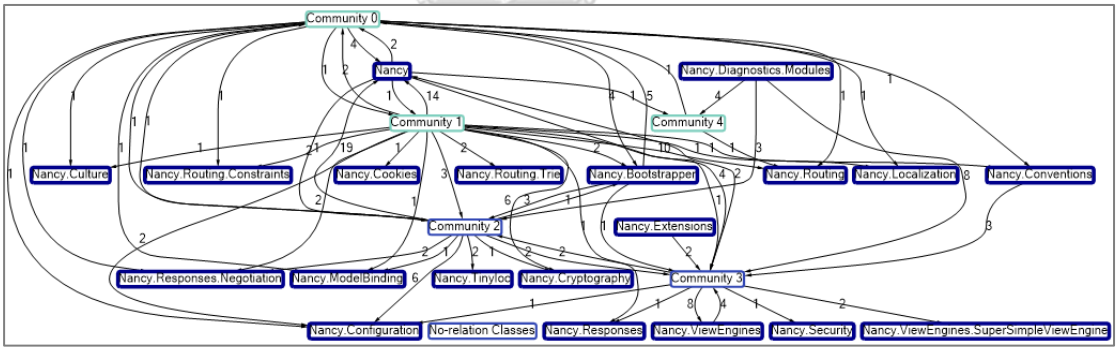
รูปที่ ข.18 กราฟการพึ่งพาของเนมสเปซใหม่ของ Nancy กับเนมสเปซรอบข้าง

(7) เนมสเปซ Nancy.Diagnostics ในโปรเจค Nancy

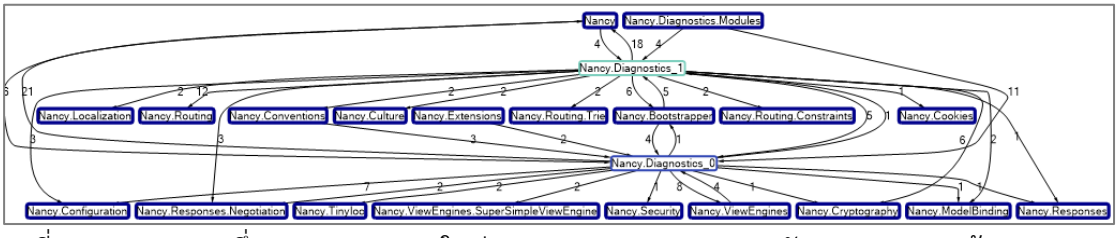
จากผลการตรวจหาคอมมิวนิตีและค้นหาวิธีการรีแฟคทอริงที่ให้ผลดีที่สุด สำหรับเนมสเปซ Nancy.Diagnostics พบวิธีการรีแฟคทอริงที่ให้ผลที่ดีขึ้น โดยการแยกคอมมิวนิตี 0 คอมมิวนิตี 1 และคอมมิวนิตี 4 ออกมาเป็นเนมสเปซใหม่ โดยแสดงผลการจัดเนมสเปซใหม่ของคอมมิวนิตีภายในเนมสเปซ ดังตารางที่ ข.31 และแสดงกราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซกับเนมสเปซรอบข้าง ดังรูปที่ ข.19 และกราฟการพึ่งพาของเนมสเปซใหม่กับเนมสเปซรอบข้าง ดังรูปที่ ข.20

ตารางที่ ข.31 ผลการแยกเนมสเปซใหม่ให้คอมมิวนิตีภายในเนมสเปซ Nancy.Diagnostics

ชื่อเนมสเปซ	คอมมิวนิตี	จำนวนคลาส	ชื่อเนมสเปซใหม่
Nancy.Diagnostics	0	5	Nancy.Diagnostics_1
Nancy.Diagnostics	1	4	Nancy.Diagnostics_1
Nancy.Diagnostics	2	10	Nancy.Diagnostics_0
Nancy.Diagnostics	3	8	Nancy.Diagnostics_0
Nancy.Diagnostics	4	7	Nancy.Diagnostics_1
Nancy.Diagnostics	กลุ่มไม่มีความสัมพันธ์	3	Nancy.Diagnostics_0



รูปที่ ข.19 กราฟการพึ่งพาของคอมมิวนิตีภายในเนมสเปซ Nancy.Diagnostics กับเนมสเปซรอบข้าง



รูปที่ ข.20 กราฟการพึ่งพาของเนมสเปซใหม่ของ Nancy.Diagnostics กับเนมสเปซรอบข้าง

ประวัติผู้เขียน

ชื่อ-สกุล	บุณยไชย จันทร์เทียน
วัน เดือน ปี เกิด	2 พฤศจิกายน 2531
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
ที่อยู่ปัจจุบัน	89/347 ซอยนวมินทร์ 81 แยก 3-10 ถนนนวมินทร์ แขวงนวมินทร์ เขตบึงกุ่ม กรุงเทพมหานคร 10240
ผลงานตีพิมพ์	งานวิจัยนี้ได้รับการคัดเลือกและตีพิมพ์เป็นบทความวิชาการเรื่อง A Refactoring Approach for Too Large Packages using Community Detection and Dependency-based Impacts โดยบุณยไชย จันทร์เทียน และพรศิริ หมื่นไชยศรี ในการประชุมวิชาการ “2019 The World Symposium on Software Engineering (WSSE 2019)” ระหว่างวันที่ 20-22 กันยายน 2562 ณ เมืองอู่ฮั่น ประเทศจีน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University