การจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

นายพิศักดิ์ คุรุเสถียร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-53-2760-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

TEST CASE PRIORITIZING FOR GRAPHICAL USER INTERFACE TESTING

Mr.Pisak Kurusathian

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic year 2005

ISBN 974-53-2760-3

หัวข้อวิทยานิพนธ์	การจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้
โดย	นายพิศักดิ์ คุรุเสถียร
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร. ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร. สาธิต วงศ์ประทีป)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. ญาใจ ลิ่มปิยะกรณ์)

..... กรรมการ

(รองศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล)

..... กรรมการ

(อาจารย์ ดร. เฉลิมเอก อินทนากรวิวัฒน์)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พิศักดิ์ คุรุเสถียร : การจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้. (TEST CASE PRIORITIZING FOR GRAPHICAL USER INTERFACE TESTING) อ. ที่ปรึกษา : ผศ. ดร. ญาใจ ลิ่มปิยะกรณ์, 111 หน้า. ISBN 974-53-2760-3.

ส่วนต่อประสานกราฟิกกับผู้ใช้เป็นส่วนประกอบสำคัญของซอฟต์แวร์ส่วนใหญ่ในปัจจุบัน ในขณะที่ส่วนต่อประสานกราฟิกกับผู้ใช้ช่วยผู้ใช้เรียนรู้และใช้งานแอพลิเคชันได้อย่างรวดเร็ว การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้กลับเป็นงานที่ยุ่งยากซับซ้อนและสิ้นเปลืองทรัพยากร

วิทยานิพนธ์นี้นำเสนอวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อให้ผู้ทดสอบสามารถเลือกทดสอบเพียงบางส่วนได้ โดยส่งผลกระทบถึงผลลัพธ์การทดสอบน้อยที่สุด ใช้ทรัพยากรในการทดสอบลดลง และทำให้การทดสอบมีประสิทธิภาพสูงขึ้น การจัดลำดับความสำคัญกรณีทดสอบที่นำเสนอใช้เกณฑ์ความครอบคลุมเป็นหลักในการพิจารณาลำดับความสำคัญของกรณีทดสอบ

วิทยานิพนธ์นี้ยังได้พัฒนาเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ตามแนวทางที่นำเสนอไว้ และทดสอบเปรียบเทียบประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบที่ได้ออกแบบไว้ ซึ่งผลการทดสอบแสดงให้เห็นว่าการจัดลำดับความสำคัญกรณีทดสอบสามารถช่วยเพิ่มอัตราการค้นพบข้อผิดพลาดของชุดทดสอบระหว่างการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมศาสตร์ ลายมือชื่อนิสิต พิศักดิ์ คุรุเสถียร
สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่ออาจารย์ที่ปรึกษา
ปีการศึกษา 2548

4670420021 : MAJOR SOFTWARE ENGINEERING

KEY WORD: GUI TESTING / PRIORITIZATION / TEST CASE PRIORITY

PISAK KURUSATHIAN : TEST CASE PRIORITIZING FOR GRAPHICAL USER INTERFACE TESTING. THESIS ADVISOR : ASST. PROF. YACHAI LIMPIYAKORN PH.D., 111 pp. ISBN 974-53-2760-3.

Graphical User Interfaces (GUIs) are an important part in most today software. Whereas GUIs make software applications easier for users to learn and use, GUI testing becomes a complicated part in software testing and consumes excessive testing resources.

This thesis proposes an approach to prioritize test cases for GUI testing. Testers can selectively test a number of test cases with higher priorities, but no major impacts on the test results. These techniques can reduce testing resources that are used in GUI testing and they can improve testing efficiency. Our approach uses coverage criteria to assess test case priorities.

We also implemented a test case prioritizing tool in order to compare the performance of each technique. The experimental results showed that the proposed prioritization techniques are promising and they could improve the fault detection rate of a given test suite during GUI testing.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Computer Engineering.....Student's signature.....

Field of study.....Software Engineering...Advisor's signature.....

Academic year .....2005.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีจากความช่วยเหลือและคำแนะนำจาก ผู้ช่วยศาสตรจารย์ ดร. ญาใจ ลิ่มปิยกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งสละเวลาอันมีค่าให้คำแนะนำถึงแนวทางในการทำวิจัย และให้คำปรึกษาอันเป็นประโยชน์ยิ่งต่อการทำงาน และขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านที่ได้กรุณาให้คำแนะนำ และชี้แนะในการทำงานวิจัย รวมถึงตรวจสอบวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณคณาจารย์ทุกท่านที่ให้วิชาความรู้ซึ่งได้นำมาใช้ในวิทยานิพนธ์ฉบับนี้

ขอขอบคุณรุ่นพี่และเพื่อนๆในห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ที่ให้คำแนะนำและกำลังใจในการทำงาน

ขอขอบคุณนักพัฒนาซอฟต์แวร์ทุกท่านที่พัฒนาซอฟต์แวร์ตัวอย่างอันเป็นประโยชน์อย่างยิ่งต่องานวิจัย และขอขอบคุณทุกๆบทความและคำแนะนำของทุกท่านที่ได้ให้ความรู้และแนวทางแก้ไขปัญหาต่างๆที่พบในงานวิจัย

สุดท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา ญาติพี่น้องทุกท่าน ที่สนับสนุนด้านการเงิน ให้กำลังใจ และช่วยเหลือในด้านต่างๆเสมอมา ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ญ
สารบัญภาพ ..... ฐ

1. บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 ขั้นตอนการดำเนินการวิจัย 3
1.5 ประโยชน์ที่คาดว่าจะได้รับ 4
1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ 4
1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 4

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5
2.1 ส่วนต่อประสานกราฟิกกับผู้ใช้ 5
2.2 การทดสอบซอฟต์แวร์ 6
2.3 การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ 6
2.4 การจัดลำดับความสำคัญกรณีทดสอบ 11
2.5 งานวิจัยที่เกี่ยวข้อง 14

3. วิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ .. 15
3.1 ขั้นตอนการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ 15
3.2 แนวคิดการจัดลำดับความสำคัญกรณีทดสอบ 16
3.3 ประเภทการจัดลำดับความสำคัญกรณีทดสอบ 22
3.4 ปัญหาที่พบและแนวทางการแก้ไข 29

หน้า

4. การออกแบบและพัฒนาเครื่องมือ 33
4.1 การวิเคราะห์และออกแบบเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ 33
4.2 สภาพแวดล้อมในการพัฒนาเครื่องมือ 40
4.3 โครงสร้างกรณีทดสอบ 40
4.4 ขั้นตอนการจัดลำดับความสำคัญกรณีทดสอบ 42

5. การทดสอบวิธีการจัดลำดับความสำคัญของกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ 46
5.1 วัตถุประสงค์การทดสอบ 46
5.2 ขั้นตอนการทดสอบ 47
5.3 การประเมินผลการทดสอบ 48
5.4 รายละเอียดการทดสอบ 52
5.5 ผลการทดสอบ 55
5.6 สรุปผลการทดสอบ 68

6. สรุปผลการวิจัยและข้อเสนอแนะ 70
6.1 สรุปผลการวิจัย 70
6.2 ข้อเสนอแนะและแนวทางการพัฒนา 71

รายการอ้างอิง 72
ภาคผนวก 74
ภาคผนวก ก ตัวอย่างโปรแกรม 75
ภาคผนวก ข ขอบเขตการทดสอบ 85
ภาคผนวก ค ข้อผิดพลาดที่พบ 92
ภาคผนวก ง การคำนวณค่าเฉลี่ยการเลือก 99
ภาคผนวก จ การแจกแจงข้อมูล 102
ภาคผนวก ฉ คู่มือการใช้งานเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ 105

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

ประวัติผู้เขียนวิทยานิพนธ์ 111

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง หน้า

ตารางที่ 3.1 ประเภทการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ทั้งหมด 23
ตารางที่ 3.2 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ 24
ตารางที่ 3.3 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเบ็ดเสร็จ 24
ตารางที่ 3.4 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเบ็ดเสร็จ 25
ตารางที่ 3.5 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเบ็ดเสร็จ 25
ตารางที่ 3.6 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม 26
ตารางที่ 3.7 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม 27
ตารางที่ 3.8 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม ... 28
ตารางที่ 3.9 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม 28
ตารางที่ 3.10 ตัวอย่างการจัดลำดับความสำคัญกรณีทดสอบที่มีความซ้ำซ้อนโดยใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ 30
ตารางที่ 3.11 ตัวอย่างปัญหา Set covering ในการจัดลำดับความสำคัญกรณีทดสอบ 31
ตารางที่ 4.1 ขั้นตอนการจัดลำดับความสำคัญกรณีทดสอบ 33
ตารางที่ 4.2 รายละเอียดยูสเคส Prioritize test case 35
ตารางที่ 4.3 รายละเอียดยูสเคส Random test case order 36
ตารางที่ 4.4 รายละเอียดยูสเคส Calculate APFD 36
ตารางที่ 4.5 รายละเอียดยูสเคส Find average result 37
ตารางที่ 4.6 ตัวอย่างกรณีทดสอบ 41
ตารางที่ 5.1 รายละเอียดโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้ทดสอบ 54
ตารางที่ 5.2 ข้อมูลผลการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมตัวอย่าง 55

หน้า

ตารางที่ 5.3 ค่า APFD ชุดทดสอบของโปรแกรมต่างๆที่ผ่านการจัดลำดับความสำคัญกรณีทดสอบ 56
ตารางที่ 5.4 จำนวนกรณีทดสอบที่มีระดับความสำคัญกับข้อผิดพลาดที่พบ สำหรับการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม 59
ตารางที่ 5.5 เปอร์เซ็นต์ของกรณีทดสอบที่มีระดับความสำคัญกับข้อผิดพลาดที่พบ สำหรับการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม 59
ตารางที่ 5.6 เปอร์เซ็นต์ของกรณีทดสอบและเหตุการณ์ที่เป็นองค์ประกอบของกรณีทดสอบที่มีระดับความสำคัญมากกว่า 0 เมื่อจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการแบบเพิ่ม 62
ตารางที่ ค-1 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Naïve 92
ตารางที่ ค-2 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Object Counter 92
ตารางที่ ค-3 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม CImage Demo 93
ตารางที่ ค-4 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Grade (โปรแกรมคำนวณเกรดเฉลี่ย) 93
ตารางที่ ค-5 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Easy Info 94
ตารางที่ ค-6 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมร้านเช่าหนังสือ 95
ตารางที่ ค-7 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Address Book Standard Edition 96
ตารางที่ ค-8 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Small Math 2D 97
ตารางที่ ค-9 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Satang (สตางค์) 97

หน้า

ตารางที่ ค-10 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม F-CRC Calculator 98
ตารางที่ ฉ-1 ทิศทางการนำเข้าและส่งออกข้อมูลกรณีทดสอบของเครื่องมือ 106
ตารางที่ ฉ-2 แท็กและคุณสมบัติต่างๆที่ใช้ในการกำหนดข้อมูลกรณีทดสอบ 107

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ หน้า
รูปที่ 2.1 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ 2 แบบ ที่ทำหน้าที่เหมือนกัน 7
รูปที่ 2.2 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ที่มีทางเลือกในการทำงานจำนวนมาก 8
รูปที่ 2.3 การตรวจสอบผลลัพธ์ในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้..... 8
รูปที่ 2.4 ตัวอย่างข้อผิดพลาดเกี่ยวกับการแสดงผลสัญรูป 8
รูปที่ 3.1 ขั้นตอนการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้การจัดลำดับความสำคัญของกรณีทดสอบ 16
รูปที่ 3.2 โครงสร้างของกรณีทดสอบ..... 17
รูปที่ 3.3 ความสัมพันธ์ระหว่างเหตุการณ์ในกรณีทดสอบ..... 19
รูปที่ 3.4 ตัวอย่างภารกิจค้นหาข้อความ 20
รูปที่ 3.5 แผนภาพเวนน์แสดงระดับความครอบคลุมของเกณฑ์ความครอบคลุมรูปแบบต่างๆ 22
รูปที่ 3.6 ความสัมพันธ์ระหว่างเหตุการณ์โดยพิจารณาสถานะเริ่มต้น..... 29
รูปที่ 4.1 แผนภาพยูสเคสของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ 34
รูปที่ 4.2 แผนภาพคลาสของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ 37
รูปที่ 4.3 แผนภาพซีเควนซ์ของการจัดลำดับความสำคัญกรณีทดสอบ..... 39
รูปที่ 4.4 แผนภาพซีเควนซ์ของการทดสอบซ้ำเพื่อหาค่าเฉลี่ย..... 39
รูปที่ 4.5 แผนภาพลำดับกิจกรรมวิธีจัดลำดับความสำคัญกรณีทดสอบเชิงเดี่ยวแบบเบ็ดเสร็จ 42
รูปที่ 4.6 แผนภาพลำดับกิจกรรมวิธีจัดลำดับความสำคัญกรณีทดสอบเชิงเดี่ยวแบบเพิ่ม 44
รูปที่ 5.1 แผนภูมิแสดงความสัมพันธ์ระหว่างชุดทดสอบกับเปอร์เซ็นต์ของข้อผิดพลาดที่พบ..... 49
รูปที่ 5.1 แผนภูมิแสดงค่า APFD ของชุดทดสอบโปรแกรม เมื่อจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการแบบเบ็ดเสร็จ 57
รูปที่ 5.2 แผนภูมิแสดงค่า APFD ของชุดทดสอบโปรแกรม เมื่อจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการแบบเพิ่ม 60
รูปที่ 5.3 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มจำนวนกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ 63

หน้า

รูปที่ 5.4 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มจำนวนกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม 64
รูปที่ 5.5 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มขนาดกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ 65
รูปที่ 5.6 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มขนาดกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม 65
รูปที่ 5.7 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มปริมาณเหตุการณ์หรือภารกิจทั้งหมดในการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ 66
รูปที่ 5.8 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มปริมาณเหตุการณ์หรือภารกิจทั้งหมดในการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมเชิงเดี่ยวแบบเพิ่ม 67
รูปที่ 5.9 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มปริมาณเหตุการณ์หรือภารกิจทั้งหมดในการจัดลำดับความสำคัญกรณีทดสอบ 67
รูปที่ ก-1 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Naive 75
รูปที่ ก-2 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Object Counter 76
รูปที่ ก-3 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม CImage Demo 77
รูปที่ ก-4 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Grade 78
รูปที่ ก-5 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Easy Info 79
รูปที่ ก-6 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมร้านเช่าหนังสือ 80
รูปที่ ก-7 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Address Book Standard Edition 81
รูปที่ ก-8 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Small Math 2D 81
รูปที่ ก-9 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมสตางค์ 82
รูปที่ ก-10 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม F-CRC Calculator 83
รูปที่ จ-1 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับกรณีทดสอบแบบสุ่ม 102
รูปที่ จ-2 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม 103
รูปที่ จ-3 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม 103

หน้า

รูปที่ จ-4 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบ โดยใช้เกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม.....104
รูปที่ จ-5 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบ โดยใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม.....104
รูปที่ ฉ-1 ตัวอย่างการจัดเก็บข้อมูลกรณีทดสอบในแฟ้มเอ็กซ์เอ็มแอล.....107
รูปที่ ฉ-2 ส่วนต่อประสานกราฟิกกับผู้ใช้หลักของเครื่องมือ 108
รูปที่ ฉ-3 ส่วนแสดงผลค่า APFD.....109
รูปที่ ฉ-4 ส่วนต่อประสานกราฟิกกับผู้ใช้ของส่วนทดสอบซ้ำ.....110

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการพัฒนาซอฟต์แวร์ คุณภาพของซอฟต์แวร์เป็นสิ่งสำคัญที่ต้องคำนึงถึง โดยเฉพาะข้อผิดพลาดที่เกิดขึ้นกับการทำงานของซอฟต์แวร์ ซึ่งส่งผลให้ซอฟต์แวร์ไม่สามารถทำหน้าที่ตามที่ออกแบบไว้ได้ ทำให้จำเป็นต้องมีการทดสอบ (Testing) ซอฟต์แวร์ ทั้งในระหว่างการพัฒนาและก่อนการนำไปใช้งานจริง เพื่อให้มั่นใจว่าซอฟต์แวร์สามารถทำงานได้ตรงตามที่ออกแบบไว้

ในปัจจุบัน เนื่องจากความสามารถในการประมวลผลของคอมพิวเตอร์ที่เพิ่มขึ้น รวมทั้งการพัฒนาอย่างรวดเร็วของเทคโนโลยีที่เกี่ยวข้อง ทำให้ส่วนต่อประสานกับผู้ใช้ (User Interface) ของซอฟต์แวร์ต่างๆ มีการพัฒนา เพื่อเสริมสร้างความเข้าใจ อำนวยความสะดวก และเพิ่มประสิทธิภาพในการทำงานของผู้ใช้ ซอฟต์แวร์ในปัจจุบันจึงมีส่วนต่อประสานกับผู้ใช้ที่มีขนาดใหญ่และซับซ้อนเพิ่มขึ้นกว่าในอดีตมาก โดยเฉพาะซอฟต์แวร์ที่มีส่วนต่อประสานกราฟิกกับผู้ใช้หรือจียูไอ (Graphical User Interface : GUI) ที่เพิ่มทางเลือกในการสื่อสารระหว่างผู้ใช้กับซอฟต์แวร์ให้ใช้งานได้สะดวกยิ่งขึ้น ทำให้ส่วนต่อประสานกราฟิกกับผู้ใช้ได้รับความนิยม และถูกนำไปใช้อย่างแพร่หลาย แต่การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้กลับยังมีประเด็นที่จำเป็นต้องได้รับการศึกษา และพัฒนาต่อไป ทั้งนี้เพราะลักษณะเฉพาะที่แตกต่างไปจากซอฟต์แวร์ปกติ (Conventional Software) และทางเลือกในการทำงานที่มีเป็นจำนวนมาก จึงทำให้การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จำเป็นต้องใช้เทคนิควิธีการที่แตกต่างออกไปจากการทดสอบซอฟต์แวร์ตามปกติ

นอกจากนี้ปัญหาสำคัญประการหนึ่งของการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ คือ กรณีทดสอบ (Test case) อันประกอบด้วยลำดับข้อมูลนำเข้าและผลลัพธ์ในแต่ละขั้นตอน ซึ่งมีจำนวนมหาศาล ทั้งนี้เนื่องจากทางเลือกในการทำงานที่มีมาก ทำให้กรณีทดสอบมีจำนวนมากตามไปด้วย การทดสอบกรณีทดสอบทั้งหมดจึงเป็นไปได้ยากในการทำงานจริง โดยเฉพาะในระบบงานที่มีขนาดใหญ่ จึงจำเป็นต้องมีวิธีการที่จะลดจำนวนกรณีทดสอบลง โดยส่งผลถึงประสิทธิภาพของการทดสอบให้น้อยที่สุดเท่าที่จะเป็นไปได้

มีงานวิจัยหลายชิ้นได้กล่าวถึงแนวคิดในการลดจำนวนกรณีทดสอบ เช่น งานวิจัยเรื่อง “Regression Testing of GUI Event Interaction” โดย Lee White [1] ที่เสนอการนำเทคนิค

Latin Square มาใช้ในการสร้างกรณีทดสอบ งานวิจัยเรื่อง “Using a Goal-driven Approach to Generate Test Cases for GUIs” โดย Atif M. Memon et al. [2] ที่เสนอแนวคิดในการสร้างกรณีทดสอบแบ่งตามหน้าที่การทำงาน และงานวิจัย “User-Based Testing of GUI Sequences and Their Interactions” โดย Lee White et al. [3] ที่เสนอการสร้างกรณีทดสอบโดยแบ่งตามหน้าที่การทำงาน และใช้การวิเคราะห์โครงสร้าง เพื่อช่วยลดจำนวนกรณีทดสอบ

งานวิจัยที่กล่าวมาส่วนใหญ่นำเสนอวิธีการสร้างกรณีทดสอบขึ้นมาใหม่ให้มีจำนวนไม่มากจนเกินไป แต่ไม่ว่าวิธีการสร้างกรณีทดสอบลักษณะใดก็อาจเกิดปัญหาเวลาที่มีไม่พอเพียงกับการทดสอบกรณีทดสอบทั้งหมดได้ งานวิจัยนี้นำเสนอการใช้การจัดลำดับความสำคัญ (Prioritization) สำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อให้ผู้ทดสอบมีทางเลือกในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ โดยสามารถปรับให้เหมาะสมกับระยะเวลาและทรัพยากรที่มีอยู่ นอกจากนี้ยังได้ศึกษาเปรียบเทียบประสิทธิภาพของวิธีการจัดลำดับความสำคัญแบบต่างๆ เพื่อให้สามารถนำวิธีการเหล่านี้ไปใช้ได้อย่างเหมาะสม

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อหาวิธีการในการจัดลำดับความสำคัญกรณีทดสอบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

1.2.2 เพื่อเปรียบเทียบประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

1.3 ขอบเขตของการวิจัย

1.3.1 ออกแบบวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ โดยใช้หลักการของความครอบคลุมในการจัดลำดับความสำคัญกรณีทดสอบ

1.3.2 พัฒนาเครื่องมือที่ช่วยในการจัดลำดับความสำคัญกรณีทดสอบ ที่ทำงานได้อย่างถูกต้อง

1.3.3 ทดสอบวิธีการที่ออกแบบกับกรณีทดสอบที่ถูกสร้างขึ้นกับโปรแกรมที่ทำงานบนระบบปฏิบัติการวินโดว์ (Windows) ตั้งแต่รุ่น XP ขึ้นไป โดยใช้วิธีสร้างกรณีทดสอบโดยการทดลองใช้งานเป็นหลัก ร่วมกับการสร้างกรณีทดสอบอัตโนมัติจากเครื่องมือ

1.3.4 ทดสอบกับตัวอย่างกรณีศึกษาอย่างน้อย 10 ตัวอย่าง โดยแต่ละตัวอย่างมีกรณีทดสอบอย่างน้อย 50 กรณีทดสอบ

1.3.5 สรุปเปรียบเทียบผลลัพธ์ที่ได้จากการทดสอบในด้านประสิทธิภาพ ได้แก่ ความเร็วในการค้นหาข้อผิดพลาด รวมถึงระยะเวลาที่ใช้ในการจัดลำดับความสำคัญและทดสอบกรณีทดสอบ

1.4 ขั้นตอนการดำเนินการวิจัย

1.4.1 ศึกษาวิธีการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ หลักการ และข้อจำกัด รวมถึงงานวิจัยที่เกี่ยวข้อง

1.4.2 ศึกษาวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับซอฟต์แวร์ปกติ และงานวิจัยที่เกี่ยวข้อง

1.4.3 ออกแบบวิธีการจัดลำดับความสำคัญกรณีทดสอบเพื่อใช้กับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

1.4.4 ออกแบบการทดสอบเพื่อเปรียบเทียบประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบที่ได้ออกแบบไว้

1.4.5 ศึกษาเครื่องมือสร้างกรณีทดสอบสำหรับทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ และเครื่องมือที่ใช้ทดสอบกรณีทดสอบ แล้วเลือกเครื่องมือที่เหมาะสมที่จะใช้ในการทดสอบ

1.4.6 สร้างเครื่องมือจัดลำดับความสำคัญกรณีทดสอบตามวิธีการที่ออกแบบไว้ และสามารถทำงานกับเครื่องมือที่เลือกไว้ได้

1.4.7 ทดสอบการทำงานของเครื่องมือ

1.4.8 ทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบต่างๆตามที่ได้ออกแบบการทดสอบไว้

1.4.9 นำผลการทดสอบที่ได้มาเปรียบเทียบ สรุปผล

1.4.10 จัดทำรายงานวิทยานิพนธ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 วิธีการจัดลำดับความสำคัญกรณีทดสอบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งสามารถลดระยะเวลาและทรัพยากรที่ใช้ในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ลง โดยอาศัยการจัดลำดับความสำคัญกรณีทดสอบ เพื่อผู้ทดสอบสามารถเลือกทดสอบเพียงบางส่วนได้

1.5.2 เครื่องมือที่ช่วยจัดลำดับความสำคัญกรณีทดสอบ ตามเกณฑ์ที่ได้นำเสนอในงานวิจัยนี้

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ บทที่ 3 กล่าวถึงแนวคิด วิธีการในการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ บทที่ 4 กล่าวถึงการออกแบบและพัฒนาเครื่องมือที่ใช้จัดลำดับความสำคัญกรณีทดสอบ บทที่ 5 กล่าวถึงรายละเอียดวิธีการทดสอบและผลการทดสอบ และบทที่ 6 เป็นข้อสรุปและข้อเสนอแนะจากการวิจัย

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Test Case Prioritization for GUI Testing” โดย Yachai Limpiyakorn and Pisak Kurusathian ในงานประชุมวิชาการ “The 2005 International Conference on Software Engineering Research and Practice (SERP’05)” ณ รัฐเนวาดา ประเทศสหรัฐอเมริกา ระหว่างวันที่ 27-29 กรกฎาคม พ.ศ. 2548

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ส่วนต่อประสานกราฟิกกับผู้ใช้

ส่วนต่อประสานผู้ใช้ (User Interface: UI) คือ ส่วนที่ทำหน้าที่เป็นสื่อกลางในการติดต่อสื่อสารระหว่างโปรแกรมกับผู้ใช้งาน โดยทำหน้าที่รับและแสดงผลข้อมูลในรูปแบบที่แต่ละฝ่ายสามารถเข้าใจและนำไปใช้งานได้

สำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ มีหน้าที่การทำงานเช่นเดียวกับส่วนต่อประสานผู้ใช้ แต่เพิ่มความสะดวกให้กับผู้ใช้งานในการติดต่อสื่อสารกับโปรแกรม โดยผ่านทางภาพและสัญลักษณ์ต่างๆ เช่น ใช้เมาส์กดเลือกสัญรูป (icon) หรือการเลือกคำสั่งตามรายการเลือกของระบบเมนู แทนการพิมพ์คำสั่งดังแต่ก่อน การใช้ภาพเป็นตัวประสานกับผู้ใช้ได้รับการยอมรับว่า มีประสิทธิภาพและผู้ใช้ทำความเข้าใจได้ง่าย

การทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ เป็นการทำงานโดยอาศัยกลไกของเหตุการณ์ (event) เป็นหลัก โดยการกระทำต่างๆของผู้ใช้จะสร้างเหตุการณ์ที่มีลักษณะเฉพาะ ซึ่งเหตุการณ์เหล่านี้จะทำหน้าที่ขับเคลื่อนระบบให้ทำงานตามที่กำหนดไว้

ส่วนต่อประสานกราฟิกกับผู้ใช้จะทำงาน เมื่อมีผู้ใช้เมาส์หรือแผงแป้นอักขระกระทำต่อมัน โดยจะสร้างอินสแตนซ์ (instance) หรือตัวแทนของเหตุการณ์ขึ้นตามประเภทของการกระทำนั้น จากนั้นตัวแทนเหล่านี้จะถูกส่งให้แก่คอมโพเนนท์ที่เกี่ยวข้องเพื่อรับไปจัดการที่เรียกว่า การจัดการเหตุการณ์ (event handler) เช่น ตรวจสอบว่าเป็นเหตุการณ์ประเภทใดแล้วตอบสนองให้เหมาะสมกับเหตุการณ์นั้น

เหตุการณ์ที่เกิดกับส่วนติดต่อกับผู้ใช้งานแบบกราฟิกแบ่งออกได้เป็น 2 ประเภทได้แก่

- เหตุการณ์ระดับล่าง (low-level events) คือเหตุการณ์ที่เกิดขึ้นในระดับพื้นฐาน ยังไม่สามารถแปลความหมายได้อย่างสมบูรณ์ เช่น การกดปุ่มซ้ายของเมาส์ เป็นต้น
- เหตุการณ์ระดับสูง (high-level events) หรือเหตุการณ์ที่มีความหมาย (semantic events) ยกตัวอย่างเช่น การกดปุ่ม (button) หรือการเลือกไอเทม เป็นต้น ซึ่งเหตุการณ์เหล่านี้มีข้อมูลบอกว่าคอมโพเนนท์ใดเกิดอะไรขึ้น

โดยลำดับในการสร้างเหตุการณ์ต่างๆเป็นไปความต้องการของผู้ใช้ และเหตุการณ์ต่างๆจะมีความหมายแตกต่างกันไปตามสถานะของโปรแกรมในขณะนั้น

2.2 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ [4] คือกระบวนการที่ใช้ตรวจสอบว่า ซอฟต์แวร์ทำงานเป็นไปตามความต้องการซอฟต์แวร์ (Software Requirements) ที่ได้กำหนดไว้หรือไม่

ในการทดสอบซอฟแวร์ เราจำเป็นต้องใช้กรณีทดสอบ เพื่อทดสอบการทำงานของซอฟต์แวร์ โดยกรณีทดสอบมีประกอบด้วย

1) ข้อมูลนำเข้า (Inputs) แบ่งออกเป็น 2 ประเภท ได้แก่
   1. เงื่อนไขเริ่มต้น (Preconditions) คือ สิ่งที่ต้องปฏิบัติก่อนเริ่มการทดสอบ
   2. ข้อมูลทดสอบ (Actual inputs) คือ ข้อมูลนำเข้าที่ใช้ในการทดสอบ
2) ผลลัพธ์ที่คาดการณ์ไว้ (Expected outputs) แบ่งออกเป็น 2 ประเภท ได้แก่
   - เงื่อนไข (Post conditions) คือ สิ่งที่คาดว่าจะเกิดขึ้นเมื่อการทดสอบเสร็จสิ้น
   - ผลลัพธ์ (Actual outputs) คือ ผลลัพธ์ที่คาดว่าจะได้รับจากการทดสอบ

กรณีทดสอบเหล่านี้เมื่อรวมเข้าด้วยกัน เรียกว่า ชุดทดสอบ (test suite)

2.3 การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ คือกระบวนการที่ใช้ตรวจสอบว่า ส่วนต่อประสานกราฟิกกับผู้ใช้ทำงานได้ถูกต้องตรงตามที่ได้ออกแบบไว้หรือไม่ ซึ่งมีหลักการและโครงสร้างใกล้เคียงกับการทดสอบซอฟต์แวร์ปกติ แต่การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้มีความซับซ้อนกว่าการทดสอบซอฟต์แวร์ตามปกติทั่วไป เนื่องจากลักษณะเฉพาะของส่วนต่อประสานกราฟิกกับผู้ใช้

2.3.1 ปัญหาที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

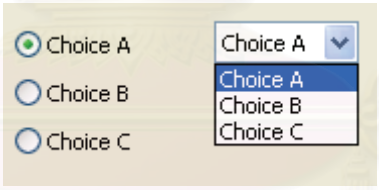
ลักษณะพิเศษของส่วนต่อประสานกราฟิกกับผู้ใช้นำมาซึ่งปัญหาในการทดสอบ [5] ซึ่งกรณีที่สำคัญ ได้แก่

1) ความแตกต่างระหว่างกลไกทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้กับส่วนประกอบอื่นของซอฟต์แวร์

ส่วนต่อประสานกราฟิกกับผู้ใช้มีกลไกการทำงานที่แตกต่างกับส่วนของโปรแกรม จึงยากที่จะเข้าใจได้โดยพิจารณาจากรหัสต้นทาง (source code) ของโปรแกรมแต่เพียงอย่างเดียว เนื่องจากส่วนใหญ่ส่วนต่อประสานกราฟิกกับผู้ใช้อาศัยเทคนิคของการพัฒนาโปรแกรม

เชิงวัตถุ (Object Oriented Programming : OOP) ที่มีลักษณะของการสืบทอด (inheritance) และการเก็บซ่อนข้อมูล (encapsulation) มาใช้ โดยอาศัยการสืบทอด หรือเรียกใช้คอมโพเนนท์สำเร็จรูปต่างๆ รวมถึงในส่วนของระบบปฏิบัติการ (Operating System : OS) ตัวแปลภาษาโปรแกรม และไลบรารีสำเร็จรูปต่างๆ อาจเข้ามามีส่วนเกี่ยวข้องด้วย ทำให้วิธีการและข้อกำหนดในการทดสอบต่างๆที่อาศัยโครงสร้างโปรแกรมไม่สามารถใช้ได้ รวมถึงการคาดการณ์ผลลัพธ์การทำงานทำได้ยาก

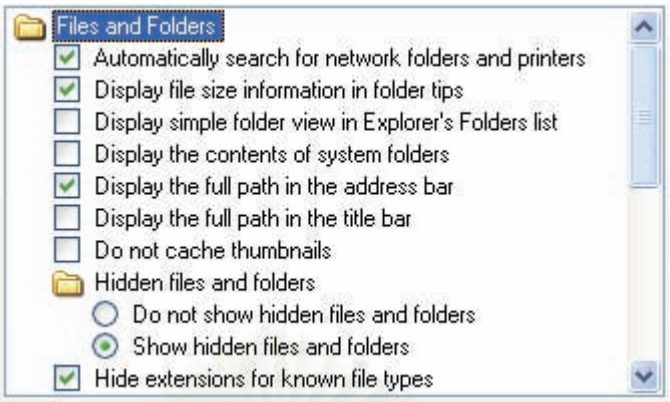
นอกจากนี้ส่วนต่อประสานกราฟิกกับผู้ใช้ยังมีลักษณะที่แตกต่างไปจากส่วนการออกแบบโปรแกรม ซึ่งจะมีรายละเอียดขั้นตอนการทำงานที่ชัดเจนกว่า รวมทั้งเอกสารการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ยังไม่มีมาตรฐานที่ชัดเจน แม้แต่เอกสารยูเอ็มแอล (Universal Modeling Language : UML) ก็มีข้อมูลที่เกี่ยวกับส่วนต่อประสานกราฟิกกับผู้ใช้น้อยมาก ตัวอย่างที่เห็นได้ชัด คือ กรณีเอกสารการออกแบบเดียวกัน หากนำไปให้นักพัฒนาซอฟต์แวร์หลายฝ่ายออกแบบส่วนต่อประสานกับผู้ใช้ ย่อมได้ส่วนต่อประสานผู้ใช้ที่แตกต่างกันไป ตัวอย่างเช่นรูป 2.1 ผู้ออกแบบสามารถเลือกใช้ส่วนต่อประสานกราฟิกกับผู้ใช้ในการรับข้อมูลจากผู้ใช้ได้ทั้ง 2 แบบ ซึ่งทำหน้าที่รับข้อมูลได้ไม่แตกต่างกัน ดังนั้นจึงเป็นการยากที่จะทดสอบโดยอาศัยเอกสารการออกแบบเพียงอย่างเดียว



รูปที่ 2.1 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ 2 แบบ ที่ทำหน้าที่เหมือนกัน

2) รูปแบบที่เป็นไปได้ในการทำงาน

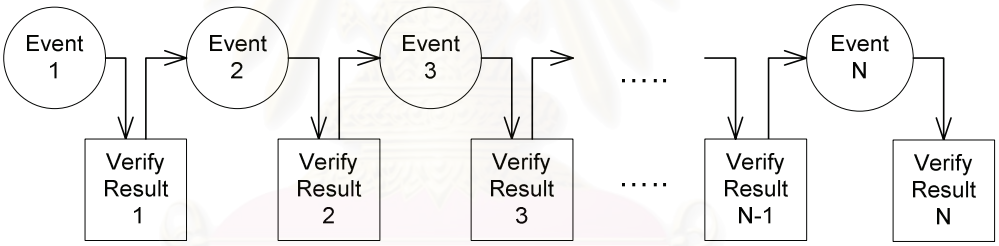
ส่วนต่อประสานกราฟิกกับผู้ใช้มีรูปแบบการทำงานที่เป็นไปได้จำนวนมาก เนื่องจากส่วนต่อประสานกราฟิกกับผู้ใช้บางประเภท ประกอบไปด้วยทางเลือกที่เป็นไปได้จำนวนมาก ดังตัวอย่างในรูป 2.2 นอกจากนี้เหตุการณ์แต่ละเหตุการณ์จะมีความหมายแตกต่างกันไปขึ้นกับลำดับเหตุการณ์ก่อนหน้า ทำให้เกิดรูปแบบความเป็นไปได้ของลำดับเหตุการณ์เป็นจำนวนมหาศาล ซึ่งยากที่จะสร้างกรณีทดสอบให้ครอบคลุมครบถ้วนทุกลำดับเหตุการณ์ นอกจากนี้ผู้ใช้ยังสามารถสร้างเหตุการณ์ที่ไม่ได้ออกแบบไว้ ซึ่งอาจส่งผลกระทบถึงการทำงานของโปรแกรมได้



รูปที่ 2.2 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ที่มีทางเลือกในการทำงานจำนวนมาก

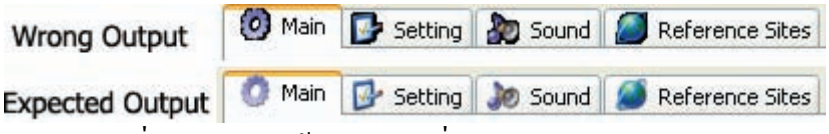
3) การตรวจสอบผลลัพธ์ของการทดสอบทำได้ยาก

ส่วนต่อประสานกราฟิกกับผู้ใช้มีลักษณะการทำงานตอบสนองกับผู้ใช้เป็นขั้นตอนตามลำดับ ดังนั้นจึงต้องตรวจสอบผลลัพธ์ในทุกๆขั้นของการทดสอบ จึงจะสามารถยืนยันได้ว่าส่วนต่อประสานกราฟิกกับผู้ใช้ทำงานได้ถูกต้องจริง ดังตัวอย่างในรูป 2.3



รูปที่ 2.3 การตรวจสอบผลลัพธ์ในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

นอกจากนี้ส่วนต่อประสานกราฟิกกับผู้ใช้ยังประกอบไปด้วยองค์ประกอบย่อยเป็นจำนวนมาก เช่น ปุ่ม เมนู เป็นต้น ซึ่งองค์ประกอบย่อยเหล่านี้มีคุณสมบัติหลายด้านที่ต้องตรวจสอบ เช่น ตำแหน่ง สีสัน ขนาดและรูปแบบตัวอักษร สีพื้น ข้อความ ฯลฯ รวมถึงการแสดงผลในลักษณะของสัญรูปหรือรูปภาพ ดังตัวอย่างในรูป 2.4 ซึ่งคุณสมบัติเหล่านี้จำเป็นต้องถูกตรวจสอบอย่างละเอียด เพื่อยืนยันว่าการตอบสนองต่อผู้ใช้เป็นไปอย่างถูกต้อง ไม่มีส่วนที่แสดงผลขาดหรือเกินไปจากที่กำหนดไว้



รูปที่ 2.4 ตัวอย่างข้อผิดพลาดเกี่ยวกับการแสดงผลสัญรูป

สืบเนื่องจากข้อ 2 การมีรูปแบบความเป็นไปได้ในการทำงานที่หลากหลาย เมื่อรวมกับรายละเอียดของผลลัพธ์ในแต่ละขั้นตอนการทำงานแล้ว ทำให้เป็นเรื่องยากและต้องใช้ทรัพยากรเป็นจำนวนมาก ในการที่จะออกแบบและกำหนดรูปแบบการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ได้อย่างครบถ้วนสมบูรณ์ ครอบคลุมถึงกรณีที่อาจเกิดขึ้นได้ทั้งหมด

4) ส่วนต่อประสานกับผู้ใช้มีการเปลี่ยนแปลงอยู่เสมอ

รูปแบบส่วนต่อประสานกราฟิกกับผู้ใช้จะมีการเปลี่ยนแปลงไปเรื่อยๆตลอดการพัฒนา เพื่อค้นหารูปแบบที่ตรงตามความต้องการและอำนวยความสะดวกต่อผู้ใช้มากที่สุด ดังนั้นจึงมักพัฒนาโดยวิธีการสร้างโปรแกรมทดลองอย่างรวดเร็ว (rapid prototyping) ทำให้ส่วนต่อประสานกราฟิกกับผู้ใช้มีการเปลี่ยนแปลงอยู่เสมอ ซึ่งทำให้เอกสารการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ไม่ละเอียดครบถ้วน ทำให้การทดสอบตามปกติและการทดสอบแบบรีเกรสชัน* (regression testing) ทำได้ยาก นอกจากนี้การเปลี่ยนแปลงรูปแบบส่วนต่อประสานกราฟิกกับผู้ใช้ยังอาจส่งผลกระทบทำให้กรณีทดสอบที่สร้างขึ้นก่อนหน้าไม่สามารถนำกลับมาใช้ทดสอบได้

2.3.2 วิธีการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

จากปัญหาข้างต้น ทำให้การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เป็นกระบวนการที่ใช้ทรัพยากรในการทดสอบเป็นจำนวนมาก ซึ่งปัจจุบันในการพัฒนาซอฟต์แวร์ส่วนใหญ่ยังไม่มีขั้นตอนและกระบวนการทดสอบที่ชัดเจนและมีประสิทธิภาพ

แม้ว่าจะมีงานวิจัยเป็นจำนวนมาก [1] [2] [3] [6] [7] [8] [9] เกี่ยวกับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ แต่ส่วนใหญ่ยังคงอยู่ในขั้นตอนการวิจัย และยังไม่มีวิธีการที่ได้รับการยอมรับเป็นมาตรฐานในเชิงอุตสาหกรรม ทั้งนี้เพราะการทดสอบมีความซับซ้อนและมีลักษณะเฉพาะ ซึ่งต้องใช้ทั้งความรู้ แรงงาน และระยะเวลาในการดำเนินการ

วิธีการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในปัจจุบัน สามารถแบ่งออกเป็นแนวทางหลักๆ ได้ดังนี้

---

* Regression Testing คือ การทวนสอบใดๆ (ซึ่งปกติจะเกิดขึ้นหลังจากมีการเปลี่ยนแปลงซอฟต์แวร์หรือข้อมูล) ที่มีจุดประสงค์ในการแสดงว่าพฤติกรรมของซอฟต์แวร์นั้นๆ ไม่มีการเปลี่ยนแปลงจากเดิม ยกเว้นกรณีที่ต้องการให้มีการเปลี่ยนแปลงในส่วนนั้นขึ้น [14]

1) การทดสอบโดยมนุษย์

การทดสอบใช้งานโดยมนุษย์ ซึ่งปัจจุบันการพัฒนาซอฟต์แวร์เชิงอุตสาหกรรมส่วนใหญ่ยังคงใช้วิธีนี้อยู่ ข้อดีของวิธีนี้ คือ สามารถทดสอบได้ง่าย ไม่ต้องเตรียมการหรือใช้ข้อมูลการออกแบบมาก แต่จะใช้เวลาในการทดสอบนาน และไม่สามารถบอกได้ว่าทดสอบเท่าใดจึงจะพอเพียง แนวทางการทดสอบแบ่งออกเป็น

- การทดสอบโดยการทดลองใช้งาน (Walkthrough) คือ การทดสอบโดยการทดลองใช้งานโปรแกรมในส่วนการทำงานหลัก เพื่อหาข้อผิดพลาด
- การทดสอบโดยการสังเกตการทำงานของผู้ใช้ (User Observation) คือ การทดสอบที่ให้ผู้ใช้ได้ทดลองใช้งานจริงภายใต้สภาพแวดล้อมที่กำหนด แล้วสังเกตปัญหา ข้อผิดพลาดที่ผู้ใช้พบ
- การทดสอบโดยการแจกจ่ายชุดทดสอบ (Beta Copies Distribution) คือ การทดสอบที่แจกจ่ายชุดทดสอบให้กับบุคคลที่สนใจไปทดลองใช้ และรายงานข้อผิดพลาดกลับมา

2) การทดสอบโดยเลียนแบบมนุษย์

การทดสอบที่พยายามลดภาระในการทดสอบของมนุษย์ลง โดยเลียนแบบพฤติกรรมการทดสอบของมนุษย์ เพื่อให้ทดสอบได้มากขึ้นและเร็วขึ้น แต่ก็ยังคงมีข้อเสียในเรื่องความเพียงพอในการทดสอบ (coverage criteria) แนวทางการทดสอบแบ่งออกเป็น

- การทดสอบโดยเครื่องมือบันทึกและเล่นซ้ำ (Capture and Replay) ส่วนใหญ่ใช้ในการทดสอบแบบรีเกรสชัน มีลักษณะการทำงานคือ เครื่องมือจะบันทึกลำดับของเหตุการณ์และผลลัพธ์ของการทดสอบ เพื่อนำกลับมาใช้ทดสอบใหม่ เมื่อมีการเปลี่ยนแปลงแก้ไข และอาจนำชุดทดสอบไปใช้กับโปรแกรมที่มีลักษณะการทำงานคล้ายกันได้
- การทดสอบโดยใช้อัลกอริทึมเชิงพันธุกรรม (Genetic Algorithm) [7] เป็นการทดสอบโดยพยายามที่จะเพิ่มความแปรผันให้กับชุดทดสอบ เพื่อให้ได้ชุดทดสอบใหม่ๆที่แตกต่างไปจากชุดทดสอบเดิม
- การทดสอบโดยการสุ่ม (Random testing) [8] เป็นการทดสอบโดยการสุ่มสร้างเหตุการณ์ต่างๆขึ้น เพื่อตรวจสอบผลลัพธ์ที่ได้

3) การทดสอบโดยอัตโนมัติ

คือ การทดสอบที่ใช้โปรแกรมในการสร้างและทดสอบชุดทดสอบ โดยเน้นความครบถ้วนของชุดทดสอบ ให้ครอบคลุมถึงทุกกรณีที่เป็นไปได้ พิจารณาตามข้อกำหนดในการออกแบบ ซึ่งข้อดีคือ สามารถบอกได้ชัดเจนว่าทดสอบเท่าใดจึงจะพอเพียง แต่ต้องใช้ข้อมูลการออกแบบที่ชัดเจน และมีกรณีทดสอบจำนวนมาก

- การทดสอบโดยใช้เครื่องสถานะ (State machine testing) [6] เป็นการทดสอบโดยพิจารณาระบบในรูปแบบของเครื่องสถานะ เพื่อตรวจสอบว่าตรงกับการออกแบบหรือไม่ ซึ่งจะมีจำนวนกรณีทดสอบมาก
- การทดสอบโดยการกำหนดเป้าหมาย (Goal-driven testing) [2] เป็นการทดสอบการทำงานของระบบแต่ละหน้าที่ว่าทำงานได้ถูกต้องครบถ้วนตามที่ได้ออกแบบไว้หรือไม่ โดยใช้กราฟการทำงานเพื่อกำหนดชุดทดสอบทั้งหมดที่เป็นไปได้ ซึ่งจะลดจำนวนกรณีทดสอบลง

4) การทดสอบโดยใช้วิธีการฟอร์มอล [9]

คือ การทดสอบโดยแปลงการออกแบบให้อยู่ในรูปแบบเชิงคณิตศาสตร์ เพื่อนำไปตรวจสอบกับชิ้นงาน เป็นวิธีที่ซับซ้อนและต้องอาศัยผู้เชี่ยวชาญ มักใช้ในระบบที่มีความสำคัญสูง และมีการกำหนดส่วนต่อประสานกราฟิกกับผู้ใช้ที่ชัดเจน

2.4 การจัดลำดับความสำคัญกรณีทดสอบ

การจัดลำดับความสำคัญกรณีทดสอบ คือ การนำกรณีทดสอบในชุดทดสอบหนึ่งๆมาจัดลำดับความสำคัญ โดยมีเป้าหมายในการจัดลำดับแตกต่างกันไป เช่น ต้องการให้ทดสอบครอบคลุมส่วนหนึ่งส่วนใดของโปรแกรมมากๆ ต้องการให้สามารถทดสอบพบข้อผิดพลาด (defect) ได้อย่างรวดเร็ว ต้องการให้ทดสอบพบข้อผิดพลาดที่มีความสำคัญสูงก่อน หรือต้องการเน้นการทดสอบไปที่ส่วนที่ถูกเปลี่ยนแปลงแก้ไขล่าสุด เป็นต้น

สำหรับซอฟต์แวร์ปกติทั่วไป การจัดลำดับความสำคัญกรณีทดสอบจะมีประโยชน์ในกรณีที่ทรัพยากรหรือเวลาในการทดสอบมีจำกัด หรือในกรณีการทดสอบแบบรีเกรสชัน เพื่อลดระยะเวลาและทรัพยากรที่ใช้ในการทดสอบ ซึ่งอาจกล่าวอีกอย่างหนึ่งว่า เพื่อลดค่าใช้จ่ายในการทดสอบนั่นเอง

นอกจากประโยชน์ดังกล่าวแล้ว สำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้อาจเกิดกรณีทดสอบที่สร้างขึ้นมีจำนวนมากเกินไป อันเป็นผลจากวิธีการสร้างกรณีทดสอบ หรือการทดสอบจำเป็นต้องใช้ทรัพยากรจำนวนมาก เนื่องจากลักษณะเฉพาะของส่วนต่อประสานกราฟิกกับผู้ใช้ดังที่กล่าวมาแล้ว ทำให้ไม่สามารถทดสอบได้ทั้งหมดหรือไม่คุ้มค่ากับการทดสอบทั้งหมด การจัดลำดับความสำคัญสามารถช่วยลดจำนวนกรณีทดสอบ โดยผู้ทดสอบจะคัดกรองกรณีทดสอบที่สำคัญออกมา ทำการทดสอบเฉพาะส่วนที่สำคัญ เพื่อลดภาระในการทดสอบ

2.4.1 ปัจจัยที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบ

เนื่องจากความสามารถในการค้นพบข้อผิดพลาดของกรณีทดสอบจะทราบได้ต่อเมื่อมีการทดสอบแล้วเท่านั้น ดังนั้นในการจัดลำดับความสำคัญกรณีทดสอบ ซึ่งต้องทำก่อนการทดสอบ จึงเป็นความพยายามในการประเมินความสามารถในการค้นพบข้อผิดพลาดของกรณีทดสอบผ่านทางปัจจัยอื่นๆ โดยวิธีการในการจัดลำดับความสำคัญกรณีทดสอบจะมีวิธีต่างๆกันไป แบ่งออกเป็นแนวทางหลักๆได้ ดังนี้ [10]

1) การจัดลำดับความสำคัญโดยพิจารณาจากปัจจัยภายในกรณีทดสอบ (General test case prioritization)

สำหรับการจัดลำดับความสำคัญลักษณะนี้ เป็นการจัดลำดับความสำคัญโดยอาศัยปัจจัยที่มีอยู่ในกรณีทดสอบเท่านั้น แนวทางหลักๆแบ่งออกเป็น

- การจัดลำดับความสำคัญโดยใช้ความครอบคลุม (coverage) เกณฑ์ความครอบคลุมเป็นกฎที่ใช้พิจารณาว่าการทดสอบซอฟต์แวร์ครบถ้วนหรือไม่ ซึ่งกฎเหล่านี้มีลักษณะแตกต่างกันไป ที่รู้จักดีในการทดสอบซอฟต์แวร์ตามปกติ ได้แก่ เกณฑ์ความครอบคลุมระดับคำสั่ง (statement coverage) เกณฑ์ความครอบคลุมระดับทางเลือก (branch coverage) และเกณฑ์ความครอบคลุมระดับเส้นทาง (path coverage) นั่นเอง นอกจากเกณฑ์ความครอบคลุมดังกล่าวแล้ว เราอาจกำหนดเกณฑ์ความครอบคลุมได้อีกหลายแบบ เช่น ระดับฟังก์ชัน (function coverage) หรือระดับข้อมูล (data coverage) เป็นต้น

  เกณฑ์ความครอบคลุมสามารถนำไปประยุกต์ใช้กับการทดสอบได้หลายรูปแบบ เช่น เป็นแนวทางสำหรับการสร้างกรณีทดสอบ เป็นต้น สำหรับการจัดลำดับความสำคัญกรณีทดสอบสามารถใช้เกณฑ์ความครอบคลุมเป็นแนวทางในการจัดลำดับความสำคัญ โดยเปรียบเทียบว่ากรณีทดสอบใดตรงกับเกณฑ์ความครอบคลุมมากกว่ากัน

- การจัดลำดับความสำคัญโดยพิจารณาจากปัจจัยอื่นๆ เช่น ทรัพยากรที่ต้องใช้ในการทดสอบกรณีทดสอบนั้นๆ เป็นต้น

2) การจัดลำดับความสำคัญโดยพิจารณาจากข้อมูลอื่น (Version-specific test case prioritization)

ใช้ข้อมูลความสามารถในการค้นพบข้อผิดพลาดของกรณีทดสอบจากการทดสอบที่มีลักษณะใกล้เคียงกัน วิธีนี้แบ่งออกเป็น

- การใช้ข้อมูลในอดีต ใช้ในการทดสอบแบบรีเกรสชัน โดยใช้ข้อมูลจากการทดสอบครั้งก่อนเป็นแนวทาง
- การใช้ข้อมูลจากทดลอง โดยการทดลองใส่ข้อผิดพลาดในซอฟต์แวร์ แล้วใช้ชุดทดสอบค้นหา เช่น การทดสอบแบบมิวเตชัน (mutation testing) แล้วเก็บสถิติมาใช้ในการจัดลำดับความสำคัญ
- การใช้ข้อมูลจากมาตรวัดซอฟต์แวร์ (software metric) กำหนดมาตรวัดขึ้นเพื่อวัดค่าความสามารถในการค้นหาข้อผิดพลาดของกรณีทดสอบ ซึ่งมาตรวัดเหล่านี้ประกอบด้วยปัจจัยต่างๆของกรณีทดสอบและซอฟต์แวร์ เช่น จำนวนฟังก์ชันที่ทดสอบ ส่วนของฟังก์ชันที่มีการแก้ไข เป็นต้น

ปัจจัยต่างๆข้างต้นสามารถนำมาใช้ร่วมกัน ทำให้เกิดวิธีจัดลำดับที่แตกต่างออกไปได้ นอกจากนี้มีอีกหลายปัจจัยที่ทำให้วิธีการจัดลำดับความสำคัญแตกต่างออกไป เช่น ระดับการทำงาน (granularity) การคำนวณโดยใช้ผลย้อนกลับ (feedback) เป็นต้น

2.4.2 วิธีการคำนวณการจัดลำดับความสำคัญกรณีทดสอบ

สำหรับวิธีการคำนวณแบ่งออกได้เป็น 2 แนวทาง ได้แก่

1) การจัดลำดับความสำคัญแบบเบ็ดเสร็จ (Total) เป็นวิธีการที่ไม่คำนึงถึงผลการจัดลำดับก่อนหน้า โดยจะพิจารณาจัดลำดับตามปัจจัยที่พบในกรณีทดสอบนั้นๆเท่านั้น

2) การจัดลำดับความสำคัญแบบเพิ่ม (Additional) เป็นวิธีการที่นำผลย้อนกลับจากการจัดลำดับก่อนหน้ามาร่วมในการคำนวณ โดยพิจารณาเฉพาะส่วนของกรณีทดสอบที่มีความแตกต่างจากกลุ่มกรณีทดสอบที่ได้รับการจัดลำดับไปแล้วก่อนหน้า ซึ่งก็คือกลุ่มของกรณีทดสอบที่มีระดับความสำคัญมากกว่านั่นเอง ซึ่งวิธีการนี้ต้องมีการคำนวณใหม่ทุกครั้งที่มีการเลือกกรณีทดสอบลำดับถัดไป

2.5 งานวิจัยที่เกี่ยวข้อง

2.5.1 งานวิจัย “Prioritizing Test Cases For Regression Testing” โดย Gregg Rothermel, Roland H. Untch, Chengyun Chu และ Mary Jean Harrold [10]

งานวิจัยนี้มีเป้าหมายในการเปรียบเทียบประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบซอฟต์แวร์ปกติจำนวน 6 วิธี โดยทดสอบกับโปรแกรมจำนวน 15 โปรแกรม ซึ่งเน้นที่การทดสอบแบบรีเกรสชัน โดยทดลองและวิเคราะห์ผลที่ได้ นอกจากนี้ยังได้กำหนดเมตริกซ์ APFD (weighted Average of the Percentage of Faults Detected) ขึ้น เพื่อวัดประสิทธิภาพของการจัดลำดับความสำคัญ

ผู้วิจัยจึงได้นำแนวคิดเกี่ยวกับการจัดลำดับความสำคัญกรณีทดสอบและการเปรียบเทียบประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบซอฟต์แวร์ปกติจากงานวิจัยนี้ มาปรับใช้กับการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

2.5.2 งานวิจัย “Coverage Criteria for GUI Testing” โดย Atif M. Memon, Mary Lou Soffa, และ Martha E. Pollack [11]

งานวิจัยนี้นำเสนอแนวทางในการกำหนดเกณฑ์ความครอบคลุม สำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ โดยพิจารณาจากความหมายของเหตุการณ์และความสัมพันธ์ระหว่างคอมโพเนนท์เป็นหลัก นอกจากนี้ยังนำเสนอโครงสร้างข้อมูลและวิธีการสำหรับการหาความครอบคลุมทั้งหมด โดยนำเสนอความครอบคลุมแบ่งออกเป็นหลายระดับ ทั้งในระดับความครอบคลุมภายในคอมโพเนนท์ (Intra-component criteria) และระดับความครอบคลุมระหว่างคอมโพเนนท์ (Inter-component criteria) นอกจากนี้ยังแสดงตัวอย่างการนำไปใช้คำนวณหาความครอบคลุมของส่วนต่อประสานกราฟิกกับผู้ใช้ด้วย

ความครอบคลุมที่นำเสนอในงานวิจัยนี้ถือเป็นรากฐานแนวคิดเกี่ยวกับความครอบคลุมสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งผู้วิจัยได้นำมาปรับใช้ต่อไป

บทที่ 3

วิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

เนื่องจากปัญหาที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ ดังที่ได้กล่าวถึงในหัวข้อ 2.3.1 บทที่ 2 ซึ่งนำมาซึ่งความยุ่งยาก ซับซ้อน และสิ้นเปลืองทรัพยากรในการทดสอบ งานวิจัยนี้จึงได้เสนอแนวคิดการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อให้ผู้ทดสอบสามารถเลือกกรณีทดสอบเพียงบางส่วนจากกรณีทดสอบทั้งหมดมาทดสอบ โดยพิจารณาจากลำดับความสำคัญ เพื่อให้สามารถปรับการทดสอบให้เข้ากับทรัพยากรที่มีอยู่อย่างจำกัด และบริหารทรัพยากรให้ได้รับประโยชน์สูงสุด โดยไม่ส่งผลกระทบถึงผลการทดสอบมากเกินไป

3.1 ขั้นตอนการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

งานวิจัยนี้มุ่งความสนใจไปที่การทดสอบการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ที่มีการกำหนดกรณีทดสอบอย่างชัดเจนแน่นอน ซึ่งเป็นการทดสอบภายในสภาวการณ์ที่กำหนด การทดสอบในลักษณะนี้ ประกอบด้วนส่วนต่างๆที่ทำหน้าที่แตกต่างกัน ได้แก่

3.1.1 ส่วนสร้างกรณีทดสอบ (Test Case Generator) ทำหน้าที่สร้างกรณีทดสอบตามวิธีการที่กำหนดไว้ ซึ่งวิธีการส่วนใหญ่ใช้แนวทางจากข้อกำหนดของส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI Specifications) ในการสร้างกรณีทดสอบ

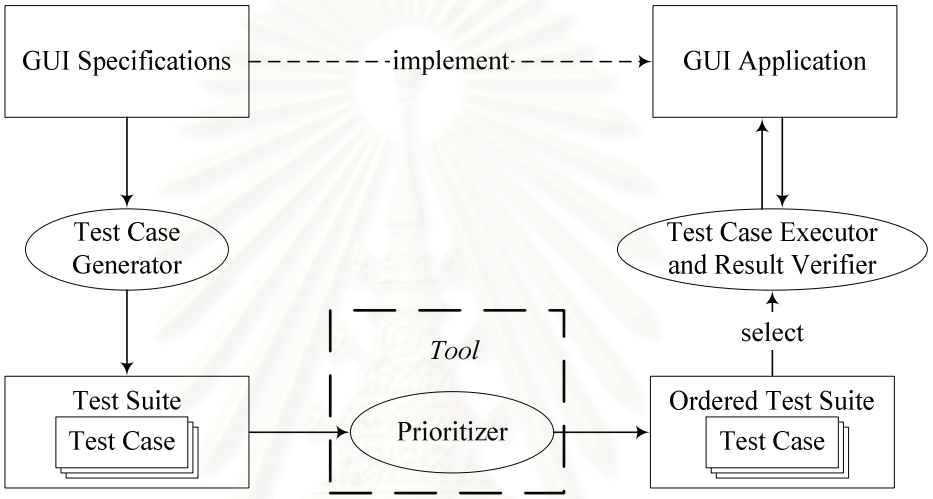
3.1.2 ส่วนประมวลผลกรณีทดสอบ (Test Case Executor) ทำหน้าที่ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ตามขั้นตอนในกรณีทดสอบที่ได้รับ

3.1.3 ส่วนตรวจสอบผลลัพธ์ (Result Verifier) ทำงานประสานกับส่วนทดสอบ โดยจะเปรียบเทียบผลลัพธ์ที่ได้จากการทดสอบกับผลลัพธ์ที่คาดหวัง เพื่อตรวจสอบว่าผลการทดสอบถูกต้องหรือไม่

3.1.4 ส่วนจัดลำดับความสำคัญกรณีทดสอบ (Prioritizer) เป็นส่วนเพิ่มเติมที่ถูกนำเสนอในงานวิจัยนี้ มีจุดประสงค์เพื่อเพิ่มประสิทธิภาพและลดภาระในการทดสอบ โดย

อาศัยการจัดลำดับความสำคัญกรณีทดสอบ ทำให้ผู้ทดสอบสามารถเลือกทดสอบกรณีทดสอบตามลำดับความสำคัญได้ โดยไม่ต้องทดสอบกรณีทดสอบทั้งหมด

ขั้นตอนการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เริ่มจากการสร้างกรณีทดสอบ โดยส่วนสร้างกรณีทดสอบ จากนั้นกรณีทดสอบที่ได้จะถูกนำไปสู่ส่วนทดสอบและตรวจสอบผลลัพธ์ ในกรณีที่มีส่วนจัดลำดับความสำคัญกรณีทดสอบ กรณีทดสอบที่สร้างเสร็จแล้วจะถูกนำมาจัดลำดับความสำคัญก่อน แล้วจึงคัดเลือกเพียงบางส่วนไปทำการทดสอบต่อไป ดังแสดงในรูป 3.1



รูปที่ 3.1 ขั้นตอนการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้การจัดลำดับความสำคัญของกรณีทดสอบ

3.2 แนวคิดการจัดลำดับความสำคัญกรณีทดสอบ

3.2.1 จุดประสงค์การจัดลำดับความสำคัญกรณีทดสอบ

ดังที่ได้กล่าวไปแล้วในบทที่ 2 หัวข้อ 2.4 การจัดลำดับความสำคัญกรณีทดสอบอาจมีเป้าหมายหรือจุดประสงค์แตกต่างกันไปตามความต้องการในการใช้งานของผู้ทดสอบ ซึ่งจุดประสงค์ที่แตกต่างกันย่อมส่งผลให้วิธีจัดลำดับความสำคัญแตกต่างกันไปด้วย ดังนั้นก่อนที่จะพิจารณาถึงวิธีการจัดลำดับความสำคัญของกรณีทดสอบ จึงจำเป็นต้องกำหนดจุดประสงค์ที่ชัดเจนเสียก่อน

กรณีนี้เราต้องการวิธีการจัดลำดับความสำคัญกรณีทดสอบที่สามารถนำไปใช้ได้โดยมีนัยทั่วไป และสามารถใช้ได้กับการทดสอบหลายรูปแบบ ดังนั้นการจัดลำดับความสำคัญที่มีรายละเอียดขึ้นกับลักษณะเฉพาะของงาน เช่น การจัดลำดับความสำคัญตามระดับความรุนแรงของข้อผิดพลาด ซึ่งต้องการการแบ่งประเภทข้อผิดพลาดที่ชัดเจน หรือการจัดลำดับความสำคัญตาม

ส่วนที่แก้ไขล่าสุด ซึ่งใช้เฉพาะกับการทดสอบแบบรีเกรสชัน เป็นต้น จึงไม่เหมาะสมสำหรับงานวิจัยนี้ จุดประสงค์ในการจัดลำดับความสำคัญกรณีทดสอบ จึงมุ่งไปที่การจัดลำดับความสำคัญตามประสิทธิภาพในการค้นพบข้อผิดพลาด หรือก็คือ มุ่งจัดลำดับความสำคัญให้ชุดทดสอบสามารถค้นพบข้อผิดพลาดได้อย่างรวดเร็ว

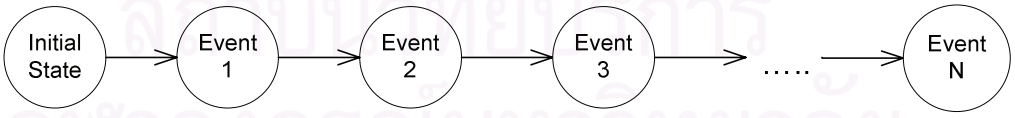
3.2.2 ปัจจัยในการจัดลำดับความสำคัญกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

เนื่องจากเราต้องการการจัดลำดับความสำคัญกรณีทดสอบที่สามารถนำไปใช้ได้กับการทดสอบรูปแบบต่างๆ แนวทางการจัดลำดับความสำคัญจึงพิจารณาถึงปัจจัยพื้นฐานของการทดสอบได้แก่ ความครอบคลุม เป็นปัจจัยหลักในการจัดลำดับความสำคัญ

ส่วนที่เราให้ความสนใจในการจัดลำดับความสำคัญของกรณีทดสอบตามเกณฑ์ความครอบคลุม คือ ข้อมูลนำเข้า เนื่องจากข้อมูลนำเข้าเป็นสิ่งที่กำหนดทิศทางและเป้าหมายการทำงานของกรณีทดสอบนั้นๆ ส่วนผลลัพธ์ที่คาดการณ์ไว้นั้นอาจไม่มีการกำหนดไว้อย่างชัดเจน ในกรณีที่เป็นการทดสอบโดยมนุษย์ และที่สำคัญคือ ข้อมูลนำเข้าที่แตกต่างกันอาจนำไปสู่ผลลัพธ์แบบเดียวกันได้ แต่ข้อมูลนำเข้าที่เหมือนกันต้องนำไปสู่ผลลัพธ์เดียวกันเสมอ หรือกล่าวได้ว่าข้อมูลนำเข้ามีความสัมพันธ์แบบ กลุ่ม ต่อ 1 (many to one) กับผลลัพธ์ที่คาดการณ์ไว้

เนื่องจากส่วนต่อประสานกราฟิกกับผู้ใช้มีลักษณะการทำงานที่แตกต่างจากซอฟต์แวร์ตามปกติ โดยมีการทำงานตามเหตุการณ์ที่เข้ามา (Event driven) ดังนั้นในกรณีที่เราสนใจเฉพาะข้อมูลนำเข้า เราสามารถเขียนนิยามของกรณีทดสอบได้ว่า

นิยาม กรณีทดสอบประกอบด้วยสถานะเริ่มต้น (หรือเงื่อนไขเบื้องต้น) กับลำดับของเหตุการณ์ (หรือการกระทำ)



รูปที่ 3.2 โครงสร้างของกรณีทดสอบ

กรณีทดสอบที่อ้างถึงจะมีลักษณะดังรูป 3.2 และสำหรับเหตุการณ์ที่เป็นองค์ประกอบของกรณีทดสอบ เราจะอ้างถึงในลักษณะของเหตุการณ์ระดับสูง เพื่อสื่อความหมายของเหตุการณ์อย่างชัดเจน

3.2.3 เกณฑ์ความครอบคลุมสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้

งานวิจัยนี้ได้กำหนดเกณฑ์ความครอบคลุมสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ โดยแบ่งออกเป็น 4 ระดับ ดังนี้

1) ความครอบคลุมระดับเหตุการณ์

เหตุการณ์ถือว่าเป็นหน่วยย่อยที่สุดที่ใช้แทนการกระทำของผู้ใช้ในการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ เช่น การกดปุ่มหรือเลือกเมนู เป็นต้น และเนื่องจากส่วนต่อประสานกราฟิกกับผู้ใช้ทำงานตอบสนองในระดับเหตุการณ์ หรืออาจกล่าวอีกแง่หนึ่งว่า ลำดับเหตุการณ์ที่ได้รับเป็นสิ่งกำหนดทิศทางการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ นอกจากนี้ยังเป็นองค์ประกอบพื้นฐานของกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ ดังนั้นเกณฑ์ความครอบคลุมระดับเหตุการณ์จึงถือเป็นเกณฑ์ความครอบคลุมระดับพื้นฐานที่สุด ซึ่งการใช้เหตุการณ์มาเป็นเกณฑ์ความครอบคลุมนี้ได้มีการนำเสนอไว้ในงานวิจัยอื่น [11] เช่นกัน แต่กรณีดังกล่าวเป็นการประเมินความครอบคลุมทั้งหมดของส่วนต่อประสานกราฟิกกับผู้ใช้ ไม่ได้มุ่งเน้นที่การนำมาใช้จัดลำดับความสำคัญกรณีทดสอบ ในงานวิจัยนี้ เราได้นิยามเกณฑ์ความครอบคลุมระดับเหตุการณ์ไว้ว่า

นิยาม “ความครอบคลุมระดับเหตุการณ์ หมายถึง ความครอบคลุมถึงเหตุการณ์ใดๆที่สามารถเกิดขึ้นได้ในส่วนต่อประสานกราฟิกกับผู้ใช้”

2) ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์

นอกจากเหตุการณ์แล้ว การทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ยังขึ้นอยู่กับข้อมูลสำคัญอีกชนิดหนึ่งนั่นคือ ลำดับเหตุการณ์ กรณีทดสอบที่มีสถานะเริ่มต้นและเหตุการณ์เหมือนๆกันอาจทำหน้าที่ในการทดสอบแตกต่างกันได้ ถ้าลำดับของเหตุการณ์เหล่านั้นแตกต่างกัน แต่กรณีทดสอบที่มีสถานะเริ่มต้นและลำดับเหตุการณ์เหมือนกันจะทำหน้าที่เดียวกันเสมอ (ซึ่งก็คือเป็นกรณีทดสอบเดียวกันนั่นเอง)

ที่เป็นเช่นนี้เนื่องจากลำดับเหตุการณ์ส่งผลกระทบถึงการแปลความหมายของเหตุการณ์นั้นๆโดยโปรแกรม ดังนั้นเหตุการณ์ที่เกิดขึ้นต่อเนื่องกันไปมีโอกาสที่จะส่งผลกระทบต่อการทำงานของเหตุการณ์ถัดๆไป ดังนั้น เราจึงกำหนดเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ หรือระดับคู่ของเหตุการณ์ขึ้น เพื่อให้มั่นใจว่าเหตุการณ์ใดเหตุการณ์หนึ่งจะไม่ส่งผลกระทบและทำให้เหตุการณ์อื่นมีการทำงานที่ผิดปกติ

ตามความเป็นจริงแล้วเหตุการณ์ใดเหตุการณ์หนึ่งสามารถส่งผลกระทบถึงเหตุการณ์อื่นๆที่ตามมาได้ โดยไม่จำเป็นต้องเป็นเหตุการณ์ที่เกิดขึ้นติดต่อกัน แต่เพื่อลดความซับซ้อนในการคำนวณ และเนื่องจากเหตุการณ์ที่อยู่ห่างกันมากโอกาสที่จะส่งผลกระทบต่อกันเป็นไปได้น้อย

นอกจากนี้ความสัมพันธ์ระหว่างเหตุการณ์ที่เกิดขึ้นติดต่อกัน สามารถบ่งชี้ถึงระดับความแตกต่างของกรณีทดสอบส่วนใหญ่ได้ กล่าวคือ กรณีทดสอบที่มีหน้าที่การทำงานใกล้เคียงกันมักประกอบด้วยความสัมพันธ์ระหว่างเหตุการณ์ที่คล้ายคลึงกัน แต่สำหรับกรณีทดสอบที่ทำงานแตกต่างกันมักประกอบด้วยความสัมพันธ์ระหว่างเหตุการณ์ที่แตกต่างกัน โดยเฉพาะวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มที่พิจารณาชุดทดสอบทั้งหมด หากพิจารณาระดับเหตุการณ์มักพบว่ากรณีทดสอบที่ทดสอบส่วนเดียวกันประกอบด้วยเหตุการณ์ใกล้เคียงกัน ทำให้เมื่อพิจารณาความครอบคลุมแล้ว กรณีทดสอบกลุ่มเล็กๆกลุ่มเดียวสามารถครอบคลุมเหตุการณ์ทั้งหมดได้ แต่หากพิจารณาความสัมพันธ์ระหว่างเหตุการณ์จะพบว่าแตกต่างกันไปตามขั้นตอนและลำดับการทำงานของกรณีทดสอบแต่ละกรณี ทำให้กรณีทดสอบที่มีลักษณะการทำงานแตกต่างกันได้รับการทดสอบมากขึ้น สำหรับความสัมพันธ์ระหว่างเหตุการณ์ที่เกิดขึ้นติดต่อกันในกรณีทดสอบแสดงในรูป 3.3

รูปที่ 3.3 ความสัมพันธ์ระหว่างเหตุการณ์ในกรณีทดสอบ

จากเหตุผลที่กล่าวมา งานวิจัยนี้จึงมุ่งสนใจเฉพาะความครอบคลุมระดับเหตุการณ์ที่เกิดติดต่อกัน ในลักษณะของคู่ของเหตุการณ์เท่านั้น โดยนิยามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ว่า

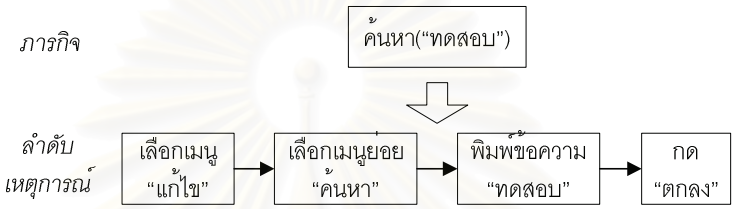
นิยาม "ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ หมายถึง ความครอบคลุมถึงสองเหตุการณ์ใดๆที่สามารถเกิดขึ้นติดต่อกันได้ในส่วนต่อประสานกราฟิกกับผู้ใช้"

3) ความครอบคลุมระดับภารกิจ

ภารกิจหรือหน้าที่การทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้เป็นองค์ประกอบในระดับที่สูงกว่าระดับเหตุการณ์ หมายถึง การทำงานร่วมกันของเหตุการณ์ เพื่อให้บรรลุเป้าหมาย เช่น การเปิดแฟ้มข้อมูล ซึ่งอาจประกอบไปด้วยเหตุการณ์ย่อยๆ ตั้งแต่การเลือกเมนู การเลือกแฟ้มข้อมูล ไปจนกระทั่งการกดปุ่ม "ตกลง" เป็นต้น เรานิยามภารกิจว่า

นิยาม “ภารกิจ (Task) หมายถึง ลำดับของเหตุการณ์ที่เกิดขึ้นในส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อบรรลุจุดมุ่งหมายอย่างใดอย่างหนึ่ง”

ตัวอย่างภารกิจที่แสดงในรูป 3.4 คือภารกิจค้นหาข้อความ ซึ่งประกอบไปด้วยเหตุการณ์ เลือกเมนู “แก้ไข” เลือกเมนูย่อย “ค้นหา” พิมพ์ข้อความที่ต้องการค้น ในกรณีนี้คือข้อความ “ทดสอบ” แล้วจึงกดปุ่ม “ตกลง” ซึ่งเหตุการณ์เหล่านี้จะทำงานต่อเนื่องกันไป



รูปที่ 3.4 ตัวอย่างภารกิจค้นหาข้อความ

สาเหตุที่ต้องรวมเหตุการณ์เข้าด้วยกันเป็นภารกิจ นั่นเพราะเหตุการณ์เหตุการณ์เดียวอาจสื่อความหมายได้ไม่ครบถ้วนสมบูรณ์ เช่น เหตุการณ์กดปุ่ม “ตกลง” เป็นต้น ซึ่งทำให้มองไม่เห็นภาพรวมของสิ่งที่กำลังทำ การรวมเหตุการณ์เข้าเป็นภารกิจนอกจากจะช่วยให้ผู้ทดสอบมองเห็นภาพของสิ่งที่กรณีทดสอบต้องการทดสอบมากขึ้นแล้ว ภารกิจยังสามารถช่วยแบ่งส่วนการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ออกเป็นส่วนๆ ทำให้สามารถกำหนดส่วนที่สำคัญ ซึ่งถูกใช้งานบ่อยอีกด้วย

จากมุมมองของการจัดลำดับความสำคัญกรณีทดสอบ ภารกิจจะช่วยทำให้ความสำคัญกระจายลงไปสู่ทุกงานอย่างเท่าเทียมกัน กรณีทดสอบจะถูกจัดลำดับความสำคัญโดยพิจารณาจากภารกิจหรือสิ่งที่ทำเป็นหลัก โดยไม่คำนึงถึงว่าภารกิจนั้นๆประกอบไปด้วยเหตุการณ์ย่อยๆมากน้อยเพียงใด ทำให้ทุกๆภารกิจมีโอกาสที่จะได้รับการทดสอบอย่างเท่าเทียมกัน

สำหรับงานวิจัยนี้ภารกิจจะถูกระบุโดยผู้ทำการทดสอบ โดยยึดตามเอกสารการออกแบบซอฟต์แวร์เป็นหลัก นอกจากนี้ยังได้เพิ่มเกณฑ์ในการกำหนดภารกิจ คือ ลำดับเหตุการณ์หนึ่งๆจะถูกกำหนดเป็นภารกิจได้เพียงภารกิจเดียว และไม่มีลำดับเหตุการณ์ 2 ลำดับใดๆที่ถูกกำหนดเป็นภารกิจเดียวกัน ทั้งนี้เพื่อป้องกันการทับซ้อนกันของภารกิจ และป้องกันการที่กรณีทดสอบ 2 กรณีกลายเป็นกรณีทดสอบเดียวกัน หลังการเปลี่ยนมุมมองจากระดับเหตุการณ์เป็นระดับภารกิจ

แนวทางการกำหนดภารกิจนั้นจำเป็นอย่างยิ่งที่ภารกิจทั้งหมดควรมีระดับความสำคัญใกล้เคียงกัน เพราะในการคำนวณตามเกณฑ์ความครอบคลุมจะถือว่าภารกิจทุกภารกิจ

ถือเป็นหนึ่งหน่วยเท่ากัน ดังนั้นหากภารกิจใดถูกแบ่งในระดับที่ละเอียดลงไป จะทำให้มีโอกาสถูกทดสอบมากกว่าภารกิจอื่นๆ เราให้นิยามเกณฑ์ความครอบคลุมระดับภารกิจว่า

นิยาม "ความครอบคลุมระดับภารกิจ หมายถึง ความครอบคลุมถึงภารกิจใดๆที่สามารถเกิดขึ้นได้ในส่วนต่อประสานกราฟิกกับผู้ใช้"

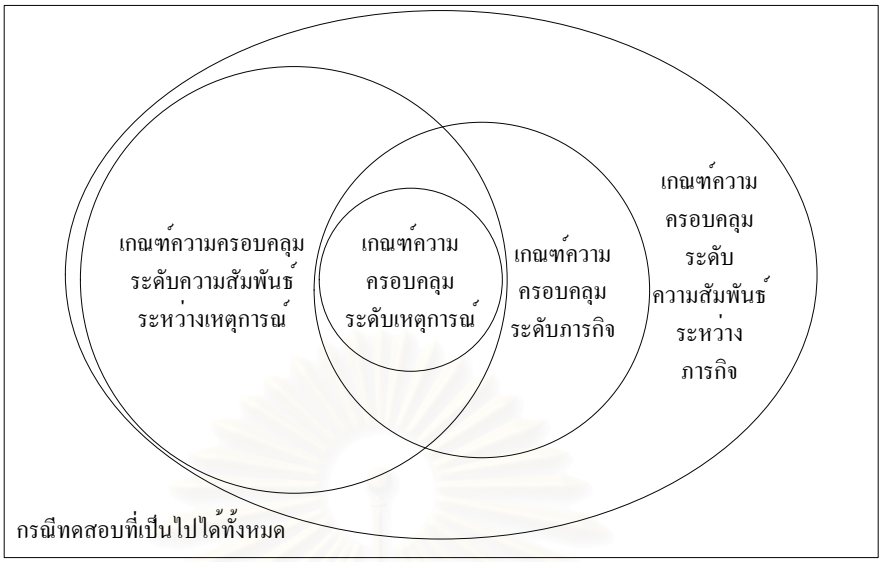
4) ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจ

กรณีทดสอบหนึ่งๆอาจประกอบด้วยภารกิจมากกว่าหนึ่งภารกิจต่อเนื่องกันไปเพื่อให้บรรลุเป้าหมายในการทำงาน โดยแต่ละภารกิจอาจก่อให้เกิดผลกระทบต่อภารกิจอื่นๆที่เกิดตามมาในภายหลัง ทำให้เป็นไปได้ที่ภารกิจหนึ่งๆจะสามารถส่งต่อผลกระทบที่ทำให้เกิดข้อผิดพลาดขึ้นในการทำงานของภารกิจอื่นๆ ดังนั้นจึงเป็นที่มาของแนวคิดเกี่ยวกับความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจ

เช่นเดียวกับความสัมพันธ์ระหว่างเหตุการณ์ ความสัมพันธ์ระหว่างภารกิจในงานวิจัยนี้ เราพิจารณาเฉพาะคู่ของภารกิจที่เกิดขึ้นติดต่อกันเท่านั้น โดยเรานิยามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจไว้ดังนี้

นิยาม "ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจ หมายถึง ความครอบคลุมถึงสองภารกิจใดๆที่สามารถเกิดขึ้นติดต่อกันได้ในส่วนต่อประสานกราฟิกกับผู้ใช้"

ความสัมพันธ์ระหว่างเกณฑ์ความครอบคลุมทั้ง 4 ระดับที่กล่าวมาข้างต้นแบ่งออกเป็น เกณฑ์ความครอบคลุมระดับเหตุการณ์จะถูกครอบคลุม (subsume) โดยเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ และเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจจะครอบคลุมเกณฑ์ความครอบคลุมระดับภารกิจ ทั้งนี้เพราะเมื่อมีเหตุการณ์หรือภารกิจใหม่ๆย่อมทำให้เกิดความสัมพันธ์ใหม่ตามมา ส่วนเกณฑ์ความครอบคลุมระดับภารกิจจะครอบคลุมเกณฑ์ความครอบคลุมระดับเหตุการณ์ เนื่องจากข้อกำหนดที่ว่าจะไม่มี 2 ลำดับเหตุการณ์ใดๆถูกกำหนดเป็นภารกิจเดียวกัน ทำให้เมื่อมีเหตุการณ์ใหม่จะเกิดภารกิจใหม่ขึ้นด้วย และเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจจะครอบคลุมเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ เพราะเมื่อมีความสัมพันธ์ระหว่างเหตุการณ์ใหม่เกิดขึ้น จะทำให้เกิดภารกิจใหม่หรือความสัมพันธ์ระหว่างภารกิจใหม่ขึ้นเสมอ สำหรับความสัมพันธ์ระหว่างเกณฑ์ความครอบคลุมระดับต่างๆแสดงในรูป 3.5



รูปที่ 3.5 แผนภาพเวนน์แสดงระดับความครอบคลุมของเกณฑ์ความครอบคลุมรูปแบบต่างๆ

3.3 ประเภทการจัดลำดับความสำคัญกรณีทดสอบ

การพิจารณาเกณฑ์ความครอบคลุมสำหรับกรณีทดสอบจะต่างจากการพิจารณาเกณฑ์ความครอบคลุมตามปกติ ปกติในการหาความครอบคลุมจะใช้ชิ้นงานเป็นที่ตั้ง แล้วพิจารณาความครอบคลุมที่เป็นไปได้ทั้งหมด แต่ในกรณีของการจัดลำดับความสำคัญของกรณีทดสอบจะพิจารณาเกณฑ์ความครอบคลุมเฉพาะที่อยู่ในชุดทดสอบเท่านั้น ส่วนที่นอกเหนือจากนี้ถือว่าอยู่นอกขอบเขตการทดสอบ

นอกจากพิจารณาจากเกณฑ์ความครอบคลุม ยังมีเทคนิคในการจัดลำดับความสำคัญของกรณีทดสอบที่เราต้องคำนึงถึง คือ วิธีการคำนวณผลย้อนกลับ (feedback) ของการจัดลำดับความสำคัญก่อนหน้า หมายถึง การนำผลการจัดลำดับความสำคัญกรณีทดสอบส่วนที่ได้จัดไปแล้วมาเป็นเครื่องชี้วัดความเหมาะสมสำหรับการจัดลำดับความสำคัญกรณีทดสอบถัดไป ซึ่งแบ่งวิธีการออกเป็น 2 ประเภท ดังที่ได้อธิบายในหัวข้อ 2.4.2 ในบทที่ 2

จากเกณฑ์ความครอบคลุมทั้ง 4 ประเภท เมื่อพิจารณาร่วมกับการคำนวณในการจัดลำดับความสำคัญกรณีทดสอบทั้ง 2 แบบ เราจะได้ประเภทการจัดลำดับความสำคัญของกรณีทดสอบทั้งหมด 8 ประเภท ดังแสดงในตาราง 3.1 โดยจะพิจารณากรณีทดสอบในลักษณะของลำดับของเหตุการณ์หรือภารกิจก่อน ในส่วนที่เกี่ยวกับสถานะเริ่มต้นหรือเงื่อนไขเบื้องต้นของกรณีทดสอบจะกล่าวถึงในหัวข้อถัดไป ซึ่งประเภทการจัดลำดับความสำคัญกรณีทดสอบรูปแบบต่างๆสามารถอธิบายได้ดังนี้

ตารางที่ 3.1 ประเภทการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ทั้งหมด

วิธีการคำนวณ	ระดับการทำงาน	ประเภทการจัดลำดับความสำคัญ
เบ็ดเสร็จ	เหตุการณ์	การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ
		การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเบ็ดเสร็จ
	ภารกิจ	การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเบ็ดเสร็จ
		การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเบ็ดเสร็จ
เพิ่ม	เหตุการณ์	การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม
		การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม
	ภารกิจ	การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม
		การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม

3.3.1 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ (Total Event Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนเหตุการณ์ที่อยู่ในกรณีทดสอบจากมากไปน้อย แสดงตัวอย่างในตารางที่ 3.2 ซึ่งพบว่ากรณีทดสอบที่ประกอบด้วยเหตุการณ์จำนวนมากจะมีระดับความสำคัญสูงกว่ากรณีทดสอบที่ประกอบด้วยเหตุการณ์จำนวนน้อยกว่า

3.3.2 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเบ็ดเสร็จ (Total Event Pair Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนความสัมพันธ์หรือคู่ของเหตุการณ์ที่มีอยู่ในกรณีทดสอบจากมากไปน้อย แสดงตัวอย่างในตารางที่ 3.3 ซึ่งพบว่ากรณีทดสอบที่ประกอบด้วยความสัมพันธ์ระหว่างเหตุการณ์จำนวนมากจะมีระดับความสำคัญสูงกว่ากรณีทดสอบที่ประกอบด้วยความสัมพันธ์ระหว่างเหตุการณ์จำนวนน้อยกว่า

ตารางที่ 3.2 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ

		ลำดับเหตุการณ์
กรณีทดสอบ	1	A – B – C – E – F
	2	A – B – D
	3	A – B – D – E
	4	A – C
ผลการจัดลำดับความสำคัญ	1	A – B – C – E – F
	3	A – B – D – E
	2	A – B – D
	4	A – C

ตารางที่ 3.3 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเบ็ดเสร็จ

		ลำดับเหตุการณ์	ความสัมพันธ์ระหว่างเหตุการณ์
กรณีทดสอบ	1	A – B – C – E – F	(A,B), (B,C), (C,E), (E,F)
	2	A – B – D	(A,B), (B,D)
	3	A – B – D – E	(A,B), (B,D), (D,E)
	4	A – C	(A,C)
ผลการจัดลำดับความสำคัญ	1	A – B – C – E – F	(A,B), (B,C), (C,E), (E,F)
	3	A – B – D – E	(A,B), (B,D), (D,E)
	2	A – B – D	(A,B), (B,D)
	4	A – C	(A,C)

3.3.3 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเบ็ดเสร็จ (Total Task Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนภารกิจในกรณีทดสอบ โดยเรียงจากมากไปน้อย แสดงตัวอย่างในตาราง 3.4 ซึ่งพบว่ากรณีทดสอบที่ประกอบด้วยภารกิจจำนวนมากกว่าจะมีระดับความสำคัญสูงกว่ากรณีทดสอบที่ประกอบด้วยภารกิจจำนวนน้อยกว่า

ตารางที่ 3.4 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเบ็ดเสร็จ

		ลำดับภารกิจ
กรณีทดสอบ	1	T1 – T2 – T3 – T5 – T6
	2	T1 – T2 – T4
	3	T1– T2 – T4 – T5
	4	T1 – T3
ผลการจัดลำดับความสำคัญ	1	T1 – T2 – T3 – T5 – T6
	3	T1– T2 – T4 – T5
	2	T1 – T2 – T4
	4	T1 – T3

3.3.4 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเบ็ดเสร็จ (Total Task Pair Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนความสัมพันธ์หรือคู่ของภารกิจในกรณีทดสอบจากมากไปน้อย แสดงตัวอย่างในตาราง 3.5 ซึ่งพบว่ากรณีทดสอบที่ประกอบด้วยความสัมพันธ์ระหว่างภารกิจจำนวนมากจะมีระดับความสำคัญสูงกว่ากรณีทดสอบที่ประกอบด้วยความสัมพันธ์ระหว่างภารกิจจำนวนน้อยกว่า

ตารางที่ 3.5 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเบ็ดเสร็จ

		ลำดับภารกิจ	ความสัมพันธ์ระหว่างภารกิจ
กรณีทดสอบ	1	T1 – T2 – T3 – T5 – T6	(T1,T2), (T2,T3), (T3,T5), (T5,T6)
	2	T1 – T2 – T4	(T1,T2), (T2,T4)
	3	T1 – T2 – T4 – T5	(T1,T2), (T2,T4), (T4,T5)
	4	T1 – T3	(T1,T3)
ผลการจัดลำดับความสำคัญ	1	T1 – T2 – T3 – T5 – T6	(T1,T2), (T2,T3), (T3,T5), (T5,T6)
	3	T1 – T2 – T4 – T5	(T1,T2), (T2,T4), (T4,T5)
	2	T1 – T2 – T4	(T1,T2), (T2,T4)
	4	T1 – T3	(T1,T3)

3.3.5 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม (Additional Event Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนเหตุการณ์ที่อยู่ในกรณีทดสอบ โดยพิจารณาเฉพาะเหตุการณ์ที่ไม่ซ้ำกับเหตุการณ์ที่ถูกครอบคลุมโดยกลุ่มกรณีทดสอบที่ถูกเลือกไปก่อนหน้า แสดงตัวอย่างในตาราง 3.6 ซึ่งพบว่าเหตุการณ์ในกรณีทดสอบที่แตกต่างไปจากเหตุการณ์ในกรณีทดสอบที่ผ่านการจัดลำดับความสำคัญก่อนหน้าจะเป็นปัจจัยที่ใช้ประเมินระดับความสำคัญของกรณีทดสอบนั้นๆ

ตารางที่ 3.6 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม

		ลำดับเหตุการณ์
กรณีทดสอบ	1	A – B – C – D – E – F
	2	A – B – C – D – E – G
	3	A – B – H – I – J
	4	A – B – H – I – D – E
ผลการจัดลำดับความสำคัญ	1	A – B – C – D – E – F
	3	A – B – H – I – J
	2	A – B – C – D – E – G
	4	A – B – H – I – D – E

หมายเหตุ เหตุการณ์ที่ขีดเส้นใต้ คือ เหตุการณ์แตกต่างจากเหตุการณ์ในกรณีทดสอบที่ถูกจัดลำดับความสำคัญก่อนหน้า

3.3.6 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม (Additional Event Pair Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนความสัมพันธ์หรือคู่ของเหตุการณ์ที่อยู่ในกรณีทดสอบจากมากไปน้อย โดยพิจารณาเฉพาะคู่ของเหตุการณ์ที่ไม่ซ้ำกับความสัมพันธ์ของเหตุการณ์ที่ถูกครอบคลุมโดยกลุ่มกรณีทดสอบที่ถูกเลือกไปก่อนหน้า แสดงตัวอย่างในตาราง 3.7 ซึ่งพบว่าความสัมพันธ์ระหว่างเหตุการณ์ในกรณีทดสอบที่แตกต่างไปจากความสัมพันธ์ระหว่างเหตุการณ์ในกรณีทดสอบที่ผ่านการจัดลำดับความสำคัญก่อนหน้าจะเป็นปัจจัยที่ใช้ประเมินระดับความสำคัญของกรณีทดสอบนั้นๆ

ตารางที่ 3.7 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม

		ลำดับเหตุการณ์	ความสัมพันธ์ระหว่างเหตุการณ์
กรณีทดสอบ	1	A – B – C – D	(A,B), (B,C), (C,D)
	2	A – B – E	(A,B), (B,E)
	3	A – B – E – D	(A,B), (B,E), (E,D)
	4	A – B – F – G	(A,B), (B,F), (F,G)
ผลการจัดลำดับความสำคัญ	1	A – B – C – D	(A,B), (B,C), (C,D)
	4	A – B – F – G	(A,B), (B,F), (F,G)
	3	A – B – E – D	(A,B), (B,E), (E,D)
	2	A – B – E	(A,B), (B,E)

หมายเหตุ ความสัมพันธ์ระหว่างเหตุการณ์ที่ขีดเส้นใต้ คือ ความสัมพันธ์ระหว่างเหตุการณ์ที่แตกต่างจากความสัมพันธ์ระหว่างเหตุการณ์ในกรณีทดสอบที่ถูกจัดลำดับความสำคัญก่อนหน้า

3.3.7 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม (Additional Task Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนภารกิจที่อยู่ในกรณีทดสอบจากมากไปน้อย โดยพิจารณาเฉพาะภารกิจที่ไม่ซ้ำกับภารกิจที่ถูกครอบคลุมโดยกลุ่มกรณีทดสอบที่ถูกเลือกไปก่อนหน้า แสดงตัวอย่างดังตาราง 3.8 ซึ่งพบว่าภารกิจในกรณีทดสอบที่แตกต่างไปจากภารกิจในกรณีทดสอบที่ผ่านการจัดลำดับความสำคัญก่อนหน้าจะเป็นปัจจัยที่ใช้ประเมินระดับความสำคัญของกรณีทดสอบนั้นๆ

3.3.8 การจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม (Additional Task Pair Coverage)

เป็นการจัดลำดับความสำคัญกรณีทดสอบตามจำนวนความสัมพันธ์หรือคู่ของภารกิจที่อยู่ในกรณีทดสอบจากมากไปน้อย โดยพิจารณาเฉพาะคู่ของภารกิจที่ไม่ซ้ำกับความสัมพันธ์ของภารกิจที่ถูกครอบคลุมโดยกลุ่มกรณีทดสอบที่ถูกเลือกไปก่อนหน้า แสดงตัวอย่างดังตาราง 3.9 ซึ่งพบว่าความสัมพันธ์ระหว่างภารกิจในกรณีทดสอบที่แตกต่างไปจากความสัมพันธ์ระหว่างภารกิจในกรณีทดสอบที่ผ่านการจัดลำดับความสำคัญก่อนหน้าจะเป็นปัจจัยที่ใช้ประเมินระดับความสำคัญของกรณีทดสอบนั้นๆ

ตารางที่ 3.8 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม

		ลำดับภารกิจ
กรณีทดสอบ	1	T1 – T2 – T3 – T4 – T5 – T6
	2	T1 – T2 – T3 – T4 – T5 – T7
	3	T1 – T2 – T8 – T9 – T10
	4	T1 – T2 – T8 – T9 – T4 – T5
ผลการจัดลำดับความสำคัญ	1	T1 – T2 – T3 – T4 – T5 – T6
	3	T1 – T2 – T8 – T9 – T10
	2	T1 – T2 – T3 – T4 – T5 – T7
	4	T1 – T2 – T8 – T9 – T4 – T5

หมายเหตุ ภารกิจที่ขีดเส้นใต้ คือ ภารกิจที่แตกต่างจากภารกิจในกรณีทดสอบที่ถูกจัดลำดับความสำคัญก่อนหน้า

ตารางที่ 3.9 ตัวอย่างการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม

		ลำดับภารกิจ	ความสัมพันธ์ระหว่างภารกิจ
กรณีทดสอบ	1	T1 – T2 – T3 – T4	(T1,T2), (T2,T3), (T3,T4)
	2	T1 – T2 – T5	(T1,T2), (T2,T5)
	3	T1 – T2 – T5 – T4	(T1,T2), (T2,T5), (T5,T4)
	4	T1 – T2 – T6 – T7	(T1,T2), (T2,T6), (T6,T7)
ผลการจัดลำดับความสำคัญ	1	T1 – T2 – T3 – T4	(T1,T2), (T2,T3), (T3,T4)
	4	T1 – T2 – T6 – T7	(T1,T2), (T2,T6), (T6,T7)
	3	T1 – T2 – T5 – T4	(T1,T2), (T2,T5), (T5,T4)
	2	T1 – T2 – T5	(T1,T2), (T2,T5)

หมายเหตุ ความสัมพันธ์ระหว่างภารกิจที่ขีดเส้นใต้ คือ ความสัมพันธ์ระหว่างภารกิจที่แตกต่างจากความสัมพันธ์ระหว่างภารกิจในกรณีทดสอบที่ถูกจัดลำดับความสำคัญก่อนหน้า

จากตัวอย่างแม้ว่าการจัดลำดับความสำคัญกรณีทดสอบบางประเภทจะให้ผลลัพธ์เหมือนกัน แต่มีกรณีบางกรณี เช่น กรณีที่มีเหตุการณ์หรือภารกิจซ้ำกันภายในกรณีทดสอบเดียวกัน เป็นต้น ซึ่งกรณีเหล่านี้จะทำให้ผลลัพธ์แตกต่างกัน ดังนั้นจึงสรุปว่ามีวิธีการจัดลำดับความสำคัญกรณีทดสอบที่แตกต่างกันทั้งหมด 8 ประเภท

3.4 ปัญหาที่พบและแนวทางการแก้ไข

3.4.1 ปัญหาสถานะเริ่มต้น

จากนิยามของกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ที่เรากำหนดไว้ประกอบด้วย 2 ส่วน ได้แก่ สถานะเริ่มต้นหรือเงื่อนไขเริ่มต้น และส่วนลำดับเหตุการณ์ ซึ่งการจัดลำดับความสำคัญกรณีทดสอบที่ผ่านมา เราพิจารณาเฉพาะลำดับเหตุการณ์เป็นหลัก โดยไม่ได้กล่าวถึงสถานะเริ่มต้น

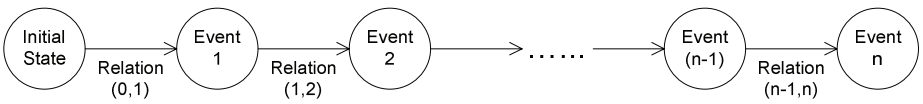
สถานะเริ่มต้นที่แตกต่างกันสามารถส่งผลต่อการทดสอบ ทั้งในด้านเป้าหมาย วิธีการดำเนินการ รวมไปถึงผลลัพธ์จากการทดสอบด้วย เราจึงต้องพิจารณาว่า การจัดลำดับลำดับความสำคัญกรณีทดสอบที่กำหนดขึ้นควรจัดการอย่างไรกับสถานะเริ่มต้น ในขั้นตอนนี้เราจำแนกประเภทการจัดลำดับความสำคัญกรณีทดสอบทั้ง 8 แบบ ออกเป็น 2 ประเภทหลักๆ ได้แก่

1) การจัดลำดับความสำคัญกรณีทดสอบที่ใช้เกณฑ์ความครอบคลุมเชิงเดี่ยว

ได้แก่ การจัดลำดับความสำคัญแบบเหตุการณ์และภารกิจทั้งแบบเบ็ดเสร็จและแบบเพิ่ม เนื่องจากวิธีการเหล่านี้เน้นการพิจารณาเพื่อค้นหาข้อผิดพลาดที่เกิดขึ้นจากเหตุการณ์หรือภารกิจนั้นๆ โดยเฉพาะ โดยไม่คำนึงถึงความสัมพันธ์ระหว่างเหตุการณ์หรือภารกิจ ดังนั้นสถานะเริ่มต้น ซึ่งโดยลำพังแล้วถือว่าไม่ได้อยู่ในส่วนที่ต้องการทดสอบ ถือว่าไม่เกี่ยวข้องกับการจัดลำดับความสำคัญในลักษณะนี้ สรุปได้ว่า การจัดลำดับความสำคัญกรณีทดสอบประเภทนี้จะไม่ให้ความสำคัญกับสถานะเริ่มต้น

2) การจัดลำดับความสำคัญกรณีทดสอบที่ใช้เกณฑ์ความครอบคลุมเชิงความสัมพันธ์

ได้แก่ การจัดลำดับความสำคัญแบบความสัมพันธ์ระหว่างเหตุการณ์และความสัมพันธ์ระหว่างภารกิจทั้งแบบเบ็ดเสร็จและแบบเพิ่ม เนื่องจากวิธีการเหล่านี้ออกแบบมาเพื่อค้นหาข้อผิดพลาดที่เกิดขึ้นจากผลกระทบระหว่างเหตุการณ์หรือภารกิจที่จะส่งผลต่อเหตุการณ์หรือภารกิจอื่นๆ ดังนั้นสถานะเริ่มต้นอาจส่งผลกระทบต่อการทำงานของเหตุการณ์หรือภารกิจ ถือว่ามีส่วนที่อาจก่อให้เกิดความผิดพลาดในการทำงานในลักษณะผลกระทบจากความสัมพันธ์ได้ สรุปว่าการจัดลำดับความสำคัญกรณีทดสอบประเภทนี้จะพิจารณาสถานะเริ่มต้นในลักษณะความสัมพันธ์ระหว่างสถานะเริ่มต้นกับเหตุการณ์หรือภารกิจแรกของกรณีทดสอบ ดังตัวอย่างในรูป 3.6



รูปที่ 3.6 ความสัมพันธ์ระหว่างเหตุการณ์โดยพิจารณาสถานะเริ่มต้น

3.4.2 ปัญหาความซ้ำซ้อนในกรณีทดสอบเดียวกัน

กรณีทดสอบหนึ่งๆอาจประกอบด้วยเหตุการณ์ ภารกิจ ความสัมพันธ์ระหว่างเหตุการณ์ หรือความสัมพันธ์ระหว่างภารกิจ เป็นจำนวนมาก ซึ่งมีโอกาสที่องค์ประกอบเหล่านี้จะเกิดขึ้นซ้ำๆภายในกรณีทดสอบเดียวกัน โดยอาจมีเป้าหมายในการทำงานเหมือนหรือแตกต่างกันไปตามตำแหน่งในลำดับการทำงานของกรณีทดสอบ หากพิจารณาตามเกณฑ์ความครอบคลุมอย่างเคร่งครัดแล้ว ไม่ว่าจะมีความซ้ำซ้อนมากเท่าไรในกรณีทดสอบจะถูกพิจารณาเพียงกรณีเดียวเท่านั้น ยกตัวอย่างเช่น กรณีทดสอบ T1 ทดสอบเหตุการณ์ A 1 ครั้ง กับกรณีทดสอบ T2 ทดสอบเหตุการณ์ A 10 ครั้ง ถ้าจัดลำดับความสำคัญแบบเบ็ดเสร็จ โดยพิจารณาจากเหตุการณ์ ถือว่ามีความสำคัญเท่ากัน ซึ่งอาจไม่เหมาะสมนักในกรณีที่มีกรณีทดสอบที่มีลำดับความสำคัญเท่ากันเป็นจำนวนมาก

เพื่อแก้ไขปัญหานี้ จึงมีการเพิ่มระดับความสำคัญเล็กน้อยลงไปในกรณีทดสอบที่มีเหตุการณ์ซ้ำ เพื่อให้กรณีที่มีระดับความสำคัญเท่ากัน กรณีทดสอบที่ทดสอบบ่อยครั้งกว่าจะมีลำดับความสำคัญสูงกว่ากรณีทดสอบที่ทดสอบครั้งเดียว ยกตัวอย่างเช่น ถ้ากรณีทดสอบมีเหตุการณ์ซ้ำให้เพิ่มระดับความสำคัญอีก 0.01 ในขณะที่ถ้าเหตุการณ์ปกติจะมีระดับความสำคัญเหตุการณ์ละ 1 ตัวอย่างแสดงในตารางที่ 3.10

ตารางที่ 3.10 ตัวอย่างการจัดลำดับความสำคัญกรณีทดสอบที่มีความซ้ำซ้อนโดยใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ

กรณีทดสอบ	ลำดับเหตุการณ์	ระดับความสำคัญ
1	A – B – C	3
2	A – A – D	2.01
3	A – E	2
4	A – A – A	1.02

แต่สำหรับการจัดลำดับความสำคัญแบบเพิ่มเราจะคิดว่าเป็นกรณีเหตุการณ์หรือภารกิจซ้ำ ก็ต่อเมื่อเป็นเหตุการณ์หรือภารกิจเป็นเหตุการณ์หรือภารกิจใหม่ที่ยังไม่เคยปรากฏมาก่อนในกรณีทดสอบที่จัดลำดับความสำคัญแล้วเท่านั้น

3.4.3 ปัญหา Set Covering

ปัญหา Set covering เป็นปัญหาที่พบในการเลือกซับเซตของเซตที่ประกอบด้วยสมาชิกที่เป็นเซต โดยให้ซับเซตที่ได้ประกอบด้วยสมาชิกของเซตที่เป็นสมาชิกครบถ้วนและมีจำนวนสมาชิกน้อยที่สุด ซึ่งนิยามได้ดังนี้

<u>นิยาม</u> ให้เซต S เป็นเซตของเซต เลือกเซต C ที่มีจำนวนสมาชิกน้อยที่สุด โดย $C \subseteq S$ และ

$$\bigcup_{i \in C} i = \bigcup_{j \in S} j$$

ปัญหา Set covering เป็นปัญหา NP-Hard (Non-deterministic Polynomial-time Hard Problem) ซึ่งมีความซับซ้อนและมีแนวทางในการแก้ปัญหาหลายวิธีการ

สำหรับการจัดลำดับความสำคัญกรณีทดสอบ เราจะพบปัญหาลักษณะนี้ในการจัดลำดับกรณีทดสอบแบบเพิ่ม ซึ่งเป็นการพิจารณาความครอบคลุมของชุดทดสอบทั้งหมด เช่นเดียวกับงานวิจัยลักษณะใกล้เคียงกันหลายชิ้น [10] [12] [13] ซึ่งปัญหาที่พบอธิบายได้ตามตาราง 3.11

ตารางที่ 3.11 ตัวอย่างปัญหา Set covering ในการจัดลำดับความสำคัญกรณีทดสอบ

กรณีทดสอบ	ครอบคลุมเหตุการณ์					
	1	2	3	4	5	6
T1	X		X		X	X
T2				X	X	X
T3	X	X	X			

จากตาราง 3.11 พบว่าแต่ละกรณีทดสอบมีเหตุการณ์ครอบคลุมแตกต่างกัน ในการจัดลำดับความสำคัญแบบเพิ่มนั้น เป็นการพิจารณาข้อมูลของชุดทดสอบทั้งหมด ดังนั้นทางเลือกในการจัดลำดับความสำคัญโดยให้ครอบคลุมเหตุการณ์ทั้งหมดที่เป็นไปได้อย่างรวดเร็ว คือ ลำดับ T2, T3 หรือ T3, T2 ซึ่งจะครอบคลุมเหตุการณ์ทั้งหมดที่เป็นไปได้ในชุดทดสอบ แต่จะทราบได้อย่างไรว่าควรเลือกกรณีทดสอบใดบ้าง เพราะมีความเป็นไปได้ในการจัดลำดับกรณีทดสอบถึง N! เมื่อ N แทนจำนวนกรณีทดสอบ ดังนั้นในกรณีที่มีกรณีทดสอบมากๆ จึงเป็นไปได้ยากที่จะคำนวณความเป็นไปได้ทั้งหมด

แนวทางในการแก้ไขปัญหาที่เลือกใช้ คือ การใช้อัลกอริทึมแบบ Greedy ในการเลือกกรณีทดสอบ โดยเลือกกรณีทดสอบที่มีความครอบคลุมมากที่สุดก่อน จากนั้นจึงพิจารณากรณีทดสอบที่เหลือ ในกรณีที่มีกรณีทดสอบมีความครอบคลุมเท่ากัน จะใช้การสุ่มเลือก ซึ่งหากใช้วิธีการดังกล่าวในการเลือกกรณีทดสอบจากตาราง 3.11 จะได้ลำดับการเลือกเป็น T1, T2, T3 ที่ไม่ใช่ลำดับที่สั้นที่สุด แต่สำหรับเหตุผลที่เราเลือกใช้อัลกอริทึมแบบ Greedy ในการแก้ปัญหาเนื่องจากเหตุผลดังนี้

1) การจัดลำดับความสำคัญกรณีทดสอบมีเป้าหมายเพื่อให้สามารถค้นพบข้อผิดพลาดอย่างรวดเร็ว การจัดลำดับความสำคัญโดยกรณีทดสอบที่มีความครอบคลุมตามเกณฑ์ความครอบคลุมมากมีระดับความสำคัญสูงและถูกทดสอบก่อน ทำให้โอกาสในการค้นพบข้อผิดพลาดมาก

2) ผู้ทดสอบเป็นผู้เลือกกรณีทดสอบที่จะทำการทดสอบหลังจากการจัดลำดับความสำคัญ สำหรับแผนการทดสอบบางรูปแบบผู้ทดสอบไม่ทราบจำนวนกรณีทดสอบที่แน่ชัดก่อนทำการทดสอบ เช่น ทดสอบไปเรื่อยๆจนกว่าจะใช้ทรัพยากรครบตามที่วางแผน หรือทดสอบจนพบข้อผิดพลาดตามจำนวนที่กำหนด หรือทดสอบจนอัตราการพบข้อผิดพลาดต่ำกว่าอัตราที่กำหนดไว้ เป็นต้น ซึ่งวิธีการต่างๆเหล่านี้เราจะไม่รู้ว่าต้องทดสอบกรณีทดสอบจำนวนเท่าใดจนกระทั่งทดสอบจริง อัลกอริทึมแบบ Greedy ไม่มีการระบุจำนวนกรณีทดสอบที่แน่นอนที่ต้องทดสอบ หรือต้องระบุจำนวนกรณีทดสอบที่ต้องการล่วงหน้า แต่จะให้ค่าที่ค่อนข้างดี ไม่ว่าเราจะเลือกทดสอบด้วยกรณีทดสอบจำนวนเท่าใดก็ตาม ซึ่งสะดวกต่อผู้ทดสอบที่จะนำไปปรับใช้กับแผนการทดสอบที่เลือกใช้

3) ใช้เวลาและทรัพยากรในการคำนวณไม่มากนัก

สำหรับผลกระทบจากการเลือกใช้วิธีการดังกล่าวในการแก้ปัญหา คือ ค่าความสำคัญของกรณีทดสอบในชุดทดสอบหนึ่งๆจะไม่คงที่ ขึ้นกับลำดับของกรณีทดสอบในชุดทดสอบที่เข้าสู่กระบวนการจัดลำดับความสำคัญ ดังนั้นการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มจึงอาจให้ผลลัพธ์ในการจัดลำดับความสำคัญที่แตกต่างกันได้ ตามการเปลี่ยนแปลงของลำดับกรณีทดสอบในชุดทดสอบ

เนื่องจากแนวทางการแก้ปัญหา Set covering มีหลายแนวทาง ดังนั้นจึงมีความจำเป็นต้องมีการศึกษาถึงแนวทางอื่นๆในแก้ปัญหา เช่น การเลือกส่วนที่มีกรณีทดสอบครอบคลุมน้อยก่อน หรือการใช้อัลกอริทึมแบบ Greedy ร่วมกับวิธีการต่างๆ ซึ่งมีความเป็นไปได้ที่จะพัฒนาประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบให้ดียิ่งขึ้นต่อไปได้

บทที่ 4

การออกแบบและพัฒนาเครื่องมือ

4.1 การวิเคราะห์และออกแบบเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ

การวิเคราะห์และออกแบบเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ใช้การออกแบบและพัฒนาโปรแกรมเชิงวัตถุ โดยใช้แผนภาพยูเอ็มแอลเป็นหลักในการออกแบบ

4.1.1 ระบบงานการจัดลำดับความสำคัญกรณีทดสอบ

ขั้นตอนการจัดลำดับความสำคัญกรณีทดสอบ นอกจากในส่วนที่เป็นหน้าที่ของเครื่องมือแล้ว ยังมีส่วนที่เป็นหน้าที่ของผู้ทดสอบที่ต้องจัดการด้วย ขั้นตอนการทำงานทั้งหมดในการจัดลำดับความสำคัญกรณีทดสอบ แสดงในตาราง 4.1

ตารางที่ 4.1 ขั้นตอนการจัดลำดับความสำคัญกรณีทดสอบ

ผู้ทดสอบ	เครื่องมือ
1. เตรียมชุดทดสอบให้อยู่ในรูปแบบที่กำหนด	
2. กำหนดวิธีการจัดลำดับความสำคัญกรณีทดสอบ	
	3. จัดลำดับความสำคัญกรณีทดสอบ
4. นำข้อมูลลำดับความสำคัญไปใช้	

จากตาราง 4.1 ถือเป็นหน้าที่ของผู้ทดสอบในการเตรียมข้อมูลชุดทดสอบให้อยู่ในลักษณะที่สามารถนำไปใช้กับเครื่องมือได้ ซึ่งรวมถึงการกำหนดภารกิจในการจัดลำดับกรณีทดสอบที่ใช้ความครอบคลุมแบบภารกิจหรือความสัมพันธ์ระหว่างภารกิจ นอกจากนี้ผู้ทดสอบยังเป็นผู้กำหนดว่าเครื่องมือจะจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการใด ส่วนเครื่องมือมีหน้าที่เพียงการจัดลำดับความสำคัญกรณีทดสอบตามที่ผู้ใช้ต้องการเท่านั้น เมื่อจัดลำดับความสำคัญเสร็จแล้วผู้ใช้อาจต้องมีการดำเนินการ เพื่อนำข้อมูลลำดับความสำคัญไปประยุกต์ใช้ตามที่ต้องการ

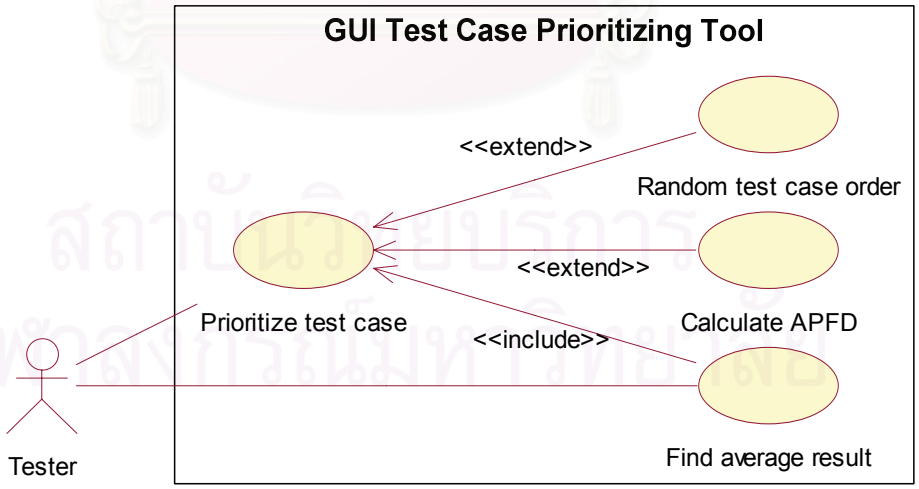
4.1.2 หน้าที่การทำงานของเครื่องมือ

เครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ ต้องสามารถทำหน้าที่ ดังนี้

- ทำหน้าที่จัดลำดับความสำคัญกรณีทดสอบตามวิธีการที่ได้กำหนดไว้ พร้อมทั้งสามารถบันทึกกรณีทดสอบที่จัดลำดับความสำคัญแล้วลงในแฟ้มข้อมูล
- คำนวณค่า APFD ซึ่งเป็นมาตรวัดซอฟต์แวร์ที่ใช้ประเมินประสิทธิภาพในการจัดลำดับความสำคัญกรณีทดสอบในงานวิจัยนี้ได้โดยอัตโนมัติ
- วัดระยะเวลาที่ใช้ในการจัดลำดับความสำคัญของกรณีทดสอบแต่ละวิธีการได้
- สุ่มเปลี่ยนแปลงลำดับของกรณีทดสอบในชุดทดสอบที่ยังไม่ได้ผ่านการจัดลำดับความสำคัญ เพื่อประโยชน์ในการทำการทดลองซ้ำ เพื่อหาค่าเฉลี่ย
- ทำงานต่อเนื่อง โดยเก็บค่า APFD ที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบไว้ในแฟ้มข้อมูล เพื่อประโยชน์ในการหาค่าเฉลี่ย

4.1.3 แผนภาพยูสเคส

แผนภาพยูสเคสเป็นแผนภาพที่แสดงขอบเขตและความสัมพันธ์ระหว่างกิจกรรมในระบบจากมุมมองของผู้ใช้งาน สำหรับเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ สามารถสร้างยูสเคสได้ดังรูป 4.1



รูปที่ 4.1 แผนภาพยูสเคสของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

จากรูป 4.1 ผู้ทดสอบ (Tester) สามารถจัดลำดับความสำคัญกรณีทดสอบ (Prioritize test case) รวมถึงสามารถเลือกที่จะสลับลำดับกรณีทดสอบแบบสุ่ม (Random test case order) เพื่อให้ได้ผลลัพธ์ที่แตกต่างไป หรือเลือกให้คำนวณเมตริกซ์ APFD โดยอัตโนมัติ (Calculate APFD) ได้ สำหรับกรณีที่ต้องการทดลองหาค่าเฉลี่ยของเวลาหรือค่า APFD ที่ต้องจัดลำดับความสำคัญหลายๆครั้ง (Find average result) ผู้ทดสอบสามารถกำหนดให้ทำงานโดยอัตโนมัติได้ สำหรับรายละเอียดของยูสเคสแสดงในตาราง 4.2 ถึง 4.5

ตารางที่ 4.2 รายละเอียดยูสเคส Prioritize test case

Use case name	Prioritize test case	ID	UC1
Primary actor	ผู้ทดสอบ		
Description	จัดลำดับความสำคัญกรณีทดสอบตามวิธีการที่กำหนด		
Trigger	ผู้ทดสอบเริ่มงานจัดลำดับความสำคัญกรณีทดสอบ	Type	ภายนอก
Relationship	Extend: Random test case order, Calculate APFD		
Pre-Condition	แฟ้มข้อมูลชุดทดสอบที่ต้องการจัดลำดับความสำคัญจัดเก็บอยู่ในรูปแบบที่กำหนด		
Normal flow of events			
1. ผู้ทดสอบเปิดแฟ้มข้อมูลชุดทดสอบ 2. เครื่องมือแสดงผลกรณีทดสอบทั้งหมดในแฟ้ม 3. ผู้ทดสอบสั่งจัดลำดับความสำคัญกรณีทดสอบตามวิธีการที่กำหนด 4. เครื่องมือแสดงกรณีทดสอบ ผลการจัดลำดับความสำคัญทั้งหมด และระยะเวลาที่ใช้ 5. ผู้ทดสอบจบการทำงาน			
Post-Condition	แฟ้มข้อมูลชุดทดสอบที่จัดลำดับความสำคัญแล้วในกรณีที่มีการบันทึกข้อมูล		
Subflows			
-			
Alternative/exception flows			
3.1a. ผู้ทดสอบสั่งสลับกรณีทดสอบแบบสุ่ม ทำงานในยูสเคส Random test case order เมื่อเสร็จกลับทำงานต่อในขั้น 3 3.2a. ผู้ทดสอบสั่งให้เครื่องมือคำนวณค่า APFD ทำงานในยูสเคส Calculate APFD 5a. ผู้ทดสอบสั่งบันทึกกรณีทดสอบและลำดับความสำคัญลงในแฟ้มข้อมูลใหม่			

ตารางที่ 4.3 รายละเอียดยูสเคส Random test case order

Use case name	Random test case order	ID	UC2
Primary actor	ผู้ทดสอบ		
Description	สลับลำดับกรณีทดสอบในชุดทดสอบด้วยวิธีสุ่ม		
Trigger	ผู้ทดสอบสั่งสลับลำดับกรณีทดสอบ	Type	ภายนอก
Relationship	-		
Pre-Condition	มีแฟ้มข้อมูลชุดทดสอบเปิดทำงานอยู่		
Normal flow of events			
1. ผู้ทดสอบสั่งสลับลำดับกรณีทดสอบ 2. เครื่องมือแสดงผลการสลับลำดับ 3. ผู้ทดสอบทำหน้าที่อื่นต่อไป			
Post-Condition	กรณีทดสอบในชุดทดสอบที่เปิดทำงานอยู่ ถูกสลับลำดับ		
Subflows			
-			
Alternative/exception flows			
-			

ตารางที่ 4.4 รายละเอียดยูสเคส Calculate APFD

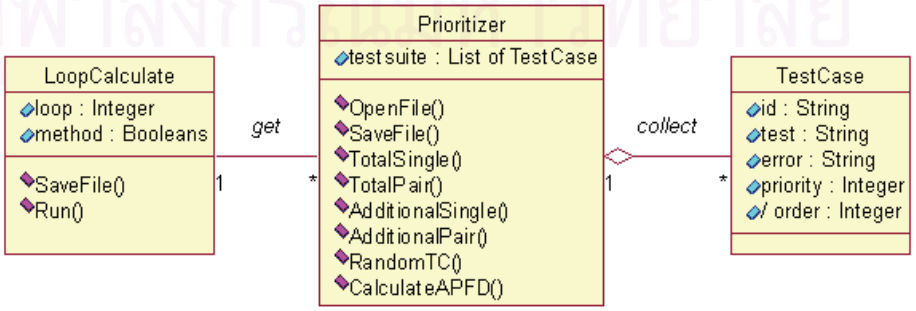
Use case name	Calculate APFD	ID	UC3
Primary actor	ผู้ทดสอบ		
Description	คำนวณค่าเมตริกซ์ APFD จากชุดทดสอบที่กำหนด		
Trigger	ผู้ทดสอบสั่งคำนวณค่า APFD	Type	ภายนอก
Relationship	-		
Pre-Condition	มีแฟ้มข้อมูลชุดทดสอบเปิดทำงานอยู่		
Normal flow of events			
1. ผู้ทดสอบสั่งคำนวณค่า APFD 2. เครื่องมือจัดลำดับความสำคัญกรณีทดสอบตามวิธีที่กำหนด แล้วคำนวณค่า APFD 3. แสดงค่า APFD ของวิธีการจัดลำดับความสำคัญกรณีทดสอบทุกประเภท 4. ผู้ทดสอบทำหน้าที่อื่นต่อไป			
Post-Condition	-		
Subflows			
-			
Alternative/exception flows			
-			

ตารางที่ 4.5 รายละเอียดยูสเคส Find average result

Use case name	Find average result	ID	UC4
Primary actor	ผู้ทดสอบ		
Description	หาค่าเฉลี่ยในการทดสอบวิธีการจัดลำดับความสำคัญแบบต่างๆ		
Trigger	ผู้ทดสอบเริ่มการทดสอบ	Type	ภายนอก
Relationship	Include: Prioritize test case		
Pre-Condition	แฟ้มข้อมูลชุดทดสอบที่ต้องการจัดลำดับความสำคัญจัดเก็บอยู่ในรูปแบบที่กำหนด		
Normal flow of events			
1. ผู้ทดสอบเปิดแฟ้มข้อมูลชุดทดสอบ 2. เครื่องมือแสดงผลกรณีทดสอบทั้งหมดในแฟ้ม 3. ผู้ทดสอบกำหนดจำนวนครั้ง แฟ้มที่จัดเก็บผลลัพธ์ และวิธีการเรียงลำดับที่ต้องการ 4. เครื่องมือเริ่มจัดลำดับกรณีทดสอบ คำนวณค่า APFD ตามกำหนด บันทึกลงในแฟ้ม 5. เครื่องมือสลับลำดับกรณีทดสอบ แล้วย้อนกลับไปทำข้อ 4 จนครบจำนวนครั้งที่กำหนด 6. เครื่องมือรายงานผลการทำงานให้ผู้ทดสอบทราบ 7. ผู้ทดสอบจบการทำงาน			
Post-Condition	แฟ้มข้อมูลค่าเมตริกซ์ APFD ของชุดทดสอบที่กำหนด		
Subflows			
-			
Alternative/exception flows			
-			

4.1.4 แผนภาพคลาส

แผนภาพคลาสแสดงถึงโครงสร้างและความสัมพันธ์ระหว่างวัตถุในระบบเครื่องมือจัดลำดับความสำคัญกรณีทดสอบมีคลาสที่สำคัญอยู่ 3 คลาส แสดงในรูป 4.2 ซึ่งรายละเอียดของคลาส มีดังนี้



รูปที่ 4.2 แผนภาพคลาสของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

1) คลาส Test case ทำหน้าที่เป็นโครงสร้างข้อมูลใช้เก็บข้อมูลกรณีทดสอบ ประกอบด้วยข้อมูลดังนี้

- id รหัสประจำตัวกรณีทดสอบ
- test ข้อมูลนำเข้าของกรณีทดสอบ ประกอบด้วยสถานะเริ่มต้นและลำดับเหตุการณ์หรือภารกิจ
- error ข้อผิดพลาดที่กรณีทดสอบนั้นค้นพบ
- priority ลำดับความสำคัญ
- order ลำดับในการทดสอบ ระบุโดยใช้ระดับความสำคัญ

2) คลาส Prioritizer มีหน้าที่จัดการกับชุดทดสอบ จัดลำดับความสำคัญกรณีทดสอบ และคำนวณค่าเมตริกซ์ APFD

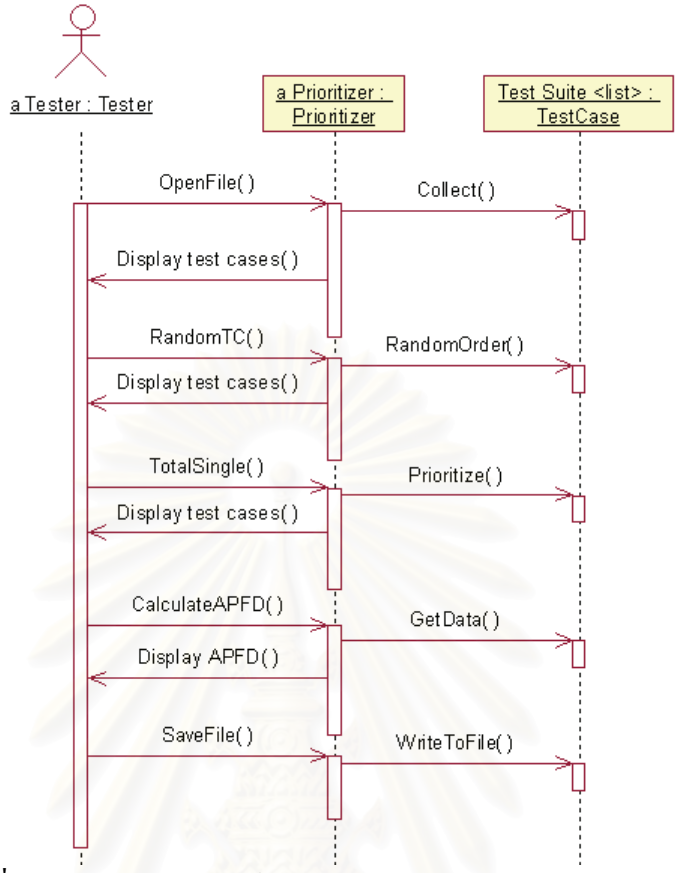
3) คลาส LoopCalculate มีหน้าที่เก็บข้อมูลการทดสอบวิธีการจัดลำดับความสำคัญแบบต่างๆ โดยจัดเก็บค่า APFD ในแฟ้มข้อมูลที่กำหนด

4.1.5 ลักษณะการทำงานของเครื่องมือ

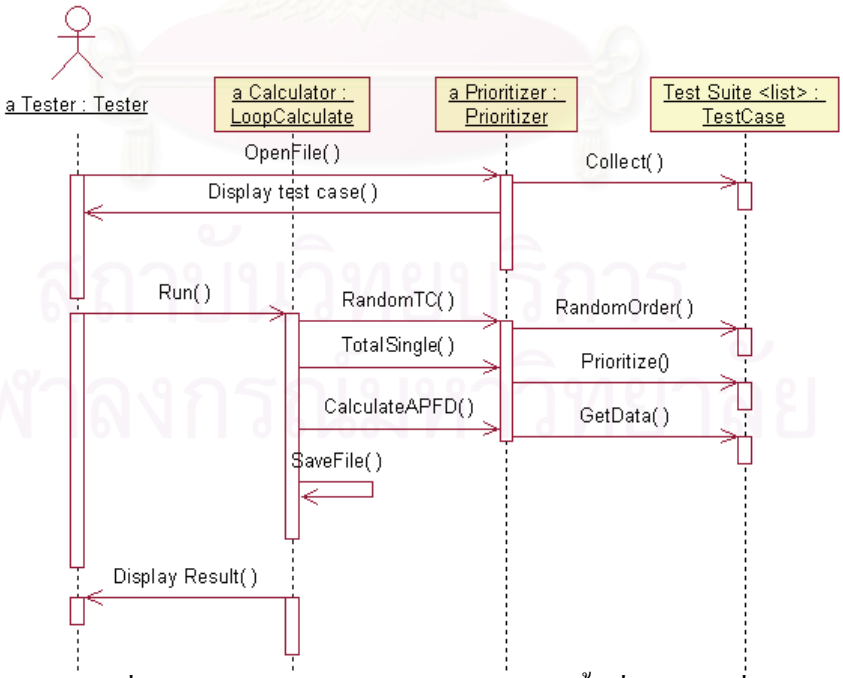
ขั้นตอนการทำงานที่สำคัญของเครื่องมือแบ่งออกเป็น 2 ส่วนได้แก่ ส่วนจัดลำดับความสำคัญกรณีทดสอบ และส่วนการทดสอบซ้ำเพื่อหาค่าเฉลี่ย โดยการทำงานทั้ง 2 ส่วนสามารถเขียนเป็นแผนภาพซีเควนซ์ ได้ดังนี้

จากรูป 4.3 ลำดับการทำงานในการจัดลำดับความสำคัญกรณีทดสอบเริ่มจากการเปิดแฟ้มข้อมูลชุดทดสอบ จากนั้นเป็นขั้นตอนในการสุ่มลำดับกรณีทดสอบ ซึ่งจะทำหรือไม่ก็ได้ แล้วจึงเป็นขั้นตอนการจัดลำดับความสำคัญ โดยในรูปจัดลำดับความสำคัญโดยใช้เกณฑ์ความครอบคลุมตามเหตุการณ์หรือภารกิจแบบเบ็ดเสร็จ จากนั้นเป็นขั้นตอนการคำนวณค่า APFD ซึ่งขึ้นกับผู้ทดสอบว่าจะดำเนินการหรือไม่ และในขั้นตอนท้ายสุดเป็นการบันทึกผลการจัดลำดับความสำคัญในแฟ้มข้อมูล

จากรูป 4.4 ลำดับการทำงานของการทดสอบซ้ำเพื่อหาค่าเฉลี่ย เริ่มจากการเปิดแฟ้มข้อมูลชุดทดสอบเช่นเดียวกับการจัดลำดับความสำคัญ จากนั้นเป็นขั้นตอนการทำงานซ้ำ ซึ่งมีการทำงานเช่นเดียวกับการจัดลำดับความสำคัญกรณีทดสอบ แต่ไม่มีการแสดงผลลัพธ์ในแต่ละขั้นตอน และมีการบันทึกค่า APFD ลงในแฟ้มข้อมูล เมื่อทำงานซ้ำครบตามจำนวนที่กำหนด เครื่องมือจะแสดงผลการทำงานให้ผู้ทดสอบทราบ



รูปที่ 4.3 แผนภาพซีเควนซ์ของการจัดลำดับความสำคัญกรณีทดสอบ



รูปที่ 4.4 แผนภาพซีเควนซ์ของการทดสอบซ้ำเพื่อหาค่าเฉลี่ย

4.2 สภาพแวดล้อมในการพัฒนาเครื่องมือ

4.2.1 ฮาร์ดแวร์

ฮาร์ดแวร์ที่ใช้ในพัฒนาและทดสอบการทำงานของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ ประกอบด้วย

- เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลเพนเทียมโฟร์ 3.0 กิกะเฮิรตซ์ (Pentium4 3.0 GHz.)
- หน่วยความจำหลัก (RAM) 1.0 กิกะไบต์ (1 GB)
- ฮาร์ดดิสก์ (Harddisk)160 กิกะไบต์ (160 GB)

4.2.2 ซอฟต์แวร์

ซอฟต์แวร์ที่ใช้ออกแบบ พัฒนา จัดการข้อมูล และทดสอบการทำงานของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ ประกอบด้วย

- ระบบปฏิบัติการ ไมโครซอฟท์วินโดวส์เอ็กซ์พี โพรเฟชชันแนล (Microsoft Windows XP Professional)
- เครื่องมือพัฒนาด้วยภาษาวิชัวล์ซีพลัสพลัส โดยใช้โปรแกรมไมโครซอฟท์วิชัวล์สตูดิโอ 2005 (Microsoft Visual Studio 2005)
- เครื่องมือจัดการและเตรียมข้อมูลทดสอบ ใช้โปรแกรมไมโครซอฟท์เอ็กเซล 2002 (Microsoft Excel 2002 SP3)

4.3 โครงสร้างกรณีทดสอบ

เครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ทำงานเกี่ยวกับกรณีทดสอบเป็นหลักทั้งในฐานะข้อมูลนำเข้าและข้อมูลส่งออก(Output) โครงสร้างข้อมูลกรณีทดสอบกำหนดขึ้นตามนิยามของกรณีทดสอบในบทที่ 3 หัวข้อ 3.2 โดยพิจารณาเฉพาะส่วนที่เครื่องมือให้ความสนใจ หากกรณีทดสอบมีส่วนประกอบอื่นๆ เช่น ผลลัพธ์ที่คาด อยู่ด้วยจำเป็นต้องมีการเตรียมข้อมูล เพื่อกำจัดส่วนที่ไม่ต้องการออกไป

โครงสร้างของกรณีทดสอบที่เครื่องมือยึดเป็นหลักในการทำงาน ได้แก่

- รหัสประจำตัวกรณีทดสอบ (ID) มีลักษณะเป็นตัวอักษรหรือตัวเลขต่อเนื่องกัน ใช้ระบุตัวตนของกรณีทดสอบ ใช้งานเฉพาะส่วนแสดงผล ไม่ส่งผลกระทบต่อการจัดลำดับความสำคัญกรณีทดสอบ

- ลำดับข้อมูลทดสอบ (Test sequence) ประกอบด้วยสถานะเริ่มต้นตามด้วยลำดับของเหตุการณ์หรือภารกิจ ใช้เครื่องหมายจุลภาค (,) เป็นตัวคั่น กรณีทดสอบทุกกรณีจำเป็นต้องมีข้อมูลนี้ ใช้เป็นข้อมูลหลักในการจัดลำดับความสำคัญกรณีทดสอบ
- ข้อผิดพลาดที่พบ (Error) ข้อผิดพลาดที่กรณีทดสอบนั้นค้นพบ สำหรับกรณีทดสอบที่พบข้อผิดพลาดมากกว่า 1 กรณี ให้ใช้เครื่องหมายจุลภาค (,) เป็นตัวคั่น ในกรณีที่ไม่พบข้อผิดพลาดหรือไม่มีข้อมูลให้ใช้เครื่องหมายยัติภังค์ (-) ใช้ในการคำนวณค่าเมตริกซ์ APFD
- ลำดับความสำคัญ (Priority) ลำดับความสำคัญที่ได้มาจากการจัดลำดับกรณีทดสอบเป็นค่าตัวเลข ซึ่งกรณีทดสอบอาจมีลำดับความสำคัญเท่ากันได้หลายกรณี กรณีที่ไม่มีข้อมูลนี้ให้ใช้เครื่องหมายยัติภังค์ (-) แทน
- ลำดับการทดสอบ (Order) ลำดับการทดสอบที่ได้จากการจัดลำดับความสำคัญของกรณีทดสอบทั้งหมดในชุดทดสอบ กรณีที่มีความสำคัญเท่ากันจะจัดตามลำดับในชุดทดสอบ ได้จากการคำนวณและแสดงผลในโปรแกรมเท่านั้น ไม่มีการอ่านหรือบันทึกข้อมูลนี้

สำหรับสถานะเริ่มต้น เหตุการณ์ ภารกิจหรือข้อผิดพลาดให้แทนด้วยชื่อเฉพาะที่ประกอบด้วยตัวอักษร ตัวเลข เครื่องหมาย ( ) “ _ หรือช่องว่าง โดยไม่ซ้ำกับสถานะเริ่มต้น เหตุการณ์ ภารกิจหรือข้อผิดพลาดอื่น

กรณีทดสอบหนึ่งๆประกอบด้วยองค์ประกอบเหล่านี้ตามลำดับ ยกเว้นลำดับการทดสอบ ตัวอย่างกรณีทดสอบแสดงดังตาราง 4.6

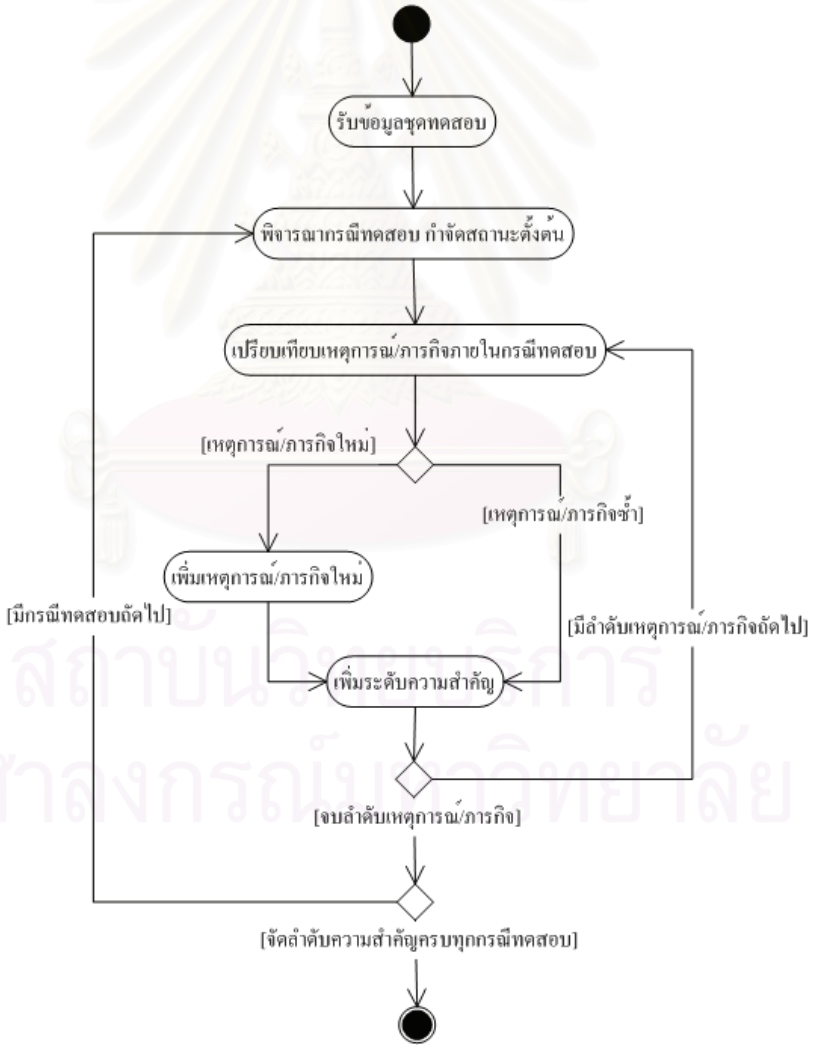
ตารางที่ 4.6 ตัวอย่างกรณีทดสอบ

รหัสประจำตัว (ID)	ลำดับข้อมูลทดสอบ (Test sequence)	ข้อผิดพลาดที่พบ (Error)	ลำดับความสำคัญ (Priority)
TC1	S0,E1,E2,E3	D1,D2	2
TC2	S0,E2,E3,E4	-	1

กรณีทดสอบนำเข้าต้องประกอบด้วยข้อมูลรหัสประจำตัว และลำดับข้อมูลทดสอบเป็นอย่างน้อย สำหรับข้อมูลอื่นๆจะมีหรือไม่ก็ได้ ส่วนข้อมูลส่งออกนั้นจะประกอบด้วยข้อมูลครบทั้งหมดเสมอ

4.4 ขั้นตอนการจัดลำดับความสำคัญกรณีทดสอบ

การจัดลำดับความสำคัญกรณีทดสอบทั้งหมด 8 ประเภท แบ่งเป็นระดับเหตุการณ์ 4 ประเภท และระดับภารกิจ 4 ประเภท ซึ่งทั้งสองรูปแบบมีลักษณะการทำงานเหมือนกัน แตกต่างกันเฉพาะข้อมูลที่ใช้จัดลำดับความสำคัญ โดยระดับเหตุการณ์กรณีทดสอบมีลักษณะเป็นลำดับของสถานะเริ่มต้นกับเหตุการณ์ ส่วนระดับภารกิจกรณีทดสอบมีลักษณะเป็นลำดับของสถานะเริ่มต้นกับภารกิจ ซึ่งการกำหนดเหตุการณ์หรือภารกิจขึ้นกับผู้ทดสอบ ไม่เกี่ยวข้องกับการทำงานของเครื่องมือ จากเหตุผลดังกล่าว เครื่องมือจึงแบ่งประเภทการจัดลำดับความสำคัญกรณีทดสอบออกเป็น 4 ประเภท ได้แก่



รูปที่ 4.5 แผนภาพลำดับกิจกรรมวิธีจัดลำดับความสำคัญกรณีทดสอบเชิงเดี่ยวแบบเบ็ดเสร็จ

4.4.1 วิธีการจัดลำดับความสำคัญกรณีทดสอบเชิงเดี่ยวแบบเบ็ดเสร็จ (Total Single)

ใช้ในการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเบ็ดเสร็จ และการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเบ็ดเสร็จ ขึ้นกับประเภทข้อมูลในกรณีทดสอบ

ขั้นตอนการทำงานของวิธีการจัดลำดับความสำคัญแบบนี้ แสดงด้วยแผนภาพลำดับกิจกรรม (Activity diagram) ในรูป 4.5 สำหรับวิธีการนี้จะไม่สนใจสถานะเริ่มต้น และถ้าพบเหตุการณ์หรือภารกิจที่ซ้ำในกรณีทดสอบเดียวกันจะเพิ่มระดับความสำคัญกรณีทดสอบในอัตราส่วนที่น้อยกว่าเหตุการณ์หรือภารกิจใหม่

4.4.2 วิธีจัดลำดับความสำคัญกรณีทดสอบเชิงคู่แบบเบ็ดเสร็จ (Total Pair)

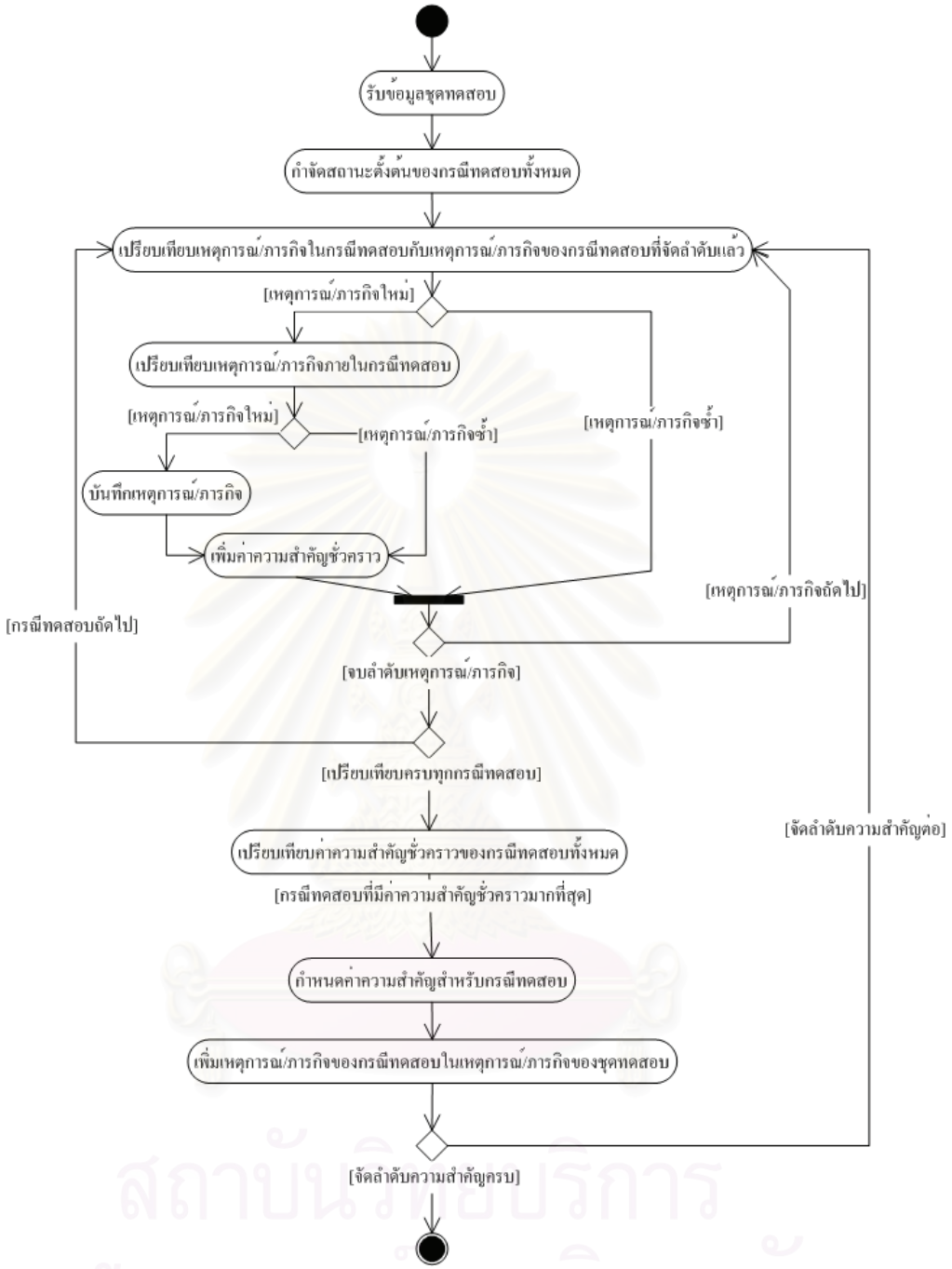
ใช้ในการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเบ็ดเสร็จ และการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเบ็ดเสร็จ ขึ้นกับประเภทข้อมูลในกรณีทดสอบ

ขั้นตอนการทำงานใกล้เคียงกับการทำงานของวิธีการจัดลำดับความสำคัญเชิงเดี่ยวแบบเบ็ดเสร็จ แต่จะพิจารณาสถานะเริ่มต้นด้วย (ไม่มีขั้นตอนกำจัดสถานะเริ่มต้น) และพิจารณาในระดับคู่ของเหตุการณ์หรือภารกิจที่อยู่ติดกัน

4.4.3 วิธีจัดลำดับความสำคัญกรณีทดสอบเชิงเดี่ยวแบบเพิ่ม (Additional Single)

ใช้ในการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม และการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม ขึ้นกับประเภทข้อมูลในกรณีทดสอบ

ขั้นตอนการทำงานของวิธีการจัดลำดับความสำคัญแบบนี้ แสดงด้วยแผนภาพลำดับกิจกรรมในรูป 4.6 สำหรับวิธีการนี้จะไม่สนใจสถานะเริ่มต้นเช่นเดียวกับวิธีจัดลำดับความสำคัญเชิงเดี่ยวแบบอื่นๆ นอกจากนี้ยังมีการเปรียบเทียบทั้งในระดับชุดทดสอบ เพื่อระบุเหตุการณ์หรือภารกิจที่ถูกครอบคลุมด้วยกรณีทดสอบที่ได้จัดลำดับความสำคัญไปแล้ว ซึ่งจะไม่นำมาคำนวณความสำคัญ และระดับกรณีทดสอบ สำหรับเหตุการณ์หรือภารกิจใหม่ซึ่งอาจปรากฏมากกว่า 1 ครั้งในกรณีทดสอบหนึ่งๆ ซึ่งจะถูกนำมาคำนวณความสำคัญ เช่นเดียวกับการจัดลำดับความสำคัญแบบเบ็ดเสร็จ



รูปที่ 4.6 แผนภาพลำดับกิจกรรมวิธีจัดลำดับความสำคัญกรณีทดสอบเชิงเดี่ยวแบบเพิ่ม

4.4.4 วิธีจัดลำดับความสำคัญกรณีทดสอบเชิงคู่แบบเพิ่ม (Additional Pair)

ใช้ในการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม และการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม ขึ้นกับประเภทข้อมูลในกรณีทดสอบ

ขั้นตอนการทำงานใกล้เคียงกับการทำงานของวิธีการจัดลำดับความสำคัญเชิงเดี่ยวแบบเพิ่ม แต่จะพิจารณาสถานะเริ่มต้นด้วย (ไม่มีขั้นตอนกำจัดสถานะเริ่มต้น) และพิจารณาในระดับคู่ของเหตุการณ์หรือภารกิจที่อยู่ติดกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การทดสอบวิธีการจัดลำดับความสำคัญของกรณีทดสอบ สำหรับส่วนต่อประสานกราฟิกกับผู้ใช้

5.1 วัตถุประสงค์การทดสอบ

การทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้มีวัตถุประสงค์เพื่อเปรียบเทียบประสิทธิภาพในด้านต่างๆของวิธีการจัดลำดับความสำคัญกรณีทดสอบ เพื่อยืนยันแนวความคิดเกี่ยวกับการนำการจัดลำดับความสำคัญกรณีทดสอบไปใช้กับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ และเพื่อให้สามารถนำวิธีการจัดลำดับความสำคัญกรณีทดสอบที่พัฒนาขึ้นไปใช้งานหรือประยุกต์ใช้ได้อย่างเหมาะสมกับงานและสถานการณ์รูปแบบต่างๆ อนึ่งการทดสอบใช้เครื่องมือจัดลำดับความสำคัญกรณีทดสอบที่ได้พัฒนาขึ้น (รายละเอียดในบทที่ 4) เป็นหลักในการทดสอบ

การทดสอบวิธีวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ในงานวิจัยนี้ได้กำหนดเป้าหมายการทดสอบไว้ 2 ประการ ได้แก่

- ทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการค้นพบข้อผิดพลาดระหว่างการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ตามปกติและการทดสอบที่ใช้วิธีการจัดลำดับความสำคัญกรณีทดสอบแบบต่างๆ
- ทดสอบเพื่อเปรียบเทียบระยะเวลาที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบตามวิธีการจัดลำดับความสำคัญแบบต่างๆ

ดังนั้นเพื่อให้บรรลุเป้าหมายดังกล่าว การทดสอบจึงถูกแบ่งออกเป็น 2 ส่วน คือ

1) การทดสอบเปรียบเทียบประสิทธิภาพ

จุดมุ่งหมายเพื่อวัดประสิทธิภาพในการค้นพบข้อผิดพลาดของกรณีทดสอบทั้งที่ผ่านและไม่ผ่านการจัดลำดับความสำคัญ การทดสอบกระทำกับตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้จริงของโปรแกรมใช้งานตามปกติทั่วไป โดยลักษณะการทดสอบใกล้เคียงกับการนำการจัดลำดับความสำคัญกรณีทดสอบไปใช้กับการทำงานทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จริง มีการวัดประสิทธิภาพโดยใช้มาตรวัดซอฟต์แวร์ที่กำหนด

2) การทดสอบเปรียบเทียบระยะเวลาการทำงาน

จุดมุ่งหมายเพื่อวัดระยะเวลาการทำงานของวิธีจัดลำดับความสำคัญกรณีทดสอบแบบต่างๆ การทดสอบอาศัยชุดทดสอบที่สร้างขึ้น โดยไม่อิงกับส่วนต่อประสานผู้ใช้จริง แต่จะเน้นที่การเปรียบเทียบอัตราการเพิ่มของระยะเวลา เมื่อตัวแปรต่างๆเปลี่ยนแปลงไป

5.2 ขั้นตอนการทดสอบ

การทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้แบ่งออก 2 ส่วน แต่ละส่วนแยกจากกันอย่างเด็ดขาด โดยมีขั้นตอนการทดสอบที่แตกต่างกันและไม่มีการใช้ข้อมูลการทดสอบร่วมกันแต่อย่างใด สำหรับรายละเอียดขั้นตอนการทดสอบแต่ละส่วนมีดังนี้

5.2.1 ขั้นตอนการทดสอบเปรียบเทียบประสิทธิภาพ

1) เลือกตัวอย่างโปรแกรมที่จะนำมาทดสอบ โดยตัวอย่างโปรแกรมต้องประกอบด้วยส่วนต่อประสานกราฟิกกับผู้ใช้เป็นหลัก มีลักษณะการทำงานที่ทำความเข้าใจได้ง่าย และมีข้อผิดพลาดในการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้

2) ออกแบบและสร้างชุดทดสอบสำหรับทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมที่นำมาทดสอบ โดยออกแบบเพื่อทดสอบทั้งโปรแกรมหรือส่วนใดส่วนหนึ่งของโปรแกรมตามความเหมาะสม

3) นำชุดทดสอบที่ได้ไปทดสอบกับโปรแกรมตัวอย่าง เพื่อบันทึกข้อมูลข้อผิดพลาดที่ถูกค้นพบโดยกรณีทดสอบแต่ละกรณี

4) นำกรณีทดสอบในชุดทดสอบมาจัดลำดับความสำคัญตามวิธีการที่กำหนดไว้ และสำหรับวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มจำเป็นต้องมีการจัดลำดับความสำคัญหลายครั้ง เพื่อหาค่าเฉลี่ย เนื่องจากผลการจัดลำดับความสำคัญแต่ละครั้งจะแตกต่างไปตามลำดับของกรณีทดสอบในชุดทดสอบ

5) เปรียบเทียบผลลัพธ์ที่ได้จากการจัดลำดับความสำคัญ โดยใช้ข้อมูลจากการทดสอบในข้อ 3

6) สรุปผลการทดสอบ

5.2.2 ขั้นตอนการทดสอบเปรียบเทียบระยะเวลาการทำงาน

1) สร้างชุดทดสอบที่ประกอบด้วยกรณีทดสอบที่มีลักษณะตามที่กำหนด โดยไม่มีความเกี่ยวข้องกับส่วนต่อประสานกราฟิกกับผู้ใช้แต่อย่างใด เป็นเพียงชุดข้อมูลที่สร้างขึ้นเพื่อการทดลองเท่านั้น

2) นำชุดทดสอบที่ได้มาจัดลำดับความสำคัญ โดยใช้วิธีจัดลำดับความสำคัญกรณีทดสอบที่กำหนดไว้

3) เปรียบเทียบระยะเวลาเฉลี่ยที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบต่างๆ

4) สรุปผลการทดสอบ

5.3 การประเมินผลการทดสอบ

การทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ทั้ง 2 ส่วน ล้วนมีจุดมุ่งหมายเพื่อเปรียบเทียบความสามารถของวิธีการจัดลำดับความสำคัญ ในด้านประสิทธิภาพและระยะเวลาการทำงาน ดังนั้นการประเมินความสามารถหรือการประเมินผลสำหรับการทดสอบแต่ละส่วน ซึ่งใช้เป็นเครื่องบ่งชี้ถึงความสามารถและใช้ในการเปรียบเทียบกันนั้น จึงมีความสำคัญอย่างยิ่ง สำหรับการประเมินผลการทดสอบจะแบ่งออกเป็น 2 ส่วนตามการทดสอบ โดยมีรายละเอียดดังนี้

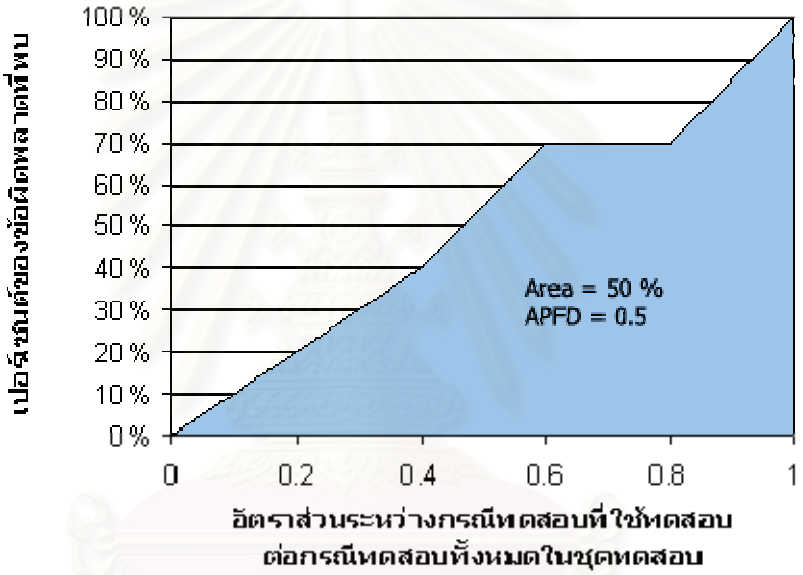
5.3.1 การประเมินผลการทดสอบเปรียบเทียบประสิทธิภาพ

ในการทดสอบเปรียบเทียบประสิทธิภาพสิ่งที่เราสนใจ คือ ประสิทธิภาพในการค้นพบข้อผิดพลาดของวิธีการจัดลำดับความสำคัญกรณีทดสอบเป็นอย่างไรเมื่อเปรียบเทียบกับการทดสอบตามปกติ และเมื่อเปรียบเทียบระหว่างวิธีการจัดลำดับความสำคัญกรณีทดสอบด้วยกันแล้ววิธีการใดเหมาะสมกับสถานการณ์แบบใด ดังนั้นปัจจัยสำคัญที่เราต้องคำนึงถึงสำหรับการประเมินการทดสอบนี้ ได้แก่ วิธีการวัดประสิทธิภาพและวิธีการที่ใช้เปรียบเทียบ

นอกจากนี้ในกรณีที่กรณีทดสอบมากกว่า 1 กรณีทดสอบมีความสำคัญเท่ากัน เราจำเป็นต้องคำนึงถึงโอกาสที่กรณีทดสอบจะถูกเลือกมาทดสอบหรือการหาค่าเฉลี่ยอีกด้วย ซึ่งปัจจัยสำคัญทั้งหมดนี้มีรายละเอียดดังนี้

1) วิธีการวัดประสิทธิภาพ

Gregg Rothermel et al. [10] ได้เสนอแนวทางการวัดประสิทธิภาพในการค้นพบข้อผิดพลาดของชุดทดสอบที่มีการจัดลำดับความสำคัญ โดยใช้เมตริกซ์ APFD เป็นมาตรวัด ซึ่งพิจารณาประสิทธิภาพในการค้นพบข้อผิดพลาดบนสมมติฐานที่ว่า ข้อผิดพลาดทั้งหมดมีความสำคัญเท่ากันและกรณีทดสอบแต่ละกรณีใช้ระยะเวลาในการทดสอบใกล้เคียงกัน ดังนั้นหากสร้างกราฟระหว่างชุดทดสอบกับเปอร์เซ็นต์ข้อผิดพลาดที่พบดังแสดงในรูป 5.1 จะสามารถเปรียบเทียบประสิทธิภาพในการค้นพบข้อผิดพลาดของชุดทดสอบที่จัดลำดับแล้วได้ โดยการเปรียบเทียบพื้นที่ใต้กราฟ



รูปที่ 5.1 แผนภูมิแสดงความสัมพันธ์ระหว่างชุดทดสอบกับเปอร์เซ็นต์ของข้อผิดพลาดที่พบ

เนื่องจากมาตรวัด APFD ไม่เกี่ยวข้องกับวิธีการหรือลักษณะเฉพาะใดๆ ในขั้นตอนการทดสอบ ทำให้สามารถนำมาใช้วัดประสิทธิภาพของการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ได้เช่นเดียวกับการใช้กับการทดสอบซอฟต์แวร์ปกติ โดยมาตรวัด APFD สามารถคำนวณได้ ดังนี้

ถ้ากำหนดให้ T แทนชุดทดสอบ

n แทนจำนวนกรณีทดสอบในชุดทดสอบ T

F แทนกลุ่มของข้อผิดพลาดทั้งหมดที่ค้นพบโดยชุดทดสอบ T

m แทนจำนวนสมาชิกของ F

TF_i แทนตำแหน่งของกรณีทดสอบกรณีแรกใน T ที่ค้นพบข้อผิดพลาด i

การจัดลำดับความสำคัญกรณีทดสอบภายใน T มีประสิทธิภาพในการค้นพบข้อผิดพลาดตามมาตรวัด APFD ดังนี้

$$APFD = 1 - \frac{TF_1 + TF_2 + ... + TF_m}{nm} + \frac{1}{2n}$$

ค่า APFD ที่ได้จะอยู่ในช่วงระหว่าง 0 ถึง 1 (ยกเว้นกรณีที่ชุดทดสอบไม่พบข้อผิดพลาด) โดยค่า APFD ที่มากกว่าแสดงถึงความเร็วเฉลี่ยในการค้นพบข้อผิดพลาดที่สูงกว่าค่าน้อย

2) วิธีการที่ใช้เปรียบเทียบ

นอกจากการเปรียบเทียบระหว่างวิธีการจัดลำดับกรณีทดสอบด้วยกันแล้ว ยังจำเป็นต้องมีวิธีการที่ใช้เปรียบเทียบ เพื่อใช้เปรียบเทียบกับกรณีอื่นๆ เช่น เปรียบเทียบกับชุดทดสอบที่ไม่ถูกจัดลำดับ หรือเปรียบเทียบกับลำดับที่ดีที่สุดที่เป็นไปได้ โดยงานวิจัยนี้ได้กำหนดวิธีการที่ใช้จัดลำดับชุดทดสอบเพื่อเปรียบเทียบ 2 แบบ ได้แก่

2.1) วิธีสุ่ม (random) คือการจัดลำดับกรณีทดสอบแบบสุ่ม เปรียบเสมือนกรณีที่กรณีทดสอบไม่ได้ถูกจัดลำดับมาก่อนเลย ถือเป็นค่าขั้นต่ำ ซึ่งเมื่อจัดลำดับความสำคัญแล้วควรจะมีประสิทธิภาพสูงกว่าวิธีการนี้ ในการทดลองเราอาศัยการคำนวณเกี่ยวกับโอกาสในการคำนวณความเป็นไปได้ที่กรณีทดสอบใดกรณีทดสอบหนึ่งจะถูกเลือกมาทดสอบ

2.2) วิธีออพติมอล (optimal) เป็นการจัดลำดับกรณีทดสอบหลังจากการทดสอบ โดยใช้ผลการทดสอบที่ได้ในการจัดลำดับ ถือเป็นลำดับที่ดีที่สุดที่เป็นไปได้ ทำให้ค่าประสิทธิภาพที่ได้เป็นค่าประสิทธิภาพสูงสุดที่เป็นไปได้สำหรับชุดทดสอบชุดนั้นๆ

3) การคำนวณโอกาส

การเลือกกรณีทดสอบสำหรับวิธีการจัดลำดับกรณีทดสอบแบบสุ่ม หรือในกรณีที่กรณีทดสอบมากกว่า 1 กรณีมีค่าความสำคัญเท่ากัน จำเป็นต้องใช้ค่าเฉลี่ยในการคำนวณโอกาสที่กรณีทดสอบจะถูกเลือก เนื่องจากกรณีทดสอบแต่ละกรณีมีโอกาสถูกเลือกเท่าเทียมกัน สำหรับการคำนวณค่าเฉลี่ยของจำนวนครั้งในการเลือกจนได้กรณีทดสอบที่ต้องการ สามารถเขียนในลักษณะของปัญหาต่อไปนี้

ปัญหา “มีของที่ต้องการอยู่ N ชิ้น จากของทั้งหมด S ชิ้น หากเลือกของขึ้นมาทีละชิ้น โดยไม่ใส่กลับคืน จำนวนครั้งเฉลี่ยในการเลือกของจนกว่าจะได้ของที่ต้องการเป็นเท่าไหร่”

สำหรับการแก้ปัญหาดังกล่าวและรายละเอียดการคำนวณแสดงในภาคผนวก ง สำหรับการนำมาใช้ในการคำนวณโอกาสที่กรณีทดสอบจะถูกเลือกสามารถคำนวณได้ดังนี้

กำหนดให้ S แทนจำนวนกรณีทดสอบที่มีโอกาสถูกเลือกเท่ากัน

N แทนจำนวนกรณีทดสอบที่ต้องการ

L แทนจำนวนกรณีทดสอบที่ถูกเลือกก่อนหน้า

C แทนค่าเฉลี่ยจำนวนครั้งการเลือกที่จะเลือกได้กรณีทดสอบที่ต้องการ

$$C = L + \frac{S + 1}{N + 1}$$

สำหรับตัวอย่างในการนำไปใช้งาน ได้แก่ ในการจัดลำดับแบบสุ่มมีกรณีทดสอบ 15 กรณี มีกรณีทดสอบที่ค้นพบข้อผิดพลาด 1 กรณี ลำดับเฉลี่ยที่กรณีทดสอบนี้จะถูกเลือก คือ $0+(15+1)/(1+1)=8$ อีกตัวอย่างหนึ่ง เช่น ถ้ามีกรณีทดสอบที่จัดลำดับความสำคัญไปแล้ว 3 กรณี อีก 5 กรณีมีค่าความสำคัญเท่ากัน ในจำนวนนี้มีกรณีทดสอบที่พบข้อผิดพลาด A อยู่ 2 กรณี ลำดับเฉลี่ยที่กรณีทดสอบที่ค้นพบข้อผิดพลาด A จะถูกเลือก คือ $3+(5+1)/(2+1)=5$ เป็นต้น ซึ่งค่าที่ได้จากการคำนวณเหล่านี้จะถูกนำไปใช้ในการคำนวณ APFD ต่อไป

5.3.2 การประเมินผลการทดสอบเปรียบเทียบระยะเวลาการทำงาน

การเปรียบเทียบระยะเวลาการทำงานจะใช้ระยะเวลาจริงในการจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการต่างๆ มีหน่วยเป็นมิลลิวินาที (millisecond) ซึ่งเป็นระยะเวลาที่ใช้ในการจัดลำดับความสำคัญชุดทดสอบที่มีลักษณะตามที่กำหนดไว้ ซึ่งลักษณะดังกล่าวแบ่งออกเป็น

1) จำนวนกรณีทดสอบ คือ จำนวนกรณีทดสอบทั้งหมดภายในชุดทดสอบ

2) ขนาดของกรณีทดสอบ คือ จำนวนเหตุการณ์หรือภารกิจที่รวมกันเข้าเป็นกรณีทดสอบหนึ่งๆ

3) ปริมาณของเหตุการณ์หรือภารกิจที่เป็นไปได้ในชุดทดสอบ คือ เหตุการณ์หรือภารกิจทั้งหมดที่เกิดขึ้นในชุดทดสอบ โดยแต่ละเหตุการณ์หรือภารกิจมีโอกาสในการเกิดเท่าๆกัน

เนื่องจากชุดทดสอบที่ใช้ในการทดสอบเกิดจากการสร้างขึ้น โดยไม่อิงกับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จริง จึงสามารถกำหนดลักษณะเหล่านี้ได้อย่างอิสระ สำหรับชุดทดสอบจะแบ่งเป็นส่วนของวิธีการแบบเบ็ดเสร็จและวิธีการแบบเพิ่มอย่างชัดเจน เนื่องจากวิธีการแบบเบ็ดเสร็จจะมีระยะเวลาการทำงานที่รวดเร็วกว่าวิธีการแบบเพิ่มมาก

5.4 รายละเอียดการทดสอบ

สำหรับรายละเอียดการทดสอบแบ่งออกเป็น 2 ส่วน ได้แก่ สภาพแวดล้อมในการทดสอบ ซึ่งใช้ในการทดสอบทั้ง 2 ส่วน และโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้ทดสอบในการทดสอบเปรียบเทียบประสิทธิภาพ ส่วนที่ 1

5.4.1 สภาพแวดล้อมในการทดสอบ

ฮาร์ดแวร์ที่ใช้ในการทดสอบการทำงานของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ ประกอบด้วย

- เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลเพนเทียมโฟร์ 3.0 กิกะเฮิรตซ์ (Pentium4 3.0 GHz.)
- หน่วยความจำหลัก (RAM) 1.0 กิกะไบต์ (1 GB)
- ฮาร์ดดิสก์ (Harddisk)160 กิกะไบต์ (160 GB)

ซอฟต์แวร์ที่ใช้ในการทดสอบการทำงานของเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ ประกอบด้วย

- ระบบปฏิบัติการ ไมโครซอฟท์วินโดวส์เอ็กซ์พี โพรเฟชชันแนล (Microsoft Windows XP Professional)

5.4.2 โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้ทดสอบ

โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้ทดสอบในส่วนการทดสอบเปรียบเทียบประสิทธิภาพใช้หลักเกณฑ์ในการคัดเลือกโปรแกรมตัวอย่าง ดังนี้

- ทำงานโดยอาศัยส่วนต่อประสานกราฟิกกับผู้ใช้เป็นหลักในการติดต่อกับผู้ใช้

- มีหน้าที่และขั้นตอนการทำงานที่เข้าใจได้ง่าย เพื่อสะดวกในการสร้างกรณีทดสอบ
- มีข้อผิดพลาดในส่วนต่อประสานกราฟิกกับผู้ใช้
- ทำงานบนระบบที่ใช้ในการทดสอบได้
- ส่วนต่อประสานกราฟิกกับผู้ใช้แบบมาตรฐานที่ใช้กันทั่วไป
- ได้รับอนุญาตให้ใช้งานได้ โดยไม่มีค่าใช้จ่ายในการใช้งาน

จากหลักเกณฑ์ดังกล่าว เราได้คัดเลือกโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้จำนวนทั้งสิ้น 10 โปรแกรม ซึ่งโปรแกรมเหล่านี้แต่ละโปรแกรมมีหน้าที่และลักษณะการทำงานที่แตกต่างกันไป พัฒนาขึ้นโดยกลุ่มนักพัฒนาคนละกลุ่ม และส่วนใหญ่พัฒนาโดยใช้ภาษาโปรแกรมที่แตกต่างกัน เพื่อให้สามารถใช้เป็นตัวแทนของโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ทั่วไปได้ โปรแกรมส่วนใหญ่มาจากเว็บไซต์ที่ให้บริการดาวน์โหลดโปรแกรมต่างๆบนอินเทอร์เน็ต สำหรับรายละเอียดของโปรแกรมต่างๆที่ใช้ทดสอบแสดงในตาราง 5.1

โปรแกรมเหล่านี้จะถูกทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ด้วยชุดทดสอบที่สร้างขึ้นโดยผู้ทดสอบ เพื่อค้นหาข้อผิดพลาดในส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งอาจทดสอบทั้งโปรแกรมหรือทดสอบเพียงบางส่วนขึ้นกับลักษณะของแต่ละโปรแกรม โดยชุดทดสอบที่สร้างขึ้นนี้ประกอบด้วยจำนวนกรณีทดสอบระหว่าง 50 – 100 กรณีทดสอบ

และเนื่องจากโปรแกรมส่วนใหญ่ไม่มีเอกสารการออกแบบ ดังนั้นการระบุข้อผิดพลาดในการทดสอบจึงยึดตามลักษณะการทำงานและคู่มือการใช้โปรแกรมที่ประกอบมาด้วยเป็นหลัก โดยใช้การตัดสินใจของผู้ใช้ (ในกรณีนี้คือ ผู้ทดสอบ) เป็นหลัก ดังนั้นการทดสอบโปรแกรมทั้งหมดจึงจำเป็นต้องกระทำโดยมนุษย์

ข้อมูลรายละเอียดของจำนวนกรณีทดสอบ ข้อผิดพลาดที่พบ และจำนวนกรณีทดสอบที่ค้นพบข้อผิดพลาดของแต่ละโปรแกรมแสดงในตาราง 5.2 และสำหรับข้อมูลโดยละเอียดของการทดสอบรายงานอยู่ในภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.1 รายละเอียดโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ใช้ทดสอบ

ลำดับที่	ชื่อโปรแกรม	รุ่น (Version)	หน้าที่การทำงาน	ภาษาโปรแกรม (Programming Language)	ผู้พัฒนา (Developers)	แหล่งที่มา (Source)
1	Naive	-	แก้ปัญหาที่กำหนดไว้โดยรับข้อมูลจากผู้ใช้	Visual Basic	Paul C. Jorgensen	หนังสือ "Software testing : A craftsman's Approach" 2nd Edition by Paul C. Jorgensen Download : http://www.crcpress.com
2	Object Counter	-	นับจำนวน Object และ LOC (line of code) ของ Source Code ภาษา Java	Java	สมชาย วงศ์รัตนรักษ์	Assignment 3 วิชา Software Methodology
3	CImage Demo	1.4.1	แก้ไขไฟล์รูปภาพ	Visual C++	Troels Knakkergaard	Download : http://www.codeproject.com
4	Grade (โปรแกรมคำนวณเกรดเฉลี่ย)	1.2c	คำนวณค่า GPA , GPS	-	อลงกร ยืนชีวิต	Download : http://www.thaiware.com
5	Easy Info	1.0	เก็บข้อมูลคนไข้ในคลินิก	-	Sunday Soft	Download : http://www.thaiware.com
6	โปรแกรมร้านเช่าหนังสือ	1.52	บริหารการให้เช่า ยืมหนังสือ	-	ชาญชัย อออุดมยุทธ (Lisoft)	Download : http://www.thaiware.com
7	Address Book Standard Edition	2.0	บันทึกที่อยู่ เบอร์โทรศัพท์ ข้อมูลส่วนตัวของบุคคลต่างๆ	-	Jon Blankenhorn (JB ENTERPRISES)	Download : http://www.download.com
8	Small Math 2D	0.1.3	สร้างกราฟจากสมการรูปแบบต่างๆ	Java	Small Math Dev	Download : http://www.download.com
9	Satang (สตางค์)	2.2 build 71	จัดการบัญชีรายรับ-รายจ่ายส่วนบุคคล	Visual Basic	สราวุธ จันทร์โชติเสถียร	Download : http://www.thaiware.com
10	F-CRC Calculator	0.6	คำนวณค่า CRC และ MD5 ของแฟ้มข้อมูล	-	Fulgore	Download : http://www.download.com

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.2 ข้อมูลผลการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมตัวอย่าง

ลำดับที่	ชื่อโปรแกรม	จำนวนกรณีทดสอบ	จำนวนข้อผิดพลาดที่พบ	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
1	Naive	63	3	27
2	Object Counter	54	2	14
3	CImage Demo	51	2	8
4	Grade	53	3	6
5	Easy Info	55	4	9
6	โปรแกรมร้านเช่าหนังสือ	100	10	15
7	Address Book Standard Edition	52	5	9
8	Small Math 2D	66	2	3
9	Satang	87	6	10
10	F-CRC Calculator	51	4	18

5.5 ผลการทดสอบ

ผลการทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ทั้ง 2 ส่วน สามารถอธิบายได้ดังนี้

5.5.1 ผลการทดสอบเปรียบเทียบประสิทธิภาพ

จากการทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการค้นพบข้อผิดพลาดของวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ โดยใช้เมตริกซ์ APFD เป็นหลักในวัดประสิทธิภาพ ได้ผลการทดสอบดังแสดงในตาราง 5.3 โดยวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มทั้งหมดใช้ค่าเฉลี่ยจากการทดสอบ 100 ครั้ง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.3 ค่า APFD ชุดทดสอบของโปรแกรมต่างๆที่ผ่านการจัดลำดับความสำคัญกรณีทดสอบ

ลำดับที่	โปรแกรม	Optimal	Random	Total				Additional			
				Event		Task		Event		Task	
				Single	Pair	Single	Pair	Single	Pair	Single	Pair
1	Naive	0.986772	0.904535	0.745503	0.745503	0.730904	0.730904	0.941767	0.950771	0.937986	0.929346
2	Object Counter	0.981481	0.718254	0.527778	0.537037	0.604938	0.594136	0.912398	0.907407	0.910645	0.897288
3	CImage Demo	0.980392	0.797386	0.307190	0.405229	0.513072	0.557190	0.750000	0.753922	0.732843	0.671569
4	Grade	0.971698	0.601887	0.686583	0.686583	0.596436	0.596436	0.694780	0.649204	0.694748	0.671688
5	Easy Info	0.963636	0.627273	0.461364	0.461364	0.465909	0.465909	0.679500	0.590826	0.620277	0.510792
6	โปรแกรมร้านเช่าหนังสือ	0.950000	0.584167	0.575000	0.567167	0.500667	0.498500	0.711112	0.746313	0.706263	0.745075
7	Address Book Standard Edition	0.978846	0.663077	0.680769	0.666667	0.625320	0.629167	0.695448	0.818269	0.789712	0.850577
8	Small Math 2D	0.984848	0.584596	0.694444	0.665404	0.363636	0.363636	0.856301	0.854040	0.698232	0.553662
9	Satang	0.975096	0.626437	0.784962	0.784962	0.820402	0.820402	0.668466	0.696917	0.753378	0.750026
10	F-CRC Calculator	0.975490	0.750654	0.588235	0.598039	0.572304	0.579657	0.834961	0.903276	0.853638	0.813873
	ค่าเฉลี่ย	0.974826	0.685826	0.605183	0.611796	0.579359	0.583594	0.774473	0.787095	0.769772	0.739389

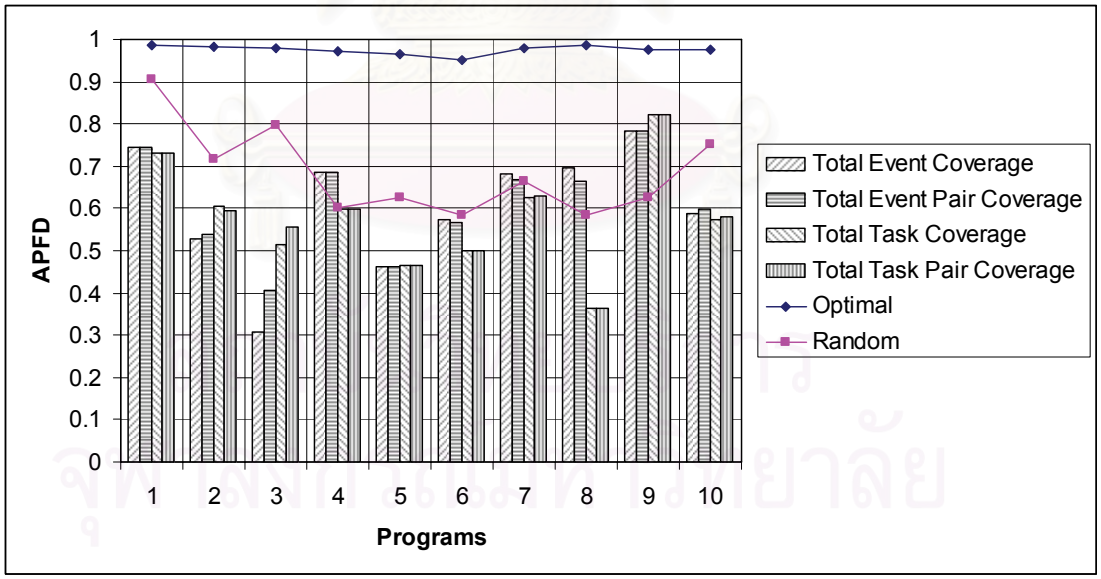
หมายเหตุ ค่า APFD ที่ขีดเส้นใต้ คือ ค่า APFD สูงสุดของโปรแกรมนั้นๆ โดยเปรียบเทียบเฉพาะวิธีการจัดลำดับความสำคัญที่อธิบายไว้ในบทที่ 3 (ไม่รวมวิธีออพติมอลและวิธีสุ่ม)

เนื่องจากวิธีการจัดลำดับความสำคัญกรณีทดสอบที่ทดสอบมีจำนวนมาก เพื่อให้สะดวกต่อการทำความเข้าใจ จึงแบ่งวิธีการเหล่านี้ออกเป็น 2 ประเภท คือ วิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จและวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม

1) ผลการทดสอบประสิทธิภาพวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ

เมื่อพิจารณาชุดทดสอบที่ผ่านการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จพบว่า กรณีทดสอบที่ประกอบด้วยเหตุการณ์หรือภารกิจจำนวนมากมีแนวโน้มที่จะมีความสำคัญในระดับสูง และทุกกรณีทดสอบที่ผ่านการจัดลำดับความสำคัญมีระดับความสำคัญมากกว่า 0 ทั้งนี้เนื่องจากลักษณะเฉพาะของการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จที่พิจารณาเฉพาะกรณีทดสอบเป็นกรณีๆไป

จากเหตุผลดังกล่าวทำให้ประสิทธิภาพในการค้นพบข้อผิดพลาดของวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จไม่สูงนัก กล่าวคือ ส่วนใหญ่มีประสิทธิภาพใกล้เคียงหรือต่ำกว่าวิธีการจัดลำดับกรณีทดสอบแบบสุ่ม โดยแผนภูมิแสดงค่า APFD ของชุดทดสอบโปรแกรมที่ใช้การจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จแสดงในรูป 5.1



รูปที่ 5.1 แผนภูมิแสดงค่า APFD ของชุดทดสอบโปรแกรม เมื่อจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการแบบเบ็ดเสร็จ

จากข้อมูลแผนภูมิในรูป 5.1 พบว่าส่วนใหญ่ค่า APFD ของวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จจะมีค่าใกล้เคียงหรือต่ำกว่าวิธีการจัดลำดับกรณี

ทดสอบแบบสุ่ม ยกเว้นในกรณีของโปรแกรมหมายเลข 9 คือ โปรแกรมสตางค์ ทั้งนี้เพราะโปรแกรมสตางค์มีข้อผิดพลาดส่วนใหญ่อยู่ในส่วนการแสดงผลกราฟ ซึ่งมีขั้นตอนการทำงานที่ยาวนานกว่าการทำงานทั่วไป ทำให้กรณีทดสอบที่เกี่ยวกับการแสดงผลกราฟประกอบด้วยเหตุการณ์หรือภารกิจมากกว่ากรณีทดสอบอื่นๆในชุดทดสอบ เมื่อใช้การจัดลำดับความสำคัญแบบเบ็ดเสร็จจึงให้ประสิทธิภาพสูง สำหรับรายละเอียดข้อผิดพลาดโปรแกรมแสดงในภาคผนวก ค

2) ผลการทดสอบประสิทธิภาพวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม

ชุดทดสอบที่ผ่านการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มมีลักษณะต่างจากการจัดลำดับความสำคัญแบบเบ็ดเสร็จ คือ กรณีทดสอบจะมีความสำคัญไม่เป็นไปตามขนาดของกรณีทดสอบ อาจมีข้อยกเว้นบ้างใน 2 - 3 กรณีทดสอบแรก และจะมีกรณีทดสอบที่มีระดับความสำคัญเป็น 0 อยู่จำนวนหนึ่ง ซึ่งจะมากน้อยตามเกณฑ์ความครอบคลุมของวิธีการที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบ โดยเกณฑ์ความครอบคลุมที่ละเอียดมากกว่าจะมีกรณีทดสอบที่มีระดับความสำคัญเป็น 0 อยู่น้อยกว่าเกณฑ์ความครอบคลุมที่ละเอียดน้อย

สาเหตุที่เป็นเช่นนี้เพราะการพิจารณาความครอบคลุมในระดับชุดทดสอบแม้ว่าในช่วงแรกๆกรณีทดสอบที่ประกอบด้วยเหตุการณ์หรือภารกิจมากกว่าจะมีโอกาสถูกเลือกมาก แต่เมื่อจัดลำดับความสำคัญต่อไปกรณีทดสอบที่มีลักษณะการทำงานต่างออกไปจะถูกเลือกโดยไม่คำนึงถึงขนาด แต่พิจารณาที่ความแตกต่างในเชิงความครอบคลุมมากกว่า และท้ายสุดจะมีกรณีทดสอบที่ไม่มีความสำคัญหรือก็คือไม่มีองค์ประกอบ (เหตุการณ์ ความสัมพันธ์ระหว่างเหตุการณ์ ภารกิจ หรือความสัมพันธ์ระหว่างภารกิจ) ใหม่ๆ ในการพิจารณาตามเกณฑ์ความครอบคลุมนั้นๆ

สำหรับประสิทธิภาพในการค้นพบข้อผิดพลาดของวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มส่วนใหญ่จะสูงกว่าการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จและการจัดลำดับกรณีทดสอบแบบสุ่ม แผนภูมิแสดงค่า APFD ของชุดทดสอบโปรแกรมที่ใช้การจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มแสดงในรูป 5.2 สำหรับความสัมพันธ์ระหว่างจำนวนกรณีทดสอบที่มีระดับความสำคัญ (ระดับความสำคัญมากกว่า 0) กับข้อผิดพลาดที่พบแสดงในตาราง 5.4 และ 5.5 ซึ่งข้อมูลทั้งหมดมาจากค่าเฉลี่ยของการทดลอง 100 ครั้ง

ตารางที่ 5.4 จำนวนกรณีทดสอบที่มีระดับความสำคัญกับข้อผิดพลาดที่พบ สำหรับการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม

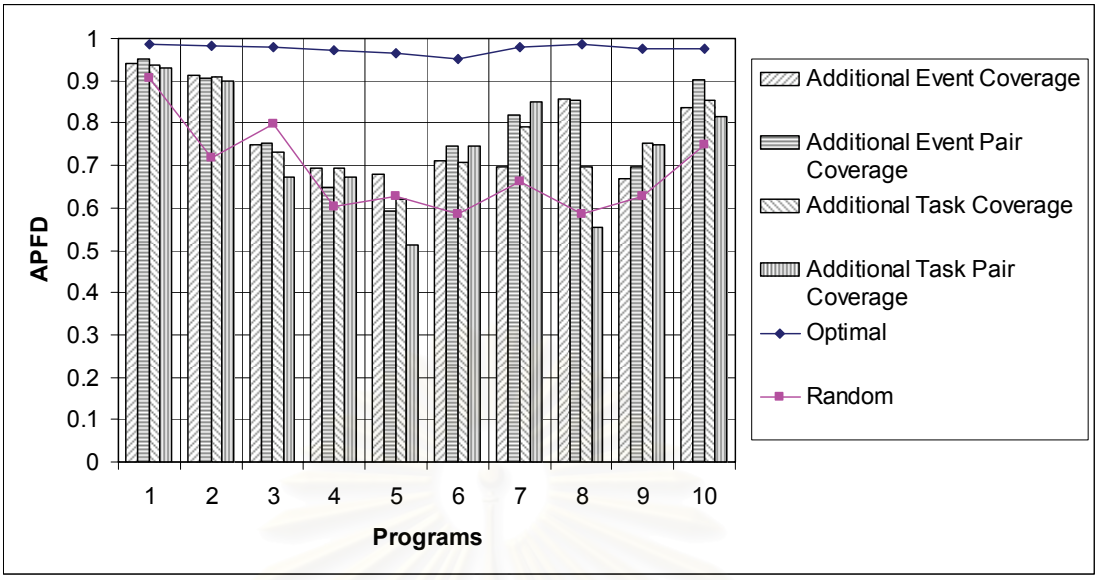
ลำดับที่	ชื่อโปรแกรม	จำนวนกรณีทดสอบทั้งหมด	จำนวนข้อผิดพลาดทั้งหมด	Event				Task			
				Single		Pair		Single		Pair	
				P>0	EF	P>0	EF	P>0	EF	P>0	EF
1	Naive	63	3	16	2.92	23	3	18	2.93	33	3
2	Object Counter	54	2	8	2	25	2	9	2	30	2
3	CImage Demo	51	2	14	1	28	2	15	1	38	2
4	Grade	53	3	33.56	3	48	3	34.44	3	48	3
5	Easy Info	55	4	35	3.91	48	3.91	45.03	3.9	53.01	4
6	โปรแกรมร้านเช่าหนังสือ	100	10	59.14	9.59	75.69	10	68.67	9.47	82.31	10
7	Address Book Standard Edition	52	5	23.81	3	47	5	40.63	4.76	50.37	5
8	Small Math 2D	66	2	20.42	2	40	2	41	2	58.3	2
9	Satang	87	6	35.54	4.68	52.12	5.64	43.67	5.67	72.64	6
10	F-CRC Calculator	51	4	15.54	3.37	35	3.95	15.54	3.37	37	3.91

หมายเหตุ P>0 แทนจำนวนกรณีทดสอบที่มีค่าความสำคัญมากกว่า 0 และ EF แทนจำนวนข้อผิดพลาดที่พบ

ตารางที่ 5.5 เปอร์เซ็นต์ของกรณีทดสอบที่มีระดับความสำคัญกับข้อผิดพลาดที่พบ สำหรับการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม

ลำดับที่	ชื่อโปรแกรม	Event				Task			
		Single		Pair		Single		Pair	
		P>0	EF	P>0	EF	P>0	EF	P>0	EF
1	Naive	25.40%	97.33%	36.51%	100.00%	28.57%	97.67%	52.38%	100.00%
2	Object Counter	14.81%	100.00%	46.30%	100.00%	16.67%	100.00%	55.56%	100.00%
3	CImage Demo	27.45%	50.00%	54.90%	100.00%	29.41%	50.00%	74.51%	100.00%
4	Grade	63.32%	100.00%	90.57%	100.00%	64.98%	100.00%	90.57%	100.00%
5	Easy Info	63.64%	97.75%	87.27%	97.75%	81.87%	97.50%	96.38%	100.00%
6	โปรแกรมร้านเช่าหนังสือ	59.14%	95.90%	75.69%	100.00%	68.67%	94.70%	82.31%	100.00%
7	Address Book Standard Edition	45.79%	60.00%	90.38%	100.00%	78.13%	95.20%	96.87%	100.00%
8	Small Math 2D	30.94%	100.00%	60.61%	100.00%	62.12%	100.00%	88.33%	100.00%
9	Satang	40.85%	78.00%	59.91%	94.00%	50.20%	94.50%	83.49%	100.00%
10	F-CRC Calculator	30.47%	84.25%	68.63%	98.75%	30.47%	84.25%	72.55%	97.75%
ค่าเฉลี่ย		40.18%	**86.32%**	67.08%	**99.05%**	51.11%	**91.38%**	79.29%	**99.78%**

หมายเหตุ P>0 แทนเปอร์เซ็นต์ของกรณีทดสอบในชุดทดสอบที่มีค่าความสำคัญมากกว่า 0 และ EF แทนเปอร์เซ็นต์ข้อผิดพลาดที่พบจากข้อผิดพลาดทั้งหมด



รูปที่ 5.2 แผนภูมิแสดงค่า APFD ของชุดทดสอบโปรแกรม เมื่อจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการแบบเพิ่ม

หากพิจารณาจากตาราง 5.4 เราจะพบว่าการจัดลำดับความสำคัญกรณีทดสอบโดยใช้วิธีการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่มกำหนดระดับความสำคัญให้กรณีทดสอบเฉลี่ย 40.18 % จากกรณีทดสอบทั้งหมดและกรณีทดสอบเหล่านี้สามารถค้นพบข้อผิดพลาดเฉลี่ย 86.32 % จากข้อผิดพลาดทั้งหมด สำหรับวิธีการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ ภารกิจ และความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม กำหนดระดับความสำคัญให้กรณีทดสอบเฉลี่ย 67.08 %, 51.11% และ 79.29 % ตามลำดับ โดยกรณีทดสอบเหล่านี้สามารถค้นพบข้อผิดพลาดเฉลี่ย 99.05 %, 91.38 % และ 99.78 % ตามลำดับ

จากข้อมูลดังกล่าวแสดงให้เห็นว่าการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมที่กำหนดสามารถลดจำนวนกรณีทดสอบลง โดยส่งผลกระทบถึงความสามารถในการค้นพบข้อผิดพลาดของชุดทดสอบน้อยมาก และเกณฑ์ความครอบคลุมที่มีความละเอียดมากขึ้นจะส่งผลให้ความสามารถในการค้นพบข้อผิดพลาดดีขึ้น

สำหรับประสิทธิภาพหรือความเร็วในการค้นพบข้อผิดพลาดของชุดทดสอบซึ่งแสดงในรูป 5.2 พบว่าเมื่อพิจารณาร่วมกับตาราง 5.4 สามารถแบ่งออกเป็นรูปแบบได้ดังนี้

รูปแบบที่ 1 วิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มทั้งหมดให้ผลลัพธ์ใกล้เคียงกัน เช่น ตัวอย่างโปรแกรมที่ 1 และ 2 มักพบในโปรแกรมที่ไม่ซับซ้อน

ประกอบด้วยภารกิจสั้นๆที่มีเหตุการณ์ไม่มากนัก จึงทำให้เหตุการณ์และภารกิจมีลักษณะใกล้เคียงกันมาก นอกจากนี้ยังประกอบด้วยเหตุการณ์จำนวนไม่มากนักทำให้การจัดลำดับกรณีทดสอบทั้งระดับปกติและความสัมพันธ์ให้ผลใกล้เคียงกัน

รูปแบบที่ 2 วิธีการจัดลำดับความสำคัญที่ใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์และความสัมพันธ์ระหว่างเหตุการณ์ให้ผลลัพธ์ดีกว่า เช่น ตัวอย่างโปรแกรมที่ 3 5 และ 8 ส่วนใหญ่เป็นโปรแกรมที่ข้อผิดพลาดมีลักษณะไม่ซับซ้อนและอยู่ในระดับการทำงานแบบเหตุการณ์ ไม่วิธีการจัดลำดับความสำคัญกรณีทดสอบแบบใดก็สามารถค้นพบข้อผิดพลาดได้ ดังนั้นวิธีการที่กำหนดความสำคัญให้กรณีทดสอบจำนวนน้อยกว่า จึงให้ประสิทธิภาพที่ดีกว่า

รูปแบบที่ 3 วิธีการจัดลำดับความสำคัญที่ใช้เกณฑ์ความครอบคลุมระดับภารกิจและความสัมพันธ์ระหว่างภารกิจให้ผลลัพธ์ดีกว่า เช่น ตัวอย่างโปรแกรมที่ 7 และ 9 ส่วนใหญ่เป็นโปรแกรมที่มีข้อผิดพลาดระดับการทำงานร่วมกัน เกณฑ์ความครอบคลุมระดับที่ละเอียดน้อยกว่าจึงไม่สามารถค้นพบข้อผิดพลาดดังกล่าวได้

รูปแบบที่ 4 วิธีการจัดลำดับความสำคัญที่ใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์และความสัมพันธ์ระหว่างภารกิจให้ผลลัพธ์ดีกว่า เช่น ตัวอย่างโปรแกรมที่ 6 7 และ 10 มีลักษณะเช่นเดียวกับรูปแบบที่ 3 แต่ลักษณะของโปรแกรมส่วนใหญ่จะมีภารกิจสั้นๆที่ใกล้เคียงกับเหตุการณ์ ทำให้ผลลัพธ์จากเกณฑ์ความครอบคลุมในลักษณะของความสัมพันธ์ให้ผลลัพธ์ดีกว่า

รูปแบบที่ 5 วิธีการจัดลำดับความสำคัญที่ใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์และภารกิจให้ผลลัพธ์ดีกว่า เช่น ตัวอย่างโปรแกรมที่ 4 และ 5 มีลักษณะเช่นเดียวกับรูปแบบที่ 2 แต่ในกรณีนี้วิธีการที่ใช้เกณฑ์ความครอบคลุมระดับภารกิจเป็นวิธีที่กำหนดระดับความสำคัญให้กรณีทดสอบน้อยกว่าเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์

สำหรับกรณีของตัวอย่างโปรแกรมที่ 3 และ 5 ที่ชุดทดสอบที่ผ่านการจัดลำดับความสำคัญกรณีทดสอบส่วนใหญ่มีประสิทธิภาพต่ำกว่าชุดทดสอบที่จัดลำดับแบบสุ่ม เนื่องจากตัวอย่างโปรแกรมที่ 3 ประกอบด้วยกรณีทดสอบที่พบข้อผิดพลาดเป็นจำนวนมาก จึงทำให้การจัดลำดับกรณีทดสอบแบบสุ่มได้เปรียบ และสำหรับตัวอย่างโปรแกรมที่ 5 ชุดทดสอบประกอบด้วยภารกิจและเหตุการณ์จำนวนมาก ทำให้กรณีทดสอบแต่ละกรณีแตกต่างกันมาก ทำให้ชุดทดสอบที่ผ่านการจัดลำดับกรณีทดสอบแล้วมีกรณีทดสอบที่ถูกกำหนดระดับความสำคัญเป็นจำนวนมาก ซึ่งทำให้ประสิทธิภาพของวิธีการจัดลำดับความสำคัญกรณีทดสอบลดลง

ตารางที่ 5.6 เปอร์เซ็นต์ของกรณีทดสอบและเหตุการณ์ที่เป็นองค์ประกอบของกรณีทดสอบที่มีระดับความสำคัญมากกว่า 0 เมื่อจัดลำดับความสำคัญกรณีทดสอบด้วยวิธีการแบบเพิ่ม

ลำดับที่	ชื่อโปรแกรม	จำนวนกรณีทดสอบ				จำนวนเหตุการณ์			
		Event		Task		Event		Task	
		Single	Pair	Single	Pair	Single	Pair	Single	Pair
1	Naive	25.40%	36.51%	28.57%	52.38%	17.23%	21.06%	19.33%	38.23%
2	Object Counter	14.81%	46.30%	16.67%	55.56%	14.28%	44.29%	17.18%	55.21%
3	CImage Demo	27.45%	54.90%	29.41%	74.51%	29.95%	52.20%	31.80%	72.64%
4	Grade	63.32%	90.57%	64.98%	90.57%	66.29%	91.77%	67.79%	91.76%
5	Easy Info	63.64%	87.27%	81.87%	96.38%	54.71%	84.38%	79.37%	97.09%
6	โปรแกรมร้านเช่าหนังสือ	59.14%	75.69%	68.67%	82.31%	53.38%	68.48%	62.33%	75.88%
7	Address Book Standard Edition	45.79%	90.38%	78.13%	96.87%	52.10%	91.94%	81.89%	94.91%
8	Small Math 2D	30.94%	60.61%	62.12%	88.33%	35.32%	57.49%	61.32%	88.53%
9	Satang	40.85%	59.91%	50.20%	83.49%	47.60%	63.02%	56.42%	86.94%
10	F-CRC Calculator	30.47%	68.63%	30.47%	72.55%	25.07%	60.00%	25.11%	64.44%
ค่าเฉลี่ย		40.18%	67.08%	51.11%	79.29%	39.59%	63.46%	50.25%	76.56%

เนื่องจากการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในงานวิจัยนี้เป็นการทดสอบโดยมนุษย์ ดังนั้นระยะเวลาที่ใช้ในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จึงไม่แน่นอน แต่หากตั้งสมมติฐานว่าในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้แต่ละเหตุการณ์ใช้ระยะเวลาใกล้เคียงกัน ดังนั้นเราสามารถประเมินระยะเวลาในการทดสอบได้จากจำนวนเหตุการณ์ที่ถูกทดสอบ

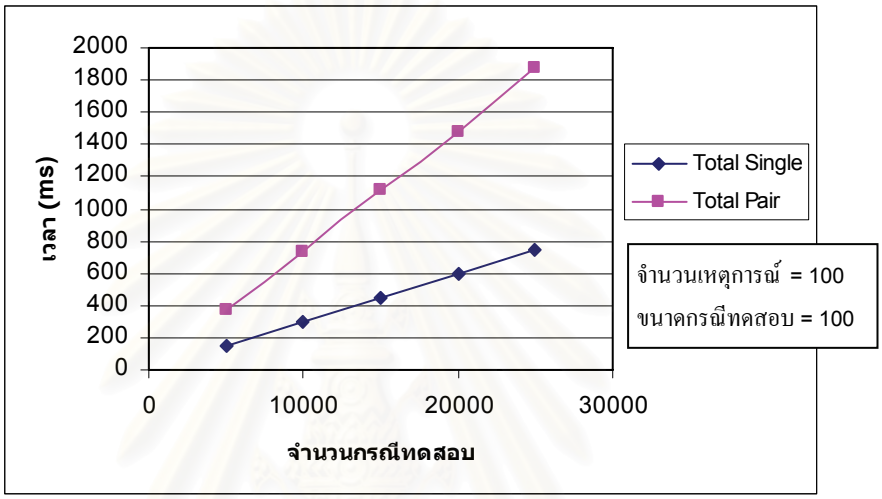
สำหรับการจัดลำดับความสำคัญของเหตุการณ์แบบเพิ่ม พบว่าจำนวนเหตุการณ์ที่ถูกทดสอบ เมื่อเทียบกับเหตุการณ์ทั้งหมดในชุดทดสอบใกล้เคียงกับจำนวนกรณีทดสอบที่ถูกทดสอบเทียบกับจำนวนกรณีทดสอบทั้งหมดในชุดทดสอบ ดังแสดงในตารางที่ 5.6 ดังนั้นระยะเวลาในการทดสอบสำหรับการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม เมื่อเทียบกับระยะเวลาที่ใช้ในการทดสอบทั้งหมด จึงมีอัตราส่วนใกล้เคียงกับจำนวนกรณีทดสอบที่ถูกทดสอบเมื่อเทียบกับจำนวนกรณีทดสอบทั้งหมดในชุดทดสอบ

5.5.2 ผลการทดสอบเปรียบเทียบระยะเวลาการทำงาน

การทดสอบเปรียบเทียบระยะเวลาการทำงานของวิธีการจัดลำดับความสำคัญกรณีทดสอบวิธีการต่างๆ โดยใช้กรณีทดสอบที่สร้างขึ้นแบบสุ่ม โดยแบ่งการทดสอบเป็นวิธีการแบบเบ็ดเสร็จและวิธีการแบบเพิ่มอย่างชัดเจนเนื่องจากวิธีการแบบเบ็ดเสร็จจะใช้ระยะเวลาการทำงานรวดเร็วกว่าวิธีการแบบเพิ่มมาก จากนั้นจึงหาระยะเวลาเฉลี่ยจากการทดสอบ 10 ครั้ง โดยการทดสอบแบ่งออกเป็นส่วนต่างๆตามปัจจัยหลักที่ประเมิน ดังนี้

1) จำนวนกรณีทดสอบ

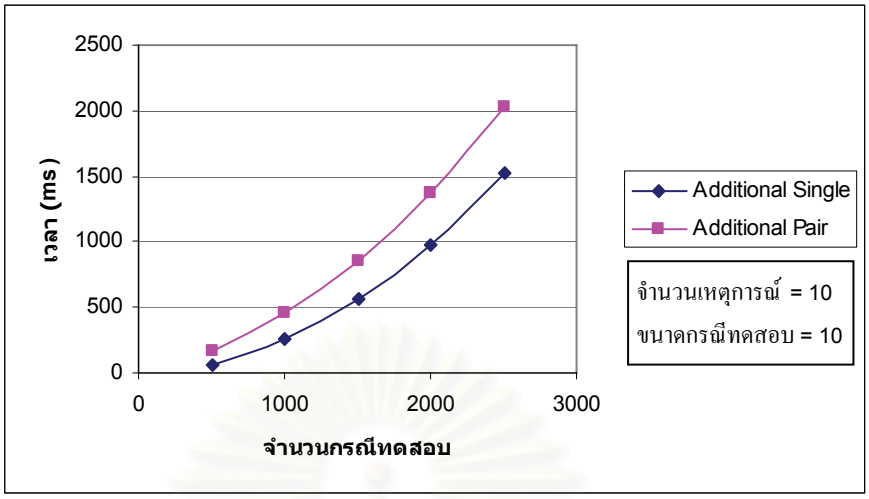
ถ้ากำหนดให้กรณีทดสอบมีขนาดคงที่เท่ากันหมดและจำนวนเหตุการณ์หรือภารกิจที่เป็นไปได้ทั้งหมดภายในชุดทดสอบคงที่ เมื่อเปลี่ยนแปลงจำนวนกรณีทดสอบภายในชุดทดสอบระยะเวลาที่ใช้ในการจัดลำดับความสำคัญของชุดทดสอบมีการเปลี่ยนแปลงตามกราฟในรูป 5.3 และ 5.4



รูปที่ 5.3 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มจำนวนกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ

จากรูป 5.3 พบว่าวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จทั้งหมดเมื่อกรณีทดสอบเพิ่มขึ้น ระยะเวลาที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบจะเพิ่มขึ้นในลักษณะของฟังก์ชันเชิงเส้น (linear function) โดยวิธีการจัดลำดับที่พิจารณาในลักษณะความสัมพันธ์จะใช้เวลามากกว่าวิธีการที่พิจารณาในระดับเหตุการณ์หรือภารกิจเดี่ยวๆ

สำหรับรูป 5.4 พบว่าวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มทั้งหมดเมื่อกรณีทดสอบเพิ่มขึ้น ระยะเวลาที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบจะเพิ่มขึ้นในลักษณะของฟังก์ชันกำลังสอง (quadratic function) ทั้งนี้เนื่องจากการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มในแต่ละลำดับต้องตรวจสอบกรณีทดสอบทั้งหมดที่ยังไม่ถูกจัดลำดับความสำคัญทุกครั้ง และวิธีการจัดลำดับที่พิจารณาในลักษณะความสัมพันธ์จะใช้เวลามากกว่าวิธีการที่พิจารณาในระดับเหตุการณ์หรือภารกิจเดี่ยวๆ

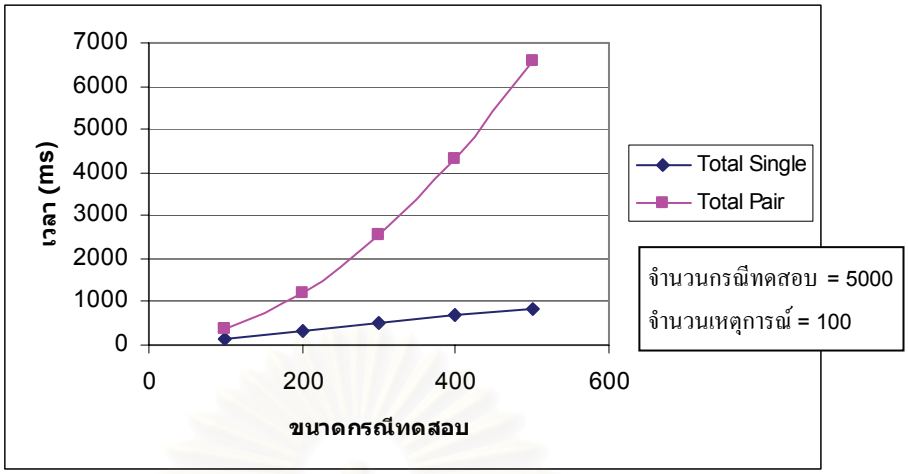


รูปที่ 5.4 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มจำนวนกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม

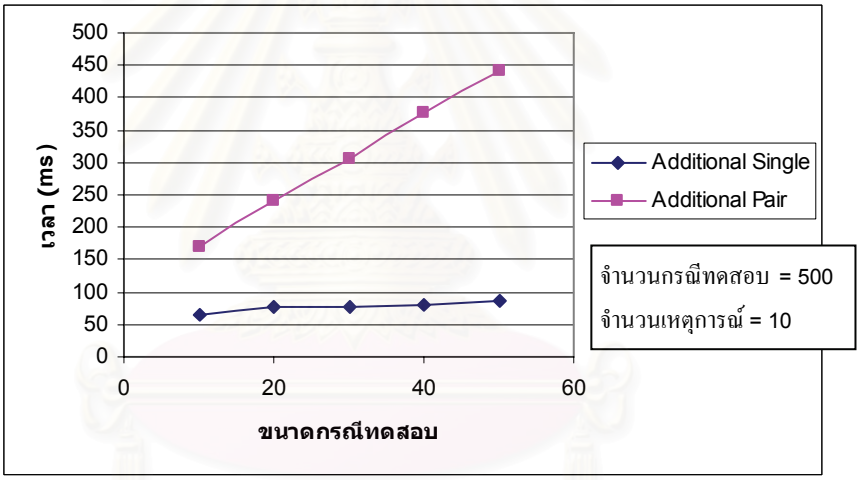
2) ขนาดของกรณีทดสอบ

ถ้ากำหนดให้กรณีทดสอบมีจำนวนคงที่และจำนวนเหตุการณ์หรือภารกิจที่เป็นไปได้ทั้งหมดภายในชุดทดสอบคงที่ เมื่อเปลี่ยนแปลงขนาดของกรณีทดสอบภายในชุดทดสอบโดยให้มีขนาดเท่ากันหมด ระยะเวลาที่ใช้ในการจัดลำดับความสำคัญของชุดทดสอบมีการเปลี่ยนแปลงตามกราฟในรูป 5.5 และ 5.6

สำหรับรูป 5.5 พบว่าวิธีการจัดลำดับกรณีทดสอบแบบเบ็ดเสร็จที่ใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์หรือภารกิจ ใช้ระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบเพิ่มขึ้นตามขนาดของกรณีทดสอบแบบเชิงเส้น แต่สำหรับวิธีการจัดลำดับความสำคัญที่ใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์หรือภารกิจ ระยะเวลาการจัดลำดับความสำคัญจะเพิ่มขึ้นในลักษณะของฟังก์ชันกำลังสอง ทั้งนี้เนื่องจากในการสร้างกรณีทดสอบแบบสุ่ม แต่ละเหตุการณ์หรือภารกิจจะมีความสัมพันธ์กันได้อย่างอิสระ สมมติว่ามีเหตุการณ์ทั้งหมด N เหตุการณ์ รูปแบบความสัมพันธ์ที่เป็นไปได้ทั้งหมดจะมี N^2 ความสัมพันธ์ ซึ่งปกติแล้วเมื่อพิจารณากรณีทดสอบๆหนึ่งจะประกอบด้วยความสัมพันธ์ระหว่างเหตุการณ์ไม่ถึง N^2 ความสัมพันธ์ ดังนั้นเมื่อเพิ่มขนาดกรณีทดสอบขึ้นจะทำให้ความสัมพันธ์ที่ไม่ซ้ำกันในกรณีทดสอบเพิ่มขึ้น ทำให้ต้องใช้เวลาในการเปรียบเทียบมากขึ้น ซึ่งแตกต่างจากการพิจารณาเหตุการณ์หรือภารกิจเดี่ยวๆซึ่งมีโอกาสซ้ำซ้อนกันบ้าง



รูปที่ 5.5 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มขนาดกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ

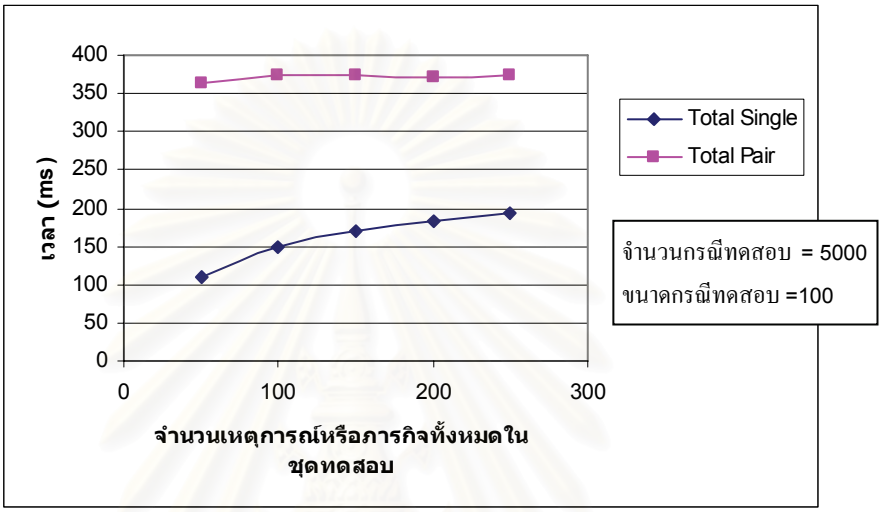


รูปที่ 5.6 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มขนาดกรณีทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่ม

จากรูป 5.6 พบว่าวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มทั้งหมดเมื่อกรณีทดสอบเพิ่มขึ้น ระยะเวลาที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบจะเพิ่มขึ้นแบบเชิงเส้น ทั้งนี้เนื่องจากการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มมีมุมมองในลักษณะรวมทั้งชุดทดสอบ ขนาดกรณีทดสอบที่เพิ่มขึ้นจึงไม่ส่งผลกระทบต่อระยะเวลาการทำงานมากนัก โดยวิธีการจัดลำดับที่พิจารณาในลักษณะความสัมพันธ์จะใช้เวลามากกว่าวิธีการที่พิจารณาในระดับเหตุการณ์หรือภารกิจเดี่ยวๆ

3) ปริมาณเหตุการณ์หรือภารกิจที่เป็นไปได้ทั้งหมดในชุดทดสอบ

ถ้ากำหนดให้กรณีทดสอบมีจำนวนและขนาดคงที่ เมื่อเปลี่ยนแปลงจำนวนเหตุการณ์หรือภารกิจที่เป็นไปได้ทั้งหมดภายในชุดทดสอบ ระยะเวลาที่ใช้ในการจัดลำดับความสำคัญของชุดทดสอบมีการเปลี่ยนแปลงตามกราฟในรูป 5.7 5.8 และ 5.9

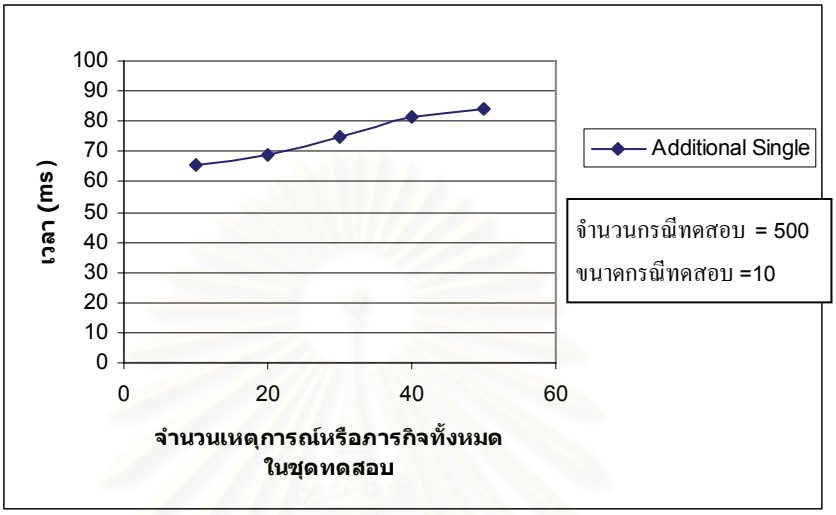


รูปที่ 5.7 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มปริมาณเหตุการณ์หรือภารกิจทั้งหมดในชุดทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จ

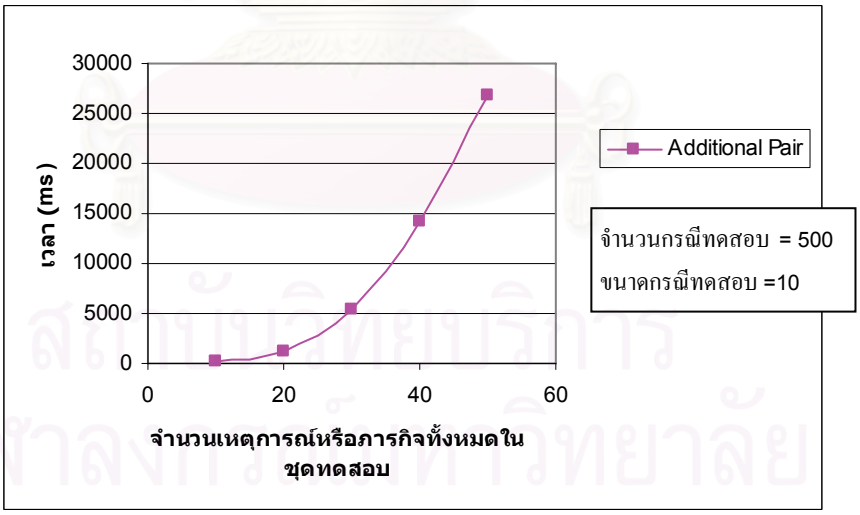
จากรูป 5.7 พบว่าแนวโน้มระยะเวลาที่ใช้จัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จทั้งหมดจะเข้าสู่ระดับอิ่มตัว ไม่มีการเปลี่ยนแปลงอีกต่อไป เมื่อมีการเพิ่มจำนวนเหตุการณ์หรือภารกิจทั้งหมดในชุดทดสอบไปถึงระดับหนึ่ง สาเหตุที่เป็นเช่นนี้เพราะการสร้างกรณีทดสอบแบบสุ่มนั้นเมื่อจำนวนเหตุการณ์หรือภารกิจทั้งหมดในชุดทดสอบมากขึ้นถึงระดับหนึ่งแล้ว มีโอกาสน้อยมากที่จะเกิดเหตุการณ์หรือภารกิจซ้ำในกรณีทดสอบหนึ่งๆ จึงทำให้ระยะเวลาที่ใช้จัดลำดับความสำคัญกรณีทดสอบคงที่

สำหรับรูป 5.8 พบว่าระยะเวลาการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมระดับเหตุการณ์หรือภารกิจเพิ่มขึ้นในลักษณะของฟังก์ชันเชิงเส้น เมื่อจำนวนเหตุการณ์หรือภารกิจเพิ่มขึ้น ทั้งนี้เพราะการจัดลำดับความสำคัญลักษณะนี้จะพิจารณาเหตุการณ์หรือภารกิจของชุดทดสอบทั้งหมด ดังนั้นเมื่อจำนวนเหตุการณ์หรือภารกิจที่เป็นไปได้ในชุดทดสอบเพิ่มขึ้นย่อมทำให้การทำงานช้าลง สำหรับรูป 5.9 ระยะเวลาการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์หรือภารกิจเพิ่มขึ้นใน

ลักษณะของฟังก์ชันกำลังสอง ทั้งนี้เนื่องจากสำหรับการสร้างกรณีทดสอบแบบสุ่ม รูปแบบความสัมพันธ์ที่เป็นไปได้ทั้งหมดจะเท่ากับจำนวนเหตุการณ์หรือภารกิจกำลังสอง



รูปที่ 5.8 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มปริมาณเหตุการณ์หรือภารกิจทั้งหมดในชุดทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมเชิงเดี่ยวแบบเพิ่ม



รูปที่ 5.9 กราฟแสดงความสัมพันธ์ระหว่างการเพิ่มปริมาณเหตุการณ์หรือภารกิจทั้งหมดในชุดทดสอบกับระยะเวลาในการจัดลำดับความสำคัญกรณีทดสอบตามเกณฑ์ความครอบคลุมเชิงความสัมพันธ์แบบเพิ่ม

อนึ่ง เนื่องจากเครื่องมือที่ใช้ในการทดสอบถูกสร้างขึ้นอย่างง่าย เพื่อยืนยันแนวคิดการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เท่านั้น จึงมิได้มีการคำนึงถึงประสิทธิภาพและความเร็วในการทำงาน โดยใช้วิธีการเปรียบเทียบเหตุการณ์ภารกิจ ความสัมพันธ์ระหว่างเหตุการณ์ และความสัมพันธ์ระหว่างภารกิจอย่างง่าย หากมีการใช้โครงสร้างข้อมูลรูปแบบต่างๆมาช่วยในการเปรียบเทียบเหล่านี้แล้ว น่าจะลดระยะเวลาในการทำงานของวิธีการจัดลำดับความสำคัญกรณีทดสอบได้อีกมาก

5.6 สรุปผลการทดสอบ

เมื่อพิจารณาผลการทดสอบทั้งสองส่วน พบว่าวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้แบบเบ็ดเสร็จ แม้ว่าจะทำงานได้อย่างรวดเร็ว และในบางกรณีให้ประสิทธิภาพในการค้นพบข้อผิดพลาดที่ดี แต่ส่วนใหญ่ให้ประสิทธิภาพในการค้นพบข้อผิดพลาดด้อยกว่าหรือใกล้เคียงกับการทดสอบกรณีทดสอบที่จัดลำดับแบบสุ่ม ดังนั้นจึงไม่เหมาะสมที่จะนำมาใช้ในการจัดลำดับความสำคัญกรณีทดสอบ

สำหรับวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้แบบเพิ่ม แม้จะทำงานได้ช้ากว่า แต่ส่วนใหญ่ให้ประสิทธิภาพในการค้นพบข้อผิดพลาดที่ดีกว่าการทดสอบกรณีทดสอบที่จัดลำดับแบบสุ่ม โดยจะมีแต่ละวิธีการจะเหมาะสมกับลักษณะโปรแกรมที่แตกต่างกัน เนื่องจากหากจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมที่ละเอียดจะมีจำนวนกรณีทดสอบที่มีระดับความสำคัญสูงอยู่มากหรือมีกลุ่มเป้าหมายในการทดสอบมาก ในขณะที่เกณฑ์ความครอบคลุมที่ละเอียดน้อยกว่าจะมีกลุ่มเป้าหมายในการทดสอบน้อยกว่า ซึ่งหากเกณฑ์ความครอบคลุมทั้งสองสามารถค้นพบข้อผิดพลาดได้ใกล้เคียงกันแล้ว เกณฑ์ความครอบคลุมที่ละเอียดน้อยกว่าจะพบข้อผิดพลาดได้รวดเร็วกว่า ดังนั้นจึงต้องเลือกเกณฑ์ความครอบคลุมให้เหมาะสมกับลักษณะข้อผิดพลาดที่พบ เพราะหากเลือกเกณฑ์ความครอบคลุมที่ละเอียดเกินไปจะทำให้พบข้อผิดพลาดช้า แต่หากเลือกเกณฑ์ความครอบคลุมที่ละเอียดน้อยเกินไปจะทำให้ไม่สามารถค้นพบข้อผิดพลาดได้อย่างครบถ้วน ซึ่งแนวทางในการเลือกเกณฑ์ความครอบคลุมที่หลักในการพิจารณาดังนี้

วิธีการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์ มีจุดเด่นในการจัดลำดับความสำคัญที่รวดเร็ว สามารถค้นพบข้อผิดพลาดที่ไม่ซับซ้อนได้ดี ไม่จำเป็นต้องกำหนดภารกิจก่อน แต่มีข้อเสียคือ ไม่สามารถค้นพบข้อผิดพลาดที่ซับซ้อนได้ เมื่อจัดลำดับความสำคัญแล้วมีกรณีทดสอบที่ไม่มีระดับความสำคัญเลยมาก อ่อนไหวต่อการ

เปลี่ยนแปลงของส่วนต่อประสานกราฟิกกับผู้ใช้ เหมาะสำหรับโปรแกรมที่มีการทำงานไม่ซับซ้อน โปรแกรมรุ่นแรกๆหรือรุ่นทดสอบที่มีข้อผิดพลาดอยู่มาก และโปรแกรมที่มีทรัพยากรในการทดสอบน้อย

วิธีการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์เหตุการณ์ มีจุดเด่นคือ ค้นหาข้อผิดพลาดได้ดี ไม่จำเป็นต้องกำหนดภารกิจก่อน แต่มีข้อเสียคือ อ่อนไหวต่อการเปลี่ยนแปลงของส่วนต่อประสานกราฟิกกับผู้ใช้ ไม่สามารถค้นพบข้อผิดพลาดที่เกิดจากการทำงานต่อเนื่องกันได้ดีนัก โดยเฉพาะในกรณีที่มีกรณีทดสอบลักษณะใกล้เคียงกันหลายกรณีทดสอบ และใช้เวลาในการจัดลำดับความสำคัญนาน ถ้าชุดทดสอบประกอบด้วยเหตุการณ์จำนวนมาก เหมาะสำหรับโปรแกรมที่ประกอบด้วยการทำงานสั้นๆ มีทางเลือกในการทำงานไม่มากนัก

วิธีการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับภารกิจ มีจุดเด่นคือ ค้นหาข้อผิดพลาดของการทำงานระดับภารกิจได้ดี สามารถใช้งานได้ในกรณีที่ส่วนต่อประสานกราฟิกกับผู้ใช้เปลี่ยนแปลงไม่มากนัก แต่มีข้อเสียคือ ต้องมีขั้นตอนการกำหนดภารกิจเพิ่มเติมในกรณีที่กรณีทดสอบเป็นระดับเหตุการณ์ ไม่สามารถค้นพบข้อผิดพลาดในการทำงานประสานกันระหว่างภารกิจได้ เมื่อจัดลำดับความสำคัญแล้วมีกรณีทดสอบที่ไม่มีระดับความสำคัญอยู่พอสมควร เหมาะสมสำหรับการทดสอบโปรแกรมที่ประกอบด้วยงานเป็นส่วนๆไม่เกี่ยวข้องกัน

วิธีการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ภารกิจ มีจุดเด่นคือ ค้นหาข้อผิดพลาดได้ดี สามารถใช้งานได้ในกรณีที่ส่วนต่อประสานกราฟิกกับผู้ใช้เปลี่ยนแปลงไม่มากนัก แต่มีข้อเสียคือ ต้องมีขั้นตอนการกำหนดภารกิจเพิ่มเติมในกรณีที่กรณีทดสอบเป็นระดับเหตุการณ์ มีกรณีทดสอบที่มีระดับความสำคัญจำนวนมาก จึงอาจส่งผลให้พบข้อผิดพลาดได้ช้าลง และใช้เวลาในการจัดลำดับความสำคัญนาน ถ้าชุดทดสอบประกอบด้วยภารกิจจำนวนมาก เหมาะสำหรับการทดสอบโปรแกรมที่ต้องการทดสอบค่อนข้างละเอียด โปรแกรมรุ่นที่ผ่านการทดสอบมาบ้างแล้ว และมีทรัพยากรในการทดสอบพอสมควร

เนื่องจากวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้แต่ละวิธีมีลักษณะเด่น ลักษณะด้อยเฉพาะตัว จึงไม่มีวิธีการที่ดีที่สุดสำหรับทุกๆสถานการณ์ แต่ขึ้นกับผู้ทดสอบที่จะเลือกใช้วิธีการที่เหมาะสมกับลักษณะส่วนต่อประสานกราฟิกกับผู้ใช้ที่ทำการทดสอบ สภาพแวดล้อม รวมถึงแนวทางการทดสอบ เพื่อให้ได้ประสิทธิภาพการทดสอบที่ดีที่สุด ภายใต้ทรัพยากรที่กำหนดไว้

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้เสนอแนวทางและวิธีการในการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งเป็นขั้นตอนหนึ่งในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้โดยมีเป้าหมายเพื่อลดปริมาณการทดสอบให้เหมาะสมกับทรัพยากรสำหรับการทดสอบที่มีอยู่

ในงานวิจัยนี้ได้เสนอวิธีการจัดลำดับความสำคัญกรณีทดสอบเพื่อมุ่งค้นหาข้อผิดพลาดได้อย่างรวดเร็ว ครบถ้วนเป็นหลัก โดยพิจารณาจากเกณฑ์ความครอบคลุมเพื่อประเมินระดับความสำคัญของกรณีทดสอบ ซึ่งแบ่งออกเป็นแนวทางหลักๆคือ การพิจารณากรณีทดสอบแต่ละกรณี หรือวิธีการแบบเบ็ดเสร็จ และการพิจารณารวมทั้งชุดทดสอบ หรือวิธีการแบบเพิ่ม

นอกจากนี้งานวิจัยนี้ยังได้เสนอเกณฑ์ความครอบคลุมสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งแบ่งออกเป็น 4 ลักษณะ ได้แก่ ความครอบคลุมระดับเหตุการณ์ ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ ความครอบคลุมระดับภารกิจ และความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจ โดยเกณฑ์ความครอบคลุมเหล่านี้ได้ถูกนำมาใช้ในการจัดลำดับความสำคัญกรณีทดสอบ นอกจากนี้งานวิจัยนี้ได้เสนอแนวทางการแก้ปัญหาที่เกิดขึ้นเมื่อมีการนำการจัดลำดับความสำคัญกรณีทดสอบมาใช้

งานวิจัยนี้ได้พัฒนาและสร้างเครื่องมือจัดลำดับความสำคัญกรณีทดสอบตามวิธีการที่ได้นำเสนอ รวมถึงทดสอบเครื่องมือดังกล่าวเพื่อเปรียบเทียบความสามารถของวิธีการจัดลำดับความสำคัญกรณีทดสอบที่นำเสนอ

ผลการทดสอบแสดงให้เห็นว่า วิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเบ็ดเสร็จให้ผลลัพธ์การจัดลำดับความสำคัญไม่ดีนัก และไม่เหมาะสมที่จะนำไปใช้ สำหรับวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบเพิ่มนั้นให้ประสิทธิภาพในการค้นพบข้อผิดพลาดที่ดี โดยแต่ละวิธีเหมาะสมกับลักษณะส่วนต่อประสานกราฟิกกับผู้ใช้ที่แตกต่างกัน เช่น วิธีการที่ใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์ให้ความสำคัญกับกรณีทดสอบที่มีเหตุการณ์แตกต่างกัน ในขณะที่เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์ให้ความสำคัญกับลำดับ ขั้นตอน และทางเลือกในกรทำงานที่แตกต่างกัน ซึ่งจะครอบคลุมกรณีทดสอบมากกว่าระดับเหตุการณ์

จากผลการทดสอบที่ได้แสดงให้เห็นว่า วิธีการจัดลำดับความสำคัญกรณีทดสอบดังกล่าวสามารถนำไปใช้เพื่อเพิ่มประสิทธิภาพในการทดสอบ ลดปริมาณกรณีทดสอบที่ต้องทดสอบลง เพิ่มความเร็วในการค้นพบข้อผิดพลาด และทำให้การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้สามารถใช้ทรัพยากรได้อย่างคุ้มค่า

6.2 ข้อเสนอแนะและแนวทางการพัฒนา

นอกจากแนวทางการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ที่นำเสนอในงานวิจัยนี้แล้ว ยังมีปัจจัยอีกมากมายที่จะทำให้การจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้แตกต่างกันออกไป เช่น เป้าหมายในการจัดลำดับความสำคัญ เกณฑ์ที่ใช้ในการพิจารณาลำดับความสำคัญ เป็นต้น ซึ่งวิธีการเหล่านี้อาจเหมาะสมกับลักษณะการทดสอบที่แตกต่างกันไป

สำหรับแนวทางการจัดลำดับความสำคัญกรณีทดสอบที่นำเสนอในงานวิจัยนี้สามารถพัฒนาต่อไปได้ ทั้งในส่วนของเกณฑ์ในการประเมินลำดับความสำคัญที่อาจพัฒนาให้อยู่ในรูปของซอฟต์แวร์เมตริกซ์หรือประเมินจากปัจจัยอื่นๆนอกจากเกณฑ์ความครอบคลุม ส่วนเกณฑ์ความครอบคลุมสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้ซึ่งสามารถนำไปใช้กับลักษณะงานอื่นๆ เช่น การสร้างกรณีทดสอบ เป็นต้น รวมถึงการพัฒนาเกณฑ์ความครอบคลุมใหม่ๆที่เหมาะสมกับลักษณะของงานที่แตกต่างออกไป และส่วนการกำหนดภารกิจของการทดสอบที่อาจพัฒนาให้สามารถกำหนดภารกิจต่างๆได้โดยอัตโนมัติ นอกจากนี้วิธีการและเทคนิคที่ใช้ในการจัดลำดับความสำคัญกรณีทดสอบยังจำเป็นต้องมีการศึกษาพัฒนาต่อไปอย่างต่อเนื่อง เพื่อเพิ่มประสิทธิภาพของการจัดลำดับความสำคัญกรณีทดสอบสำหรับทั้งการทดสอบซอฟต์แวร์ปกติและการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ต่อไป

แนวทางและวิธีการจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในงานวิจัยนี้ได้เสนอแนวทางและวิธีการแบบปกติ เน้นให้สามารถใช้ได้กับกรณีทั่วไป สำหรับงานที่มีลักษณะพิเศษหรืองานเฉพาะทางอย่างใดอย่างหนึ่งสามารถปรับเปลี่ยนให้มีลักษณะเหมาะสมตามต้องการได้อย่างอิสระ และแม้ว่าขั้นตอนการจัดลำดับความสำคัญกรณีทดสอบไม่ใช่ขั้นตอนปกติของการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ แต่งานวิจัยนี้เป็นสิ่งยืนยันว่าการจัดลำดับความสำคัญกรณีทดสอบสามารถช่วยลดภาระ ทรัพยากรที่ใช้ในกิจกรรมการทดสอบ รวมถึงเพิ่มประสิทธิภาพการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ได้ ซึ่งหากมีการพัฒนาต่อไปอย่างต่อเนื่องแล้ว ย่อมสามารถช่วยเพิ่มคุณภาพของซอฟต์แวร์ที่มีส่วนต่อประสานกราฟิกกับผู้ใช้เป็นองค์ประกอบได้อย่างแน่นอน

รายการอ้างอิง

1. White, L. Regression testing of GUI event interactions. Proceedings of the 1996 International Conference on Software Maintenance (November 1996): 350-358.
2. Memon, A. M., Pollack, M. E., and Soffa, M. L. Using a goal-driven approach to generate test cases for GUIs. Proceedings of the 21st International Conference on Software Engineering (May 1999): 257-266.
3. White, L., Almezen, H.,and Alzeidi N. User-based testing of GUI sequences and their interactions. Proceedings of the 12th International Symposium on Software Reliability Engineering (ISSRE'01) (November 2001) : 54-65.
4. Jorgensen, P. C. Software testing: A craftsman's approach. 2nd edition. Florida: CRC Press, 2002.
5. Memon, A. M. GUI testing: Pitfalls and process. IEEE Computer 35, 8 (August 2002): 87-88.
6. Shehady, R. K., and Siewiorek, D. P. A method to automate user interface testing using variable finite state machines. Proceedings of the 27th International Symposium on Fault-Tolerant Computing (FTCS '97) (June 1997): 80-88.
7. Kasik, D. J., George, H. G. Toward automatic generation of novice user test scripts. Proceedings of the 1996 Conference on Human Factors in Computing Systems (CHI 96) (April 1996): 244-251.
8. Dabóczi, T., Kollár, I., Simon, G., and Megyeri T. How to test graphical user interfaces. IEEE Instrumentation & Measurement Magazine 6, 3 (September 2003): 27-33.
9. Graham, T.C. N. A method for the formal testing of program visualization tools. Proceedings of the 4th Workshop on Program Comprehension (March 1996): 45-54.
10. Rothermel, G., Untch, R. H., and Harrold M. J. Prioritizing test cases for regression testing. IEEE Transactions on Software Engineering 27, 10 (October 2001): 929-948.
11. Memon, A. M., Pollack, M. E., and Soffa, M. L. Coverage criteria for GUI testing. Proceedings of the 8th European Software Engineering Conference (ESEC) and

9th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-9) (September 2001): 256-267.

12. Elbaum, S., Malishevsky, A. G., Rothermel, G. Test case prioritization: A family of empirical studies. IEEE Transactions on Software Engineering 28, 2 (February 2002): 159-182.
13. Srivastava, A., Thiagarajan, J. Effectively prioritizing tests in development environment. Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis (February 2002): 97-106.
14. Bizer, B. Software testing techniques. 2nd Edition. New York: Van Nostrand Reinhold, 1990.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

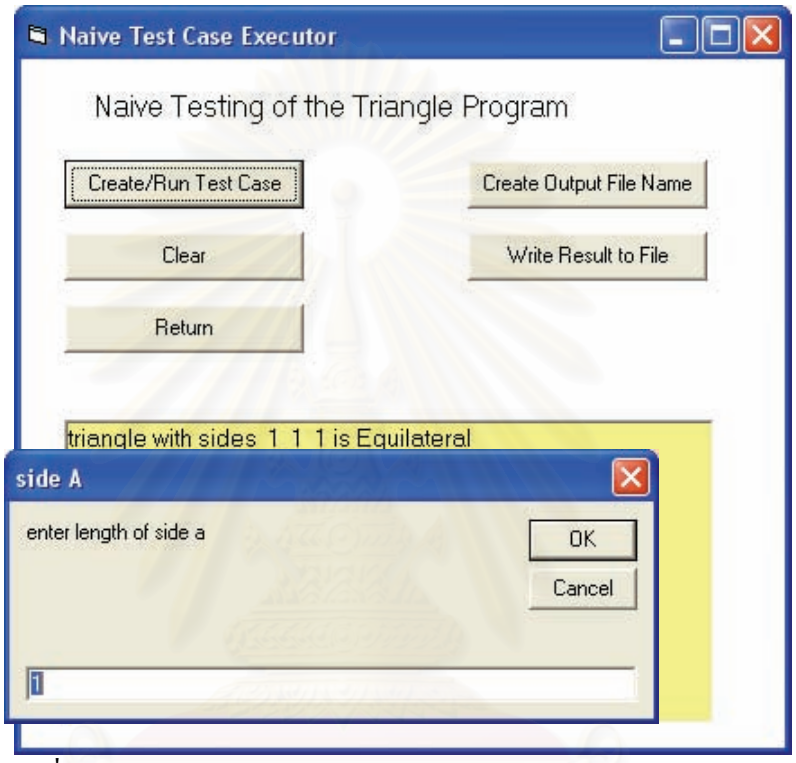
ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ตัวอย่างโปรแกรม

1. Naïve



รูปที่ ก-1 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Naive

ชื่อโปรแกรม Naive รุ่น -

ผู้พัฒนา Paul C. Jorgensen

ภาษาโปรแกรมที่ใช้พัฒนา Visual Basic

ที่มา หนังสือ "Software testing : A craftsman's Approach" 2nd Edition โดย Paul C. Jorgensen (ดาว์นโหลดจาก http://www.crcpress.com)

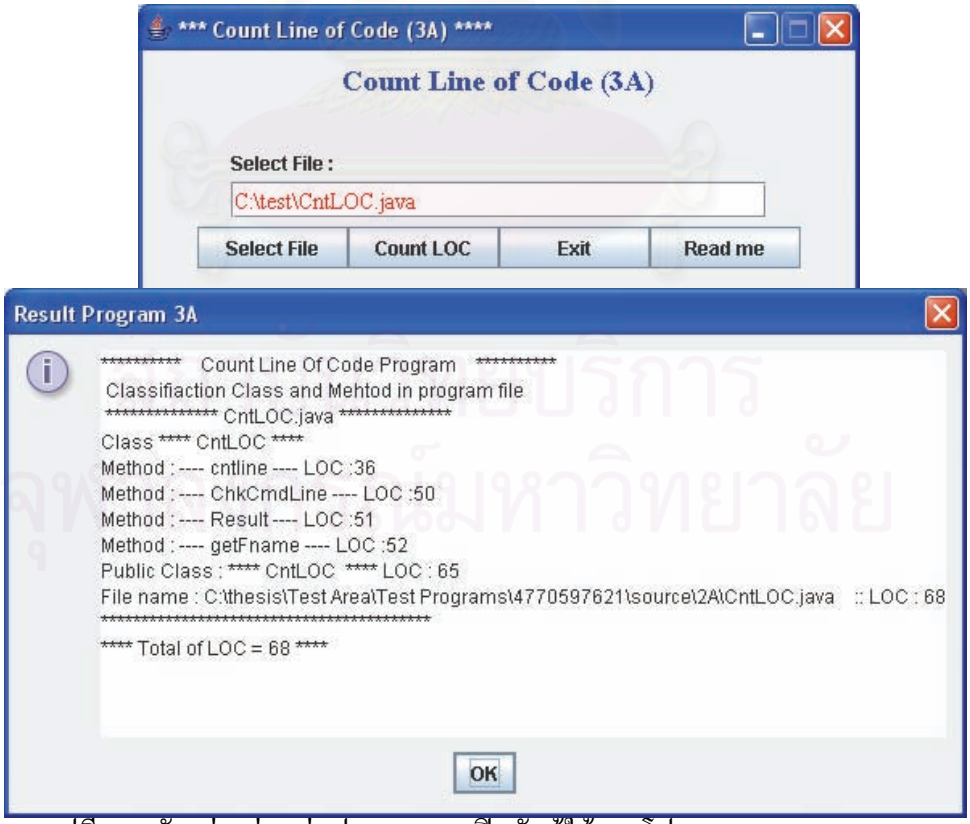
วัตถุประสงค์ โปรแกรมตัวอย่างเพื่อฝึกค้นหาข้อผิดพลาด

หน้าที่การทำงาน

- ทำงานตามตัวอย่างโปรแกรมในหนังสือ 3 ตัวอย่าง ได้แก่ Triangle problem, NextDate problem และ Commission problem มีส่วนช่วยเหลือแสดงข้อกำหนดของปัญหาแต่ละปัญหา และสามารถบันทึกผลการทำงานในแฟ้มข้อมูล

- Triangle problem เป็นโปรแกรมระบุประเภทของสามเหลี่ยม โดยแบ่งออกเป็น 4 ประเภท ได้แก่ สามเหลี่ยมด้านเท่า (Equilateral), สามเหลี่ยมหน้าจั่ว (Isosceles), สามเหลี่ยมด้านไม่เท่า (Scalene) และไม่เป็นสามเหลี่ยม โดยรับข้อมูลนำเข้าเป็นความยาวของด้านแต่ละด้านของสามเหลี่ยม
- NextDate problem เป็นโปรแกรมหาวันที่ เดือน และปีถัดไป โดยรับข้อมูลเข้าเป็นวันที่ เดือน และปี โดยปีต้องอยู่ระหว่างค.ศ. 1812 ถึงค.ศ. 2012
- Commission problem เป็นโปรแกรมคำนวณค่านายหน้า (commission) ของตัวแทนจำหน่ายปืน โดยจำหน่ายเป็นส่วนๆได้แก่ ไกปืน (lock), พานท้ายปืน (stock) และลำกล้องปืน (barrel) ซึ่งมีราคา 45, 30 และ 25 ดอลลาร์ตามลำดับ ในแต่ละเดือนต้องจำหน่ายให้ได้อย่างน้อยอย่างละ 1 ชิ้น แต่ไม่เกิน 70, 80 และ 90 ชิ้นตามลำดับ ภายใน 1000 ดอลลาร์แรก ผู้ขายจะได้ค่านายหน้า 10% และภายใน 800 ดอลลาร์ถัดมาได้ 15% และส่วนที่เกิน 1800 ดอลลาร์ จะได้ค่านายหน้า 20% โปรแกรมจะรับข้อมูลจำนวนชิ้นส่วนปืนแต่ละชนิดที่ขายได้ใน 1 เดือน

2. Object Counter



รูปที่ ก-2 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Object Counter

ชื่อโปรแกรม Object Counter รุ่น -

ผู้พัฒนา สมชาย วงศ์รัตนรักษ์

ภาษาโปรแกรมที่ใช้พัฒนา Java

ที่มา Assignment ลำดับที่ 3 วิชา Software Methodology (โปรแกรม 3A จากหนังสือ A Discipline for Software Engineering โดย Watts S. Humphrey)

วัตถุประสงค์ งานส่งเพื่อเก็บคะแนน

หน้าที่การทำงาน นับจำนวนบรรทัดของโปรแกรม (Line of code : LOC) ในแฟ้มที่เก็บโปรแกรมต้นฉบับ (Source code) ภาษาจาวา (Java) โดยแยกตามคลาสและเมท็อด (method) และมีการนับในแฟ้มที่เกี่ยวข้องด้วย (แฟ้มที่ปรากฏในคำสั่ง import)

3. CImage Demo

รูปที่ ก-3 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม CImage Demo

ชื่อโปรแกรม CImage Demo รุ่น 1.4.1

ผู้พัฒนา Troels Knakkergaard

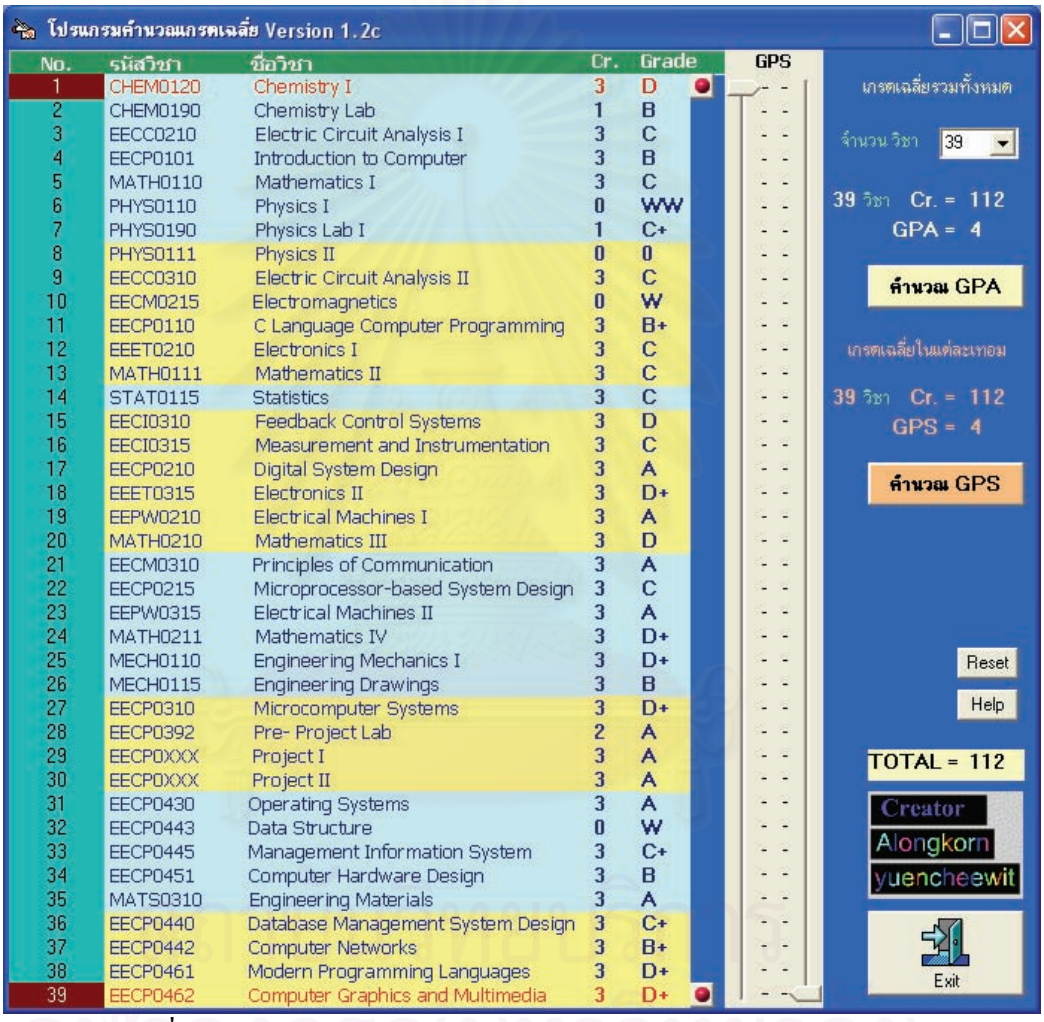
ภาษาโปรแกรมที่ใช้พัฒนา Visual C++

ที่มา ตัวอย่างโปรแกรมใน http://www.codeproject.com

<u>วัตถุประสงค์</u> สาธิตการใช้งานของคลาส CImage

<u>หน้าที่การทำงาน</u> แสดงภาพในแฟ้มข้อมูลสกุล BMP, GIF, JPEG, PNG, ICO และ TIF สามารถทำการตัด คัดลอก วางภาพดังกล่าวได้ และสามารถกลับข้อมูลภาพซ้ายขวา (Mirror), กลับข้อมูลภาพบนล่าง (Flip), กลับสีภาพ (Negative), เปลี่ยนสีเป็นระดับสีเทา (Grayscale), หมุนภาพ (Rotate) และแปลงรูปแบบวงรี (Ellipse transform) ได้

4. Grade **(โปรแกรมคำนวณเกรดเฉลี่ย)**



รูปที่ ก-4 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Grade

<u>ชื่อโปรแกรม</u> Grade (โปรแกรมคำนวณเกรดเฉลี่ย) <u>รุ่น</u> 1.2c

<u>ผู้พัฒนา</u> อลงกร ยืนชีวิต

<u>ภาษาโปรแกรมที่ใช้พัฒนา</u> -

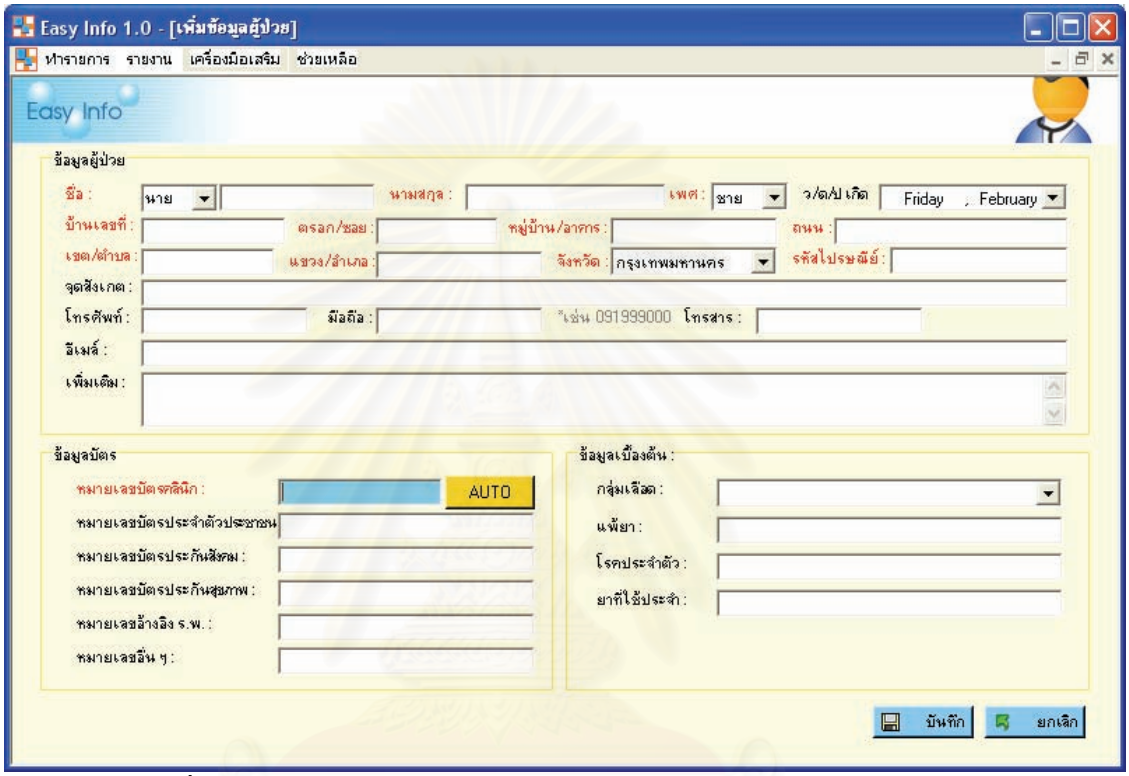
<u>ที่มา</u> โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย

ดาว์นโหลดจาก http://www.thaiware.com

วัตถุประสงค์ โปรแกรมคำนวณเกรดเฉลี่ย

หน้าที่การทำงาน คำนวณเกรดเฉลี่ยรวมทั้งหมด (Grade Point Average : GPA) และเกรดเฉลี่ยแต่ละเทอม (Grade Point average of Semester : GPS) โดยใช้ข้อมูลรหัสวิชา ชื่อวิชา หน่วยกิต และเกรด

5. Easy Info



รูปที่ ก-5 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Easy Info

ชื่อโปรแกรม Easy Info รุ่น 1.0

ผู้พัฒนา Sunday Soft (บริษัท เอส.ดี. อินโนเวชั่น จำกัด)

ภาษาโปรแกรมที่ใช้พัฒนา -

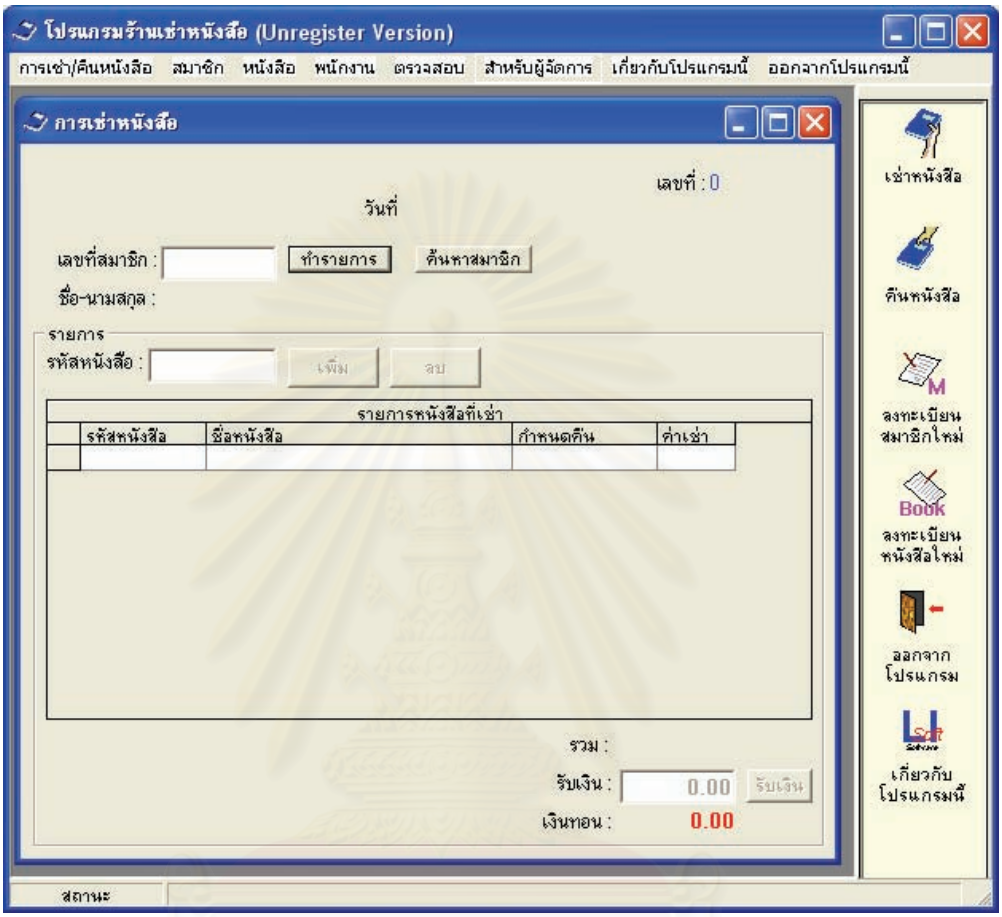
ที่มา โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย
ดาว์นโหลดจาก http://www.thaiware.com
(เว็บไซต์อย่างเป็นทางการ http://www.sdinno.com)

วัตถุประสงค์ โปรแกรมจัดเก็บข้อมูลคนไข้ในคลินิก

หน้าที่การทำงาน แบ่งการทำงานออกเป็น 2 ส่วนได้แก่ ส่วนคลินิกทั่วไป และส่วนคลินิกรักษาสัตว์ สามารถทำงานเก็บข้อมูลผู้ป่วย ค้นหา และแก้ไข รวมถึงพิมพ์รายงานข้อมูลผู้ป่วย พิมพ์

สติกเกอร์ที่อยู่ และส่ง SMS (Short Message Service) ทางโทรศัพท์ให้ผู้ป่วยได้ ในกรณีของคลินิกสัตว์จะเพิ่มการเก็บ ค้นหา แก้ไข และพิมพ์รายงานข้อมูลเจ้าของ

6. โปรแกรมร้านเช่าหนังสือ



รูปที่ ก-6 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมร้านเช่าหนังสือ

ชื่อโปรแกรม โปรแกรมร้านเช่าหนังสือ รุ่น 1.52

ผู้พัฒนา ชาญชัย อออุดมยุทธ (LISoft Software)

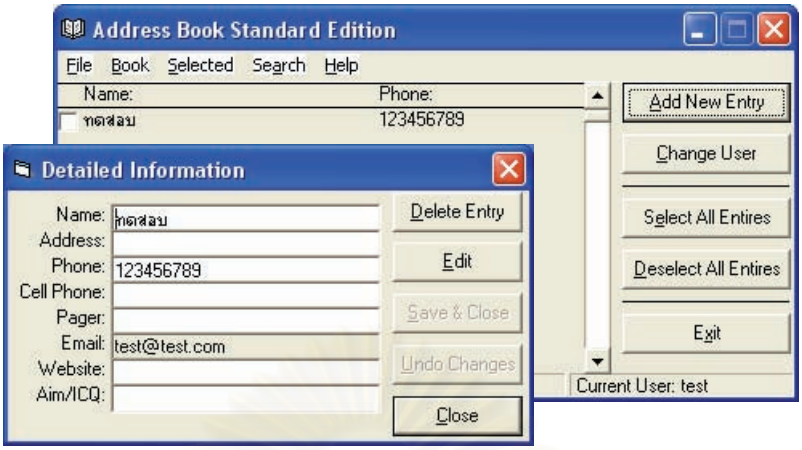
ภาษาโปรแกรมที่ใช้พัฒนา -

ที่มา โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย
ดาว์นโหลดจาก http://www.thaiware.com
(เว็บไซต์อย่างเป็นทางการ http://se-ed.net/lisoft)

วัตถุประสงค์ โปรแกรมบริหารการให้เช่า ยืมหนังสือ

หน้าที่การทำงาน จัดเก็บ แก้ไข ค้นหาข้อมูลการยืมหนังสือ คืนหนังสือ ข้อมูลสมาชิก ข้อมูลหนังสือ และข้อมูลพนักงาน รวมทั้งรายงานรายได้ ข้อมูลการเช่า รายงานสรุป รายชื่อสมาชิก รายชื่อหนังสือ

7. Address Book Standard Edition



รูปที่ ก-7 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของ
โปรแกรม Address Book Standard Edition

ชื่อโปรแกรม Address Book Standard Edition รุ่น 2.0

ผู้พัฒนา Jon Blankenhorn (JB ENTERPRISES)

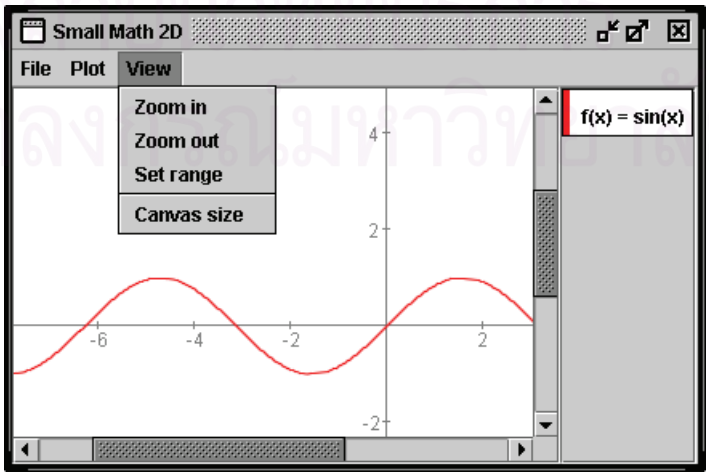
ภาษาโปรแกรมที่ใช้พัฒนา -

ที่มา โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย
ดาว์นโหลดจาก http://www.download.com

วัตถุประสงค์ โปรแกรมบันทึกที่อยู่ เบอร์โทรศัพท์ ข้อมูลส่วนตัวของบุคคลต่างๆ

หน้าที่การทำงาน จัดเก็บข้อมูลของบุคคลต่างๆที่ติดต่อด้วย เช่น ชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น สามารถเพิ่ม ลด ค้นหา พิมพ์ข้อมูล หรือส่งอีเมลให้บุคคลเหล่านั้นได้ โดยจะทำงานแยกจากกันตามข้อมูลของผู้ใช้แต่ละคน

8. Small Math 2D



รูปที่ ก-8 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Small Math 2D

ชื่อโปรแกรม Small Math 2D รุ่น 0.1.3

ผู้พัฒนา Small Math Dev

ภาษาโปรแกรมที่ใช้พัฒนา Java

ที่มา โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย

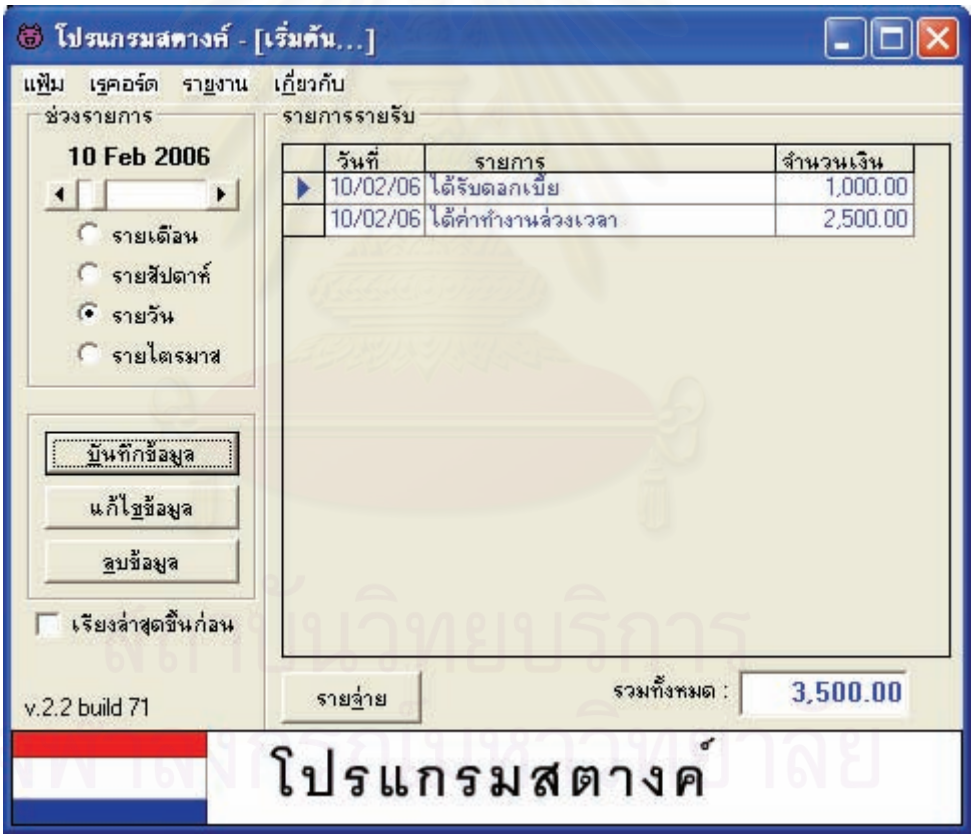
ดาว์นโหลดจาก http://www.download.com

(เว็บไซต์อย่างเป็นทางการ http://small-math.hiczechia.info)

วัตถุประสงค์ โปรแกรมสร้างกราฟจากสมการรูปแบบต่างๆ

หน้าที่การทำงาน วาดกราฟจากสมการลักษณะต่างๆ สามารถแสดงผลพร้อมกันหลายสมการ รวมถึงการกำหนดช่วง ตำแหน่ง อัตราขยายของส่วนแสดงผลได้

9. Satang (สตางค์)



รูปที่ ก-9 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมสตางค์

ชื่อโปรแกรม Satang (สตางค์) รุ่น 2.2 build 71

ผู้พัฒนา สราวุธ จันทร์โชติเสถียร

ภาษาโปรแกรมที่ใช้พัฒนา Visual Basic

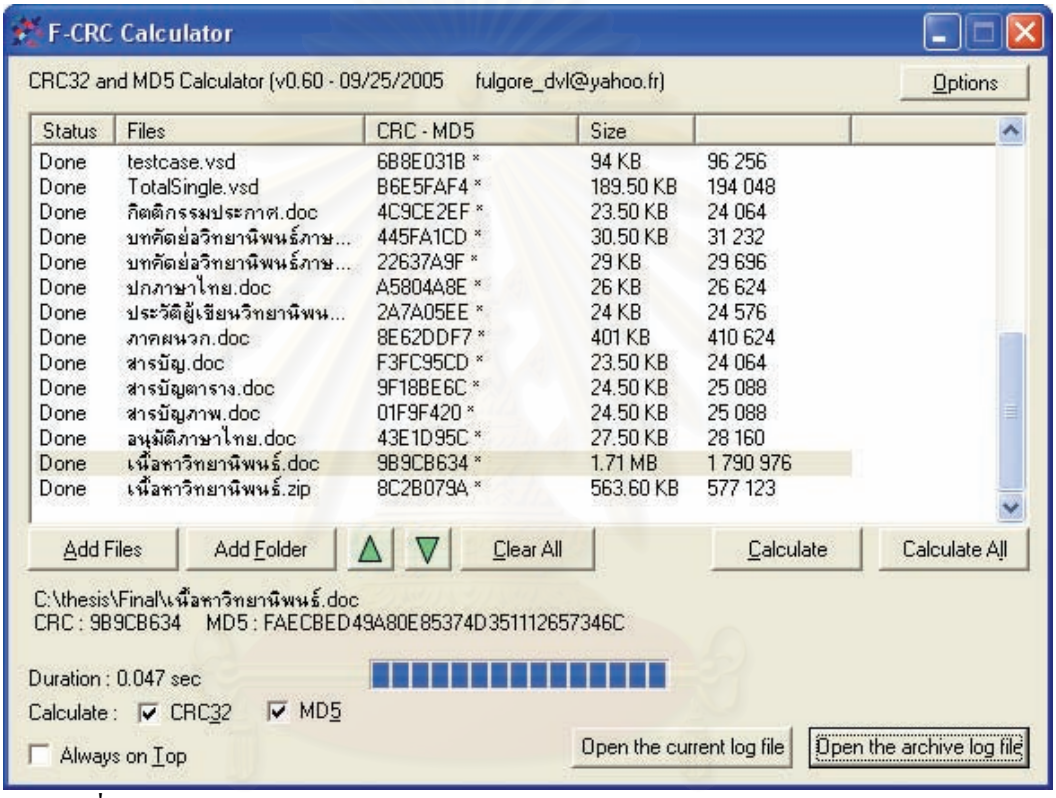
ที่มา โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย

ดาว์นโหลดจาก http://www.thaiware.com

วัตถุประสงค์ โปรแกรมจัดการบัญชีรายรับ-รายจ่ายส่วนบุคคล

หน้าที่การทำงาน โปรแกรมจัดการข้อมูลรายรับ รายจ่าย และข้อมูลบัตรเครดิต โดยสามารถเพิ่ม ลด แก้ไข และค้นหาข้อมูลได้ นอกจากนี้ยังสามารถรายงานรายรับ รายจ่าย ดุลรายรับรายจ่าย รายงานสรุป และแสดงผลในลักษณะกราฟ รวมทั้งสั่งพิมพ์รายงานบางประเภทได้

10. F-CRC Calculator



รูปที่ ก-10 ตัวอย่างส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม F-CRC Calculator

ชื่อโปรแกรม F-CRC Calculator รุ่น 0.6

ผู้พัฒนา Fulgore

ภาษาโปรแกรมที่ใช้พัฒนา -

ที่มา โปรแกรมที่ผู้พัฒนาเผยแพร่โดยไม่เก็บค่าใช้จ่าย

ดาว์นโหลดจาก http://www.download.com

(เว็บไซต์อย่างเป็นทางการhttp://fulgore-software.new.fr/f-crc/index.htm)

วัตถุประสงค์ โปรแกรมคำนวณค่า CRC และ MD5 ของแฟ้มข้อมูล

<u>หน้าที่การทำงาน</u> คำนวณค่า CRC (cyclic redundancy check หรือ การตรวจสอบด้วยส่วนซ้ำซ้อนแบบวน) และ MD5 ของแฟ้มข้อมูล สามารถคำนวณพร้อมกันได้ทีละหลายๆแฟ้ม สนับสนุนการลากและวาง (Drag and Drop) และบันทึกผลลัพธ์จากการทำงานในแฟ้มลงบันทึกเข้าออก (log file)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ขอบเขตการทดสอบ

ปกติก่อนดำเนินการทดสอบ จำเป็นต้องมีการกำหนดแผนการทดสอบหรือแนวทางในการทดสอบ ซึ่งประกอบด้วยรายละเอียดต่างๆ ไม่ว่าจะเป็นขอบเขต วิธีการ ระยะเวลา ผลลัพธ์ ความเสี่ยง ฯลฯ สำหรับมาตรฐานแผนการทดสอบซอฟต์แวร์สามารถอ้างอิงได้โดยอาศัยมาตรฐาน IEEE 829 แต่เนื่องจากรายละเอียดการทดสอบส่วนใหญ่ในงานวิจัยนี้ได้แสดงอย่างชัดเจนในเนื้อหาวิทยานิพนธ์แล้ว และรายละเอียดเกี่ยวกับระยะเวลา หน้าที่รับผิดชอบ หรือแผนงานทางธุรกิจต่างๆไม่ใช่ประเด็นที่เราสนใจ ดังนั้นในภาคผนวก ข นี้เราจะมุ่งเน้นที่ขอบเขตการทดสอบเป็นหลัก

1. Naïve

1.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนการทำงานตามปกติของส่วนหาชนิดสามเหลี่ยม (Triangle problem)
- ทดสอบการย่อ (minimize) ขยาย (maximize) คืนสภาพ (restore) และการปิดหน้าต่าง
- ทดสอบการกลับไปยังหน้าจอหลักและการล้างผลลัพธ์เดิม
- ทดสอบการยกเลิกการทำงานระหว่างการทำงาน
- ทดสอบการใส่ข้อมูลที่ผิดพลาด
- ทดสอบการทำงานซ้ำ 2 ครั้ง

1.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วน NextDate problem และ Commission problem
- ส่วนการจัดเก็บผลลัพธ์ลงในแฟ้มข้อมูล
- ส่วนช่วยเหลือและหน้าจอหลักที่เป็นทางเลือกในการเลือกทำงานปัญหาต่าง

2. Object Counter

2.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนการทำงานตามปกติของการนับบรรทัดคำสั่ง
- ทดสอบการย่อ (minimize) คืนสภาพ (restore) และปิดหน้าต่าง
- ทดสอบการใส่ชื่อแฟ้มข้อมูลด้วยการพิมพ์
- ทดสอบการยกเลิกการทำงานระหว่างการทำงาน
- ทดสอบการใส่ข้อมูลผิดพลาด
- ทดสอบส่วนช่วยเหลือ
- ทดสอบการออกจากโปรแกรม
- ทดสอบการทำงานซ้ำ 2 ครั้ง
- ทดสอบการทำงานร่วมกันของส่วนต่างๆ

2.2) ส่วนที่ไม่ต้องการทดสอบ

- ทดสอบกรณีแฟ้มข้อมูลเกิดข้อผิดพลาด
- รายละเอียดในขั้นตอนเลือกเปิดแฟ้มข้อมูลจากหน้าต่าง Open

3. CImage Demo

3.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบการเปิด ปิดแฟ้มข้อมูล และสร้างแฟ้มข้อมูลใหม่
- ทดสอบการออกจากโปรแกรม
- ทดสอบการคัดลอก (copy) และวาง (paste) ภาพ
- ทดสอบการ Mirror, Flip, Negative, Gray scale และ Rotate รูป รวมถึงการทำงานร่วมกัน
- ทดสอบการทำงานกับแฟ้มข้อมูลทีละหลายแฟ้ม
- ทดสอบการใส่ข้อมูลผิดพลาด
- ทดสอบการย่อ (minimize) ขยาย (maximize) คืนสภาพ (restore) และการปิดหน้าต่าง
- ทดสอบการใช้ปุ่มลัด (Hot key)

3.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนที่เกี่ยวข้องกับการพิมพ์
- ส่วนการยกเลิก (Undo) และทำซ้ำ (Redo)
- ส่วนการปรับแต่งการแสดงผล (View)
- ส่วนแสดงรีซอส (Load resource) แบบต่างๆ
- ส่วนการแปลงแบบวงรี (Demo ellipse transform)
- ส่วนจัดการหน้าต่าง
- ส่วนช่วยเหลือ
- ส่วนการทำงานโดยใช้ทูลบาร์ (Tool bar)
- ส่วนรายละเอียดในการเปิดแฟ้มข้อมูล

4. Grade **(โปรแกรมคำนวณเกรดเฉลี่ย)**

4.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบการคำนวณค่า GPA และ GPS
- ทดสอบการย่อ (minimize) ขยาย (maximize) คืนสภาพ (restore) และการปิดหน้าต่าง
- ทดสอบการ Reset ระบบช่วยเหลือ และการออกจากโปรแกรม
- ทดสอบการใส่ข้อมูลรูปแบบต่างๆตามที่ระบุไว้
- ทดสอบการใส่ข้อมูลที่ผิดพลาด
- ทดสอบการย้ายข้อมูลด้วยการตัด คัดลอก และวาง
- ทดสอบการกำหนดจำนวนวิชาในการคำนวณ GPA
- ทดสอบการใช้ GPS bar ในการกำหนดช่วงในการคำนวณ GPS

4.2) ส่วนที่ไม่ต้องการทดสอบ

-

5. Easy Info

5.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนลงทะเบียนใช้งานและส่วนเลือกประเภทการทำงาน
- ทดสอบการย่อ (minimize) ขยาย (maximize) คืนสภาพ (restore) การปิดหน้าต่าง และการออกจากโปรแกรมของส่วนคลินิกทั่วไป
- ทดสอบการเพิ่มข้อมูลผู้ป่วยของคลินิกทั่วไป
- ทดสอบการยกเลิกการเพิ่มข้อมูลผู้ป่วยระหว่างการทำงานในส่วนคลินิกทั่วไป
- ทดสอบการใส่ข้อมูลที่ผิดพลาดแล้วทำการบันทึก

5.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนคลินิกสัตว์
- ส่วนแก้ไขและค้นหาข้อมูล
- ส่วนรายงาน
- ส่วนพิมพ์สติกเกอร์และส่ง SMS
- ส่วนช่วยเหลือ
- การทำงานหลายงานพร้อมกัน

6. **โปรแกรมร้านเช่าหนังสือ**

6.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนลงทะเบียนหนังสือใหม่ และส่วนแก้ไข ลบข้อมูลหนังสือ
- ทดสอบส่วนลงทะเบียนสมาชิกใหม่ และส่วนแก้ไข ลบข้อมูลสมาชิก
- ทดสอบส่วนการเช่าและคืนหนังสือ
- ทดสอบการยกเลิกการทำงานระหว่างการทำงาน
- ทดสอบการใส่ข้อมูลที่ผิดพลาด

6.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนจัดการข้อมูลพนักงาน
- ส่วนตรวจสอบ สรุปข้อมูล และรายงาน
- ส่วนการกำหนดค่าสำหรับผู้จัดการ
- ส่วนแสดงรายละเอียดเกี่ยวกับโปรแกรม
- การปรับขนาดหน้าต่าง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- การออกจากโปรแกรม
- การใช้ทูลบาร์ด้านข้างและปุ่มลัดในการทำงาน
- การทำงานหลายงานพร้อมกัน

7. Address Book Standard Edition

7.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนเข้าใช้งาน (Log on) และออกจากโปรแกรม
- ทดสอบส่วนเปลี่ยนผู้ใช้ และเพิ่มผู้ใช้ใหม่
- ทดสอบส่วนเพิ่มข้อมูล
- ทดสอบการแก้ไขและลบข้อมูล
- ทดสอบการเลือกข้อมูลรูปแบบต่างๆ
- ทดสอบการค้นหาข้อมูล

7.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนการส่งออกและนำเข้าข้อมูล
- ส่วนการพิมพ์ และการส่งอีเมล
- ส่วนช่วยเหลือ และแสดงรายละเอียดเกี่ยวกับโปรแกรม
- การปรับขนาดหน้าต่าง
- การใช้ปุ่มลัดในการทำงาน

8. Small Math 2D

8.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนสร้าง บันทึกแฟ้มข้อมูล
- ทดสอบส่วนสร้างกราฟรูปแบบต่างๆ
- ทดสอบการลบกราฟที่ได้สร้างไว้
- ทดสอบการย่อขยาย (zoom) การกำหนดช่วง (range) และการกำหนดพื้นที่ทำงาน (canvas)
- ทดสอบการใส่ข้อมูลที่ผิดพลาด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- ทดสอบการย่อ (minimize) ขยาย (maximize) คืนสภาพ (restore) การปิดหน้าต่าง และการออกจากโปรแกรม

8.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนการปรับรูปแบบกราฟ โดยการเปลี่ยนค่าตัวแปรต่างๆ
- ส่วนรายละเอียดของการบันทึกแฟ้มข้อมูล

9. Satang (**สตางค์**)

9.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

- ทดสอบส่วนเพิ่ม แก้ไข และลบข้อมูลของส่วนรายจ่ายและรายรับ
- ทดสอบการปรับช่วงการแสดงผลระหว่าง รายเดือน รายสัปดาห์ รายวัน รายไตรมาส ของส่วนรายรับและรายจ่าย
- ทดสอบส่วนรายงานรูปแบบต่างๆ
- ทดสอบส่วนการออกจากโปรแกรม
- ทดสอบการใส่ข้อมูลที่ผิดพลาด
- ทดสอบการยกเลิกการทำงานระหว่างการทำงาน
- ทดสอบการใช้ปุ่มสั่งงานด้านข้าง

9.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนเปลี่ยนรหัสผ่าน
- ส่วนบัตรเครดิต
- ส่วนเครื่องคำนวณ
- ส่วนค้นหาข้อมูล
- การปรับขนาดหน้าต่าง
- ส่วนรายละเอียดเกี่ยวกับโปรแกรม
- การทำงานหลายงานพร้อมกัน

10. F-CRC Calculator

10.1) ส่วนที่ต้องการทดสอบ

ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ในส่วนต่างๆ ดังนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- ทดสอบส่วนการเพิ่ม ลบแฟ้มข้อมูลและโฟลเดอร์ ทั้งเดี่ยวและหลายๆแฟ้ม
- ทดสอบการเปลี่ยนลำดับของแฟ้ม
- ทดสอบการคำนวณรูปแบบต่างๆ
- ทดสอบการคำนวณซ้ำ
- ทดสอบการลากและวางแฟ้มข้อมูล
- ทดสอบการย่อ (minimize) คืนสภาพ (restore) การปิดหน้าต่าง และการออกจากโปรแกรม
- ทดสอบการบันทึกผลลัพธ์ลงในแฟ้ม

10.2) ส่วนที่ไม่ต้องการทดสอบ

- ส่วนบันทึกผลลัพธ์ลงในแฟ้ม
- ส่วนรายละเอียดการเลือกแฟ้มข้อมูลและโฟลเดอร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

ข้อผิดพลาดที่พบ

1. Naïve

ตารางที่ ค-1 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Naïve

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E1-1	ส่วนรับข้อมูลความยาวด้านของสามเหลี่ยมทั้ง 3 ด้าน	เกิด Run-time error '13' : Type mismatch โปรแกรมยุติการทำงาน	6
	รายละเอียด	เกิดข้อผิดพลาดเมื่อพยายามยุติการทำงานโดยกดปุ่ม Cancel หรือปิดหน้าต่างรับข้อมูล	
E1-2	ส่วนแสดงผลลัพธ์	ผลลัพธ์หายไป	9
	รายละเอียด	เกิดขึ้นเมื่อมีการย่อ (Minimize) หน้าต่าง	
E1-3	ส่วนรับข้อมูลว่าผลลัพธ์ถูกต้องหรือไม่	ไม่มีการแจ้งเตือนเมื่อใส่ข้อมูลผิดพลาด	15
	รายละเอียด	ไม่มีการแจ้งเตือนไม่ว่าจะใส่ข้อมูลแบบใดก็ตาม	

2. Object Counter

ตารางที่ ค-2 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Object Counter

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E2-1	ส่วนรับข้อมูลชื่อแฟ้ม	แสดงผล Exception บน Console ไม่มีข้อความเตือน โปรแกรมทำงานต่อได้	13
	รายละเอียด	เกิดเมื่อมีการใส่ชื่อแฟ้มโดยการพิมพ์แล้วแฟ้มดังกล่าวไม่มีอยู่จริง หรือไม่ใส่ชื่อแฟ้มเลย	

E2-2	ส่วนรับข้อมูลชื่อแฟ้ม	แสดงผล Exception บน Console ไม่มีข้อความเตือน โปรแกรมทำงานต่อได้	1
	รายละเอียด	เกิดเมื่อมีการใส่ชื่อแฟ้มเฉยๆ โดยไม่ระบุ Path	

3. CImage Demo

ตารางที่ ค-3 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม CImage Demo

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E3-1	ส่วนการคัดลอกและวางข้อมูลรูปภาพ	เกิดหน้าต่างแสดงข้อผิดพลาดขึ้นและโปรแกรมยุติการทำงาน	3
	รายละเอียด	เมื่อมีการเปิดแฟ้มใหม่ ขณะที่ยังไม่มีข้อมูล จะสามารถเลือกคำสั่งคัดลอก (Copy) ได้ เมื่อเลือกแล้วจะขึ้นหน้าต่างแสดงข้อความ “Unable to set Clipboard data” แต่ยังสามารถใช้คำสั่งวาง (Paste) ได้ ซึ่งเมื่อใช้คำสั่งดังกล่าวจะทำให้โปรแกรมเกิดข้อผิดพลาดและยุติการทำงาน	
E3-2	ส่วนคำสั่งหมุนรูป (Rotate)	สามารถเลือกคำสั่งได้ แต่ไม่มีอะไรเกิดขึ้น	5
	รายละเอียด	เมื่อมีการเปิดแฟ้มใหม่ สามารถเลือกคำสั่งดังกล่าวได้ ขณะที่คำสั่งอื่นมีสถานะเป็น disable แต่เมื่อเลือกแล้วไม่เกิดอะไรขึ้น	

4. Grade (โปรแกรมคำนวณเกรดเฉลี่ย)

ตารางที่ ค-4 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Grade (โปรแกรมคำนวณเกรดเฉลี่ย)

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E4-1	ส่วนกำหนดช่วงการคิดค่า GPS	ช่วงที่กำหนดไม่เปลี่ยนแปลง ทั้งที่มีการกำหนดใหม่	4

	รายละเอียด	เกิดขึ้นเมื่อมีการเลื่อน GPS slide bar โดยการใช้แป้นพิมพ์ ซึ่งแสดงผลการเลื่อนจริง แต่ไม่ส่งผลต่อช่วงที่ใช้คำนวณ		
E4-2	ส่วนรับข้อมูลเกรดของแต่ละวิชา		ทำให้ค่าที่คำนวณได้ผิดพลาด	1
	รายละเอียด	สามารถใส่เกรดได้มากกว่า 4		
E4-3	ส่วนรับข้อมูลเกรดของแต่ละวิชา		ไม่ยอมรับข้อมูลในรูปแบบที่กำหนด	1
	รายละเอียด	โปรแกรมระบุในส่วนช่วยเหลือว่าสามารถกำหนดเกรดเป็น EF ได้ แต่เมื่อใส่ข้อมูลจริง โปรแกรมกลับไม่ยอมรับข้อมูลดังกล่าว		

5. Easy Info

ตารางที่ ค-5 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Easy Info

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E5-1	ส่วนการลงทะเบียน	สามารถผ่านส่วนลงทะเบียนไปได้ โดยไม่ต้องทำการลงทะเบียนใช้งาน	1
	รายละเอียด	สามารถกดปิดหน้าต่างเพื่อผ่านส่วนนี้ไปได้เลย	
E5-2	ส่วนเพิ่มข้อมูลผู้ป่วย	ทำให้เกิดข้อผิดพลาด ซึ่งอาจทำให้โปรแกรมยุติการทำงาน	2
	รายละเอียด	สามารถกำหนดอายุผู้ป่วยเป็นลบ โดยการกำหนดวันเดือนปีเกิด ซึ่งทำให้เมื่อโปรแกรมอ้างถึงข้อมูลดังกล่าว เช่น การแก้ไข จะยุติการทำงาน	
E5-3	ส่วนช่วยเหลือในขั้นตอนการลงทะเบียน	ไม่มีการตอบสนองต่อการกระทำของผู้ใช้	1
	รายละเอียด	Link ไปที่ Web site ของบริษัทในส่วนช่วยเหลือการลงทะเบียนไม่ทำงาน	
E5-4	ส่วนเพิ่มข้อมูลผู้ป่วย	สามารถรับข้อมูลที่ขัดแย้งกันเองได้	5
	รายละเอียด	การกำหนดคำนำหน้าชื่อ ไม่มีผลต่อการกำหนดเพศ	

6. โปรแกรมร้านเช่าหนังสือ

ตารางที่ ค-6 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรมร้านเช่าหนังสือ

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E6-1	ส่วนแก้ไขข้อมูลสมาชิก	ข้อมูลที่บันทึกผิดพลาด	2
	รายละเอียด	การออกจากการแก้ไขข้อมูลสมาชิกโดยไม่มีการบันทึก การแก้ไขนั้นจะยังส่งผลอยู่	
E6-2	ส่วนแก้ไขข้อมูลหนังสือ	ข้อมูลที่บันทึกผิดพลาด	2
	รายละเอียด	การออกจากการแก้ไขข้อมูลหนังสือโดยไม่มีการบันทึก การแก้ไขนั้นจะยังส่งผลอยู่	
E6-3	ส่วนแก้ไขข้อมูลสมาชิก	ไม่รายงานผลการทำงาน ทำให้ผู้ใช้สับสน	1
	รายละเอียด	เมื่อค้นหาสมาชิกไม่พบ ไม่มีการแจ้งแก่ผู้ใช้ ข้อมูลเลื่อนไปอยู่ที่ลำดับท้ายสุด	
E6-4	ส่วนแก้ไขข้อมูลหนังสือ	ไม่รายงานผลการทำงาน ทำให้ผู้ใช้สับสน	1
	รายละเอียด	เมื่อค้นหาหนังสือไม่พบ ไม่มีการแจ้งแก่ผู้ใช้ ข้อมูลเลื่อนไปอยู่ที่ลำดับท้ายสุด	
E6-5	ส่วนการเช่าหนังสือ	เกิด Run-time Error '6160' : Data access error โปรแกรมยุติการทำงาน	2
	รายละเอียด	เกิดข้อผิดพลาดเมื่อมีการดับเบิ้ลคลิกที่ตารางขณะที่ไม่มีข้อมูล	
E6-6	ส่วนการลงทะเบียนและแก้ไขหนังสือ	ข้อมูลที่บันทึกผิดพลาด	2
	รายละเอียด	สามารถพิมพ์ประเภทหนังสือที่ไม่ได้กำหนดไว้ล่วงหน้าได้ แต่ข้อมูลที่จัดเก็บจะเป็นประเภทที่กำหนดไว้แล้วเท่านั้น ซึ่งไม่ตรงกับข้อมูลที่ใส่	
E6-7	ส่วนแก้ไขข้อมูลหนังสือ	ข้อมูลที่บันทึกผิดพลาด ไม่มีข้อความเตือน	1
	รายละเอียด	สามารถแก้ไขโดยลบชื่อหนังสือได้ ในขณะที่การลงทะเบียนหนังสือใหม่จะมีข้อความเตือน ถ้าพยายามไม่ใส่ชื่อหนังสือ	
E6-8	ส่วนแก้ไขข้อมูลสมาชิก	ข้อมูลที่บันทึกผิดพลาด ไม่มีข้อความเตือน	1
	รายละเอียด	สามารถแก้ไขโดยลบชื่อสมาชิกได้ ในขณะที่การลงทะเบียนสมาชิกใหม่จะมีข้อความเตือน ถ้าพยายามไม่ใส่ชื่อสมาชิก	
E6-9	ส่วนการลงทะเบียนและแก้ไขหนังสือ	ข้อมูลที่บันทึกไม่ตรงกับความเป็นจริง	2

	รายละเอียด	สามารถกำหนดราคาหนังสือเป็นค่าลบได้		
E6-10	ส่วนแก้ไขข้อมูลหนังสือ		ข้อมูลที่แสดงผลกับข้อมูลที่บันทึกไม่ตรงกัน ไม่มีการแจ้งเตือน	1
	รายละเอียด	สามารถกำหนดราคาหนังสือเป็นตัวอักษรได้ แม้จะไม่ส่งผลต่อข้อมูลที่บันทึก แต่ไม่มีข้อความแจ้งเตือน แม้เมื่อกดปุ่มบันทึกแล้ว รวมทั้งแสดงผลข้อมูลนั้นจนกว่าจะเลื่อนไปทำงานต่อไป		

7. Address Book Standard Edition

ตารางที่ ค-7 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Address Book Standard Edition

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E7-1	ส่วนการเพิ่มผู้ใช้ใหม่	ข้อมูลสูญหาย	2
	รายละเอียด	สร้างชื่อผู้ใช้ใหม่ทับผู้ใช้เดิม ทำให้ข้อมูลของผู้ใช้เดิมหายไป	
E7-2	ส่วนการเพิ่มและแก้ไขข้อมูลส่วนบุคคล	แสดงผลผิดพลาด ข้อมูลสูญหาย	2
	รายละเอียด	สามารถไม่ใส่ข้อมูลชื่อบุคคลได้ ทำให้ไม่มีการแสดงผลข้อมูลและแก้ไขไม่ได้	
E7-3	ส่วนลบข้อมูลส่วนบุคคล	แสดงผลผิดพลาด	2
	รายละเอียด	เมื่อพยายามลบข้อมูลทั้งหมด จะเกิดข้อผิดพลาดในการแสดงผล	
E7-4	ส่วนการเพิ่มผู้ใช้ใหม่	รับข้อมูลซ้ำซ้อน	4
	รายละเอียด	ในการเพิ่มผู้ใช้ใหม่ต้องกดปุ่ม New User 2 ครั้ง (เฉพาะการทำงานรอบแรก)	
E7-5	ส่วนการเปลี่ยนผู้ใช้	ทำงานไม่เป็นไปตามข้อกำหนด	2
	รายละเอียด	เมื่อยกเลิกการเปลี่ยนผู้ใช้โดยกดปุ่ม Cancel ต้องLog on ใหม่อีกครั้ง	

8. Small Math 2D

ตารางที่ ค-8 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Small Math 2D

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E8-1	ส่วนกำหนดช่วงแสดงผล	แสดงผลผิดพลาด ไม่มีการแจ้งเตือนข้อมูลผิดพลาด	1
	รายละเอียด	สามารถกำหนดช่วง โดยใช้ข้อมูลตัวอักษรได้	
E8-2	ส่วนกำหนดพื้นที่ทำงาน	ไม่มีการแจ้งเตือนข้อมูลผิดพลาด	2
	รายละเอียด	สามารถกำหนดขนาดพื้นที่ทำงานเป็นค่าลบได้	

9. Satang (สตางค์)

ตารางที่ ค-9 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม Satang (สตางค์)

รหัส	ส่วนที่เกิดข้อผิดพลาด	ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E9-1	ส่วนบันทึกและแก้ไข ข้อมูลรายรับ-รายจ่าย	ข้อมูลผิดพลาด ไม่มีการแจ้งเตือน	2
	รายละเอียด	สามารถใส่ข้อมูลเป็นค่าลบได้	
E9-2	ส่วนบันทึกและแก้ไข ข้อมูลรายรับ-รายจ่าย	ไม่มีการแจ้งเตือน	2
	รายละเอียด	เมื่อไม่ได้เลือกประเภทของรายรับ รายจ่ายจะกำหนดให้เป็นประเภทแรกสุด	
E9-3	ส่วนรายงานดุลรายรับ รายจ่าย	ชื่อหน้าต่างผิดพลาด	3
	รายละเอียด	ชื่อหน้าต่างแสดงรายเดือนเสมอ แม้จะเป็นข้อมูลรายวัน รายสัปดาห์ หรือรายไตรมาส	
E9-4	รายงานกราฟประวัติการใช้จ่าย	ไม่มีทางเลือกให้ผู้ใช้	2

	รายละเอียด	ไม่สามารถยกเลิกการสร้างกราฟกลางคันได้		
E9-5	รายงานกราฟประวัติการใช้จ่าย		เกิด Run-time error '11' : Division by zero โปรแกรมยุติการทำงาน	1
	รายละเอียด	เกิดขึ้นเมื่อพยายามสร้างกราฟที่มีข้อมูลเพียงชุดเดียวในช่วง		
E9-6	รายงานกราฟประวัติการใช้จ่าย		ไม่มีการแจ้งเตือน	1
	รายละเอียด	ไม่สามารถสร้างกราฟที่มีวันที่วันเดียวกันได้		

10. F-CRC Calculator

ตารางที่ ค-10 ข้อผิดพลาดที่พบในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของโปรแกรม F-CRC Calculator

รหัส	ส่วนที่เกิดข้อผิดพลาด		ลักษณะข้อผิดพลาด	จำนวนกรณีทดสอบที่พบข้อผิดพลาด
E10-1	ส่วนเริ่มการทำงาน		เกิดข้อผิดพลาด โปรแกรมยุติการทำงาน	3
	รายละเอียด	เกิดเมื่อเลือกข้อความที่แสดงผลในหน้าจอเริ่มต้นการทำงาน แล้วกดปุ่ม Calculate		
E10-2	ส่วนแสดงผล		Progress bar แสดงผลค้าง ไม่มีการ reset	14
	รายละเอียด	หลังการคำนวณ แม้ว่าจะเลือกแฟ้มอื่นแล้ว ส่วน Progress bar จะค้างอยู่ที่เดิม		
E10-3	ส่วนการคำนวณ		ไม่มีการคำนวณเกิดขึ้น	4
	รายละเอียด	เมื่อสั่งคำนวณโดยใช้ปุ่ม Calculate All แล้วครั้งหนึ่ง เมื่อเปลี่ยนแปลงรูปแบบการคำนวณใหม่แล้วสั่งทำงานอีกครั้ง จะไม่เกิดการคำนวณขึ้น		
E10-4	ส่วนการรับข้อมูล		ทำงานผิดพลาด ไม่มีการแจ้งเตือน	1
	รายละเอียด	เมื่อเลือกโฟลเดอร์พิเศษบางโฟลเดอร์ เช่น Control Panel จะนำข้อมูลแฟ้มในโฟลเดอร์ปัจจุบันมาทำงานแทน		

ภาคผนวก ง

การคำนวณค่าเฉลี่ยการเลือก

ปัญหาเกี่ยวกับค่าเฉลี่ยในการเลือกกรณีทดสอบที่ต้องการจากกลุ่มของกรณีทดสอบที่มีระดับความสำคัญเท่ากัน หรือในกรณีของการจัดลำดับกรณีทดสอบแบบสุ่มคือการเลือกกรณีทดสอบที่ต้องการจากกลุ่มของกรณีทดสอบทั้งหมด สามารถอธิบายในลักษณะของปัญหาดังต่อไปนี้

ปัญหา “มีของที่ต้องการอยู่ N ชิ้น จากของทั้งหมด S ชิ้น หากเลือกของขึ้นมาทีละชิ้น โดยไม่ใส่กลับคืน จำนวนครั้งเฉลี่ยในการเลือกของจนกว่าจะได้ของที่ต้องการเป็นเท่าไหร่”

สำหรับการแก้ปัญหาดังกล่าว เมื่อใช้คอมพิวเตอร์คำนวณโดยมีจำนวนรอบการทำซ้ำมากๆ พบว่าค่าที่ได้มีค่าใกล้เคียงกับสูตร

$$\mathrm{C} = \frac{S+1}{N+1}$$

เมื่อ C แทนจำนวนครั้งเฉลี่ยในการเลือกจนพบของที่ต้องการ

สำหรับการพิสูจน์ทางคณิตศาสตร์ สามารถพิสูจน์ได้ดังนี้ (พิสูจน์โดยคุณวรุตม์ สมาชิกอาวุโส Website http://www.mathcenter.net)

ของทั้งหมดมีอยู่ S ชิ้น ประกอบด้วยของที่ต้องการ N ชิ้น และของที่ไม่ต้องการ $S-N$ ชิ้น ในการเลือกครั้งที่ n จำนวนวิธีการหยิบที่จะได้ของที่ต้องการ คือ

$$N(P_{S-N,n-1})$$

โอกาสในการหยิบ n ครั้งจึงจะได้ของที่ต้องการ คือ

$$\frac{N(P_{S-N,n-1})}{P_{S,n}}$$

จำนวนครั้งในการหยิบที่เป็นไปได้ ได้แก่ $n = 1,\ 2,\ 3,\ \ldots,\ S-N+1$

ดังนั้นค่าเฉลี่ยจำนวนครั้งในการหยิบให้ได้ของที่ต้องการ คือ

$$N\sum_{n=1}^{S-N+1}\frac{n(P_{S-N,n-1})}{P_{S,n}}$$

หากพิจารณาโอกาสในการหยิบ n ครั้งจึงจะได้ของที่ต้องการอีกรูปแบบหนึ่งจะได้ว่า

$$\frac{P_{S-N,n-1}}{P_{S,n-1}} - \frac{P_{S-N,n}}{P_{S,n}}$$

ดังนั้น

$$\frac{N(P_{S-N,n-1})}{P_{S,n}} = \frac{P_{S-N,n-1}}{P_{S,n-1}} - \frac{P_{S-N,n}}{P_{S,n}}$$

ค่าเฉลี่ยจำนวนครั้งในการหยิบให้ได้ของที่ต้องการ คือ

$$\sum_{n=1}^{S-N+1} n\frac{N(P_{S-N,n-1})}{P_{S,n}} = \sum_{n=1}^{S-N+1} n\left(\frac{P_{S-N,n-1}}{P_{S,n-1}} - \frac{P_{S-N,n}}{P_{S,n}}\right)$$

$$\sum_{n=1}^{S-N+1} n\left(\frac{P_{S-N,n-1}}{P_{S,n-1}} - \frac{P_{S-N,n}}{P_{S,n}}\right) = \left(\frac{P_{S-N,0}}{P_{S,0}} - \frac{P_{S-N,1}}{P_{S,1}}\right) + 2\left(\frac{P_{S-N,1}}{P_{S,1}} - \frac{P_{S-N,2}}{P_{S,2}}\right) + \ldots$$

$$+(S-N+1)\left(\frac{P_{S-N,S-N}}{P_{S,S-N}} - \frac{P_{S-N,S-N+1}}{P_{S,S-N+1}}\right)$$

$$= \sum_{n=0}^{S-N} \frac{P_{S-N.n}}{P_{S,n}}$$

โดย

$$\frac{P_{S-N,n}}{P_{S,n}} = \frac{(S-N)!(S-n)!}{(S-N-n)!S!}$$

$$= \frac{(S-N)!(S-n)!}{(S-N-n-1)!S!}\left(\frac{1}{S-N-n}\right)$$

$$= \frac{(S-N)!(S-n)!}{(S-N-n-1)!S!}\left(\frac{1}{S-N-n}\right)\left(\frac{N+1}{N+1}\right)$$

$$= \frac{(S-N)!(S-n)!}{(S-N-n-1)!S!}\left(\frac{N+1}{S-N-n}\right)\left(\frac{1}{N+1}\right)$$

$$= \frac{(S-N)!(S-n)!}{(S-N-n-1)!S!}\left(\frac{(S-n+1)}{(S-N-n)}-1\right)\left(\frac{1}{N+1}\right)$$

$$= \left(\frac{1}{N+1}\right)\left(\frac{(S-N)!(S-n+1)!}{(S-N-n)!S!} - \frac{(S-N)!(S-n)!}{(S-N-n-1)!S!}\right)$$

$$= \left(\frac{1}{N+1}\right)\left(\frac{P_{S-N,n}}{P_{S,n-1}} - \frac{P_{S-N,n+1}}{P_{S,n}}\right)$$

เมื่อแทนค่าดังกล่าวลงในสมการค่าเฉลี่ยจะได้ว่า

$$\sum_{n=0}^{S-N}\frac{P_{S-N.n}}{P_{S,n}} = \frac{P_{S-N,0}}{P_{S,0}}+\sum_{n=1}^{S-N-1}\frac{P_{S-N,n}}{P_{S,n}}+\frac{P_{S-N,S-N}}{P_{S,S-N}}$$

$$= 1+\frac{1}{N+1}\left(\sum_{n=1}^{S-N-1}\frac{P_{S-N,n}}{P_{S,n-1}}-\frac{P_{S-N,n+1}}{P_{S,n}}\right)+\frac{(S-N)!N!}{S!}$$

$$\sum_{n=0}^{S-N}\frac{P_{S-N.n}}{P_{S,n}} = 1+\frac{1}{N+1}\left(\sum_{n=1}^{S-N-1}\frac{P_{S-N,n}}{P_{S,n-1}}-\frac{P_{S-N,n+1}}{P_{S,n}}\right)+\frac{(S-N)!N!}{S!}$$

$$= 1+\frac{1}{N+1}\left(\frac{P_{S-N,1}}{P_{S,0}}-\frac{P_{S-N,S-N}}{P_{S,S-N-1}}\right)+\frac{(S-N)!N!}{S!}$$

$$= 1+\frac{1}{N+1}\left((S-N)-\frac{(S-N)!(N+1)!}{S!}\right)+\frac{(S-N)!N!}{S!}$$

$$= 1+\frac{(S-N)}{N+1}$$

$$= \frac{S+1}{N+1}$$

ดังนั้นค่าเฉลี่ยจำนวนครั้งในการหยิบให้ได้ของที่ต้องการ คือ

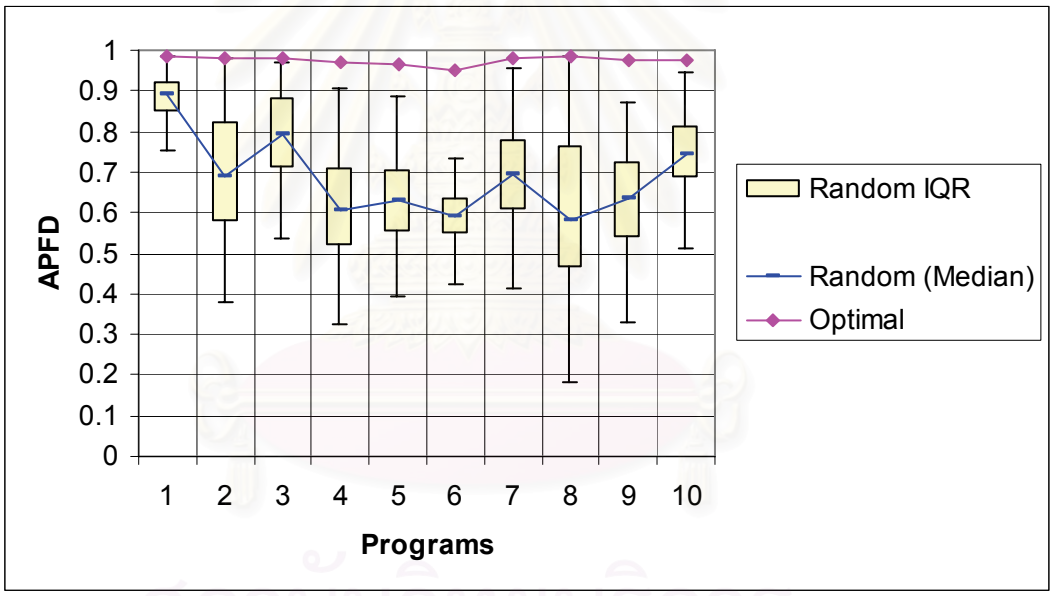
$$\frac{S+1}{N+1}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก จ

การแจกแจงข้อมูล

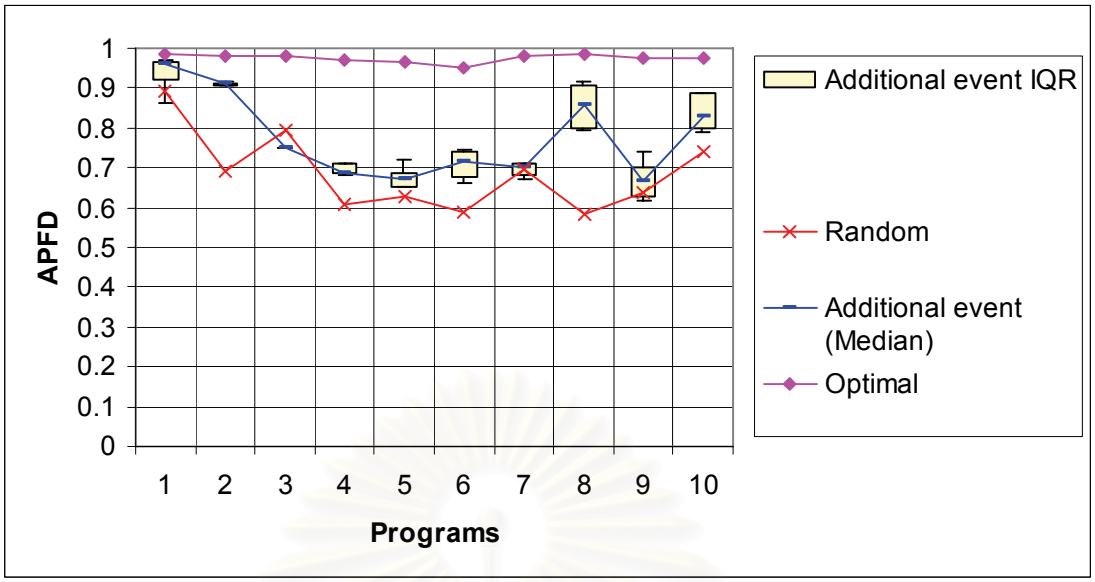
เนื่องจากการจัดลำดับความสำคัญกรณีทดสอบสำหรับส่วนต่อประสานกราฟิกกับผู้ใช้โดยใช้วิธีการจัดลำดับความสำคัญตามเกณฑ์ความครอบคลุมแบบเพิ่มให้ผลลัพธ์แตกต่างกันขึ้นกับลำดับกรณีทดสอบในชุดทดสอบก่อนการจัดลำดับ ซึ่งค่า APFD ที่ได้นั้นเป็นการหาค่าเฉลี่ยเลขคณิตจากการทดสอบทั้งสิ้น 100 ครั้ง ในภาคผนวก ง นี้จึงได้นำเสนอแผนภูมิซึ่งแสดงถึงการแจกแจงข้อมูล APFD ทั้งหมดในลักษณะของ Boxplot[†] โดยแผนภูมิทั้งหมดสร้างขึ้นจากข้อมูลการจัดลำดับกรณีทดสอบจำนวน 100 ครั้ง



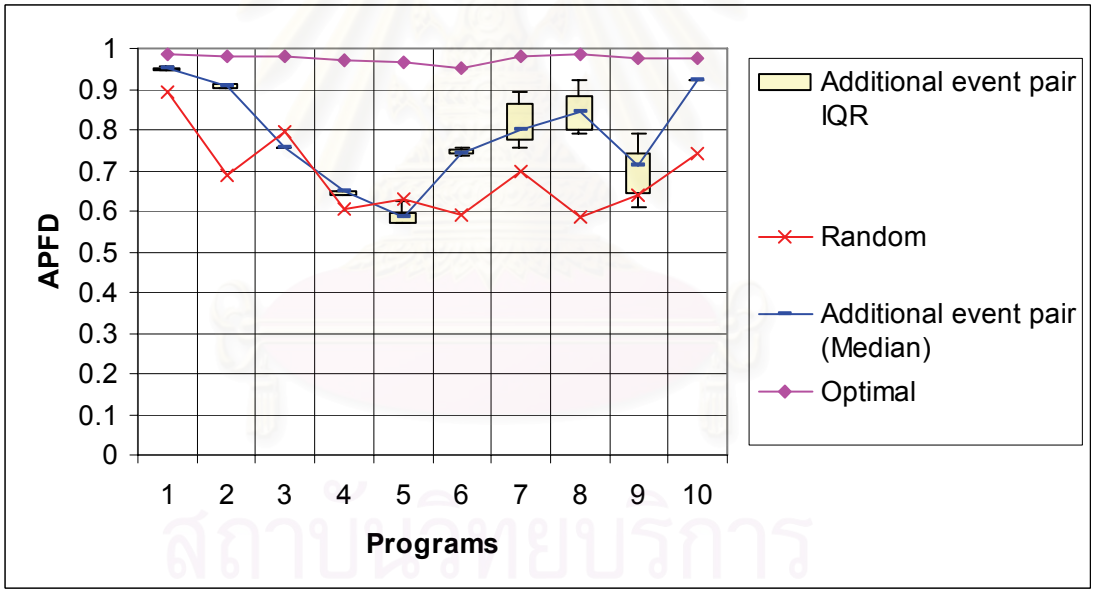
รูปที่ จ-1 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับกรณีทดสอบแบบสุ่ม

จุฬาลงกรณ์มหาวิทยาลัย

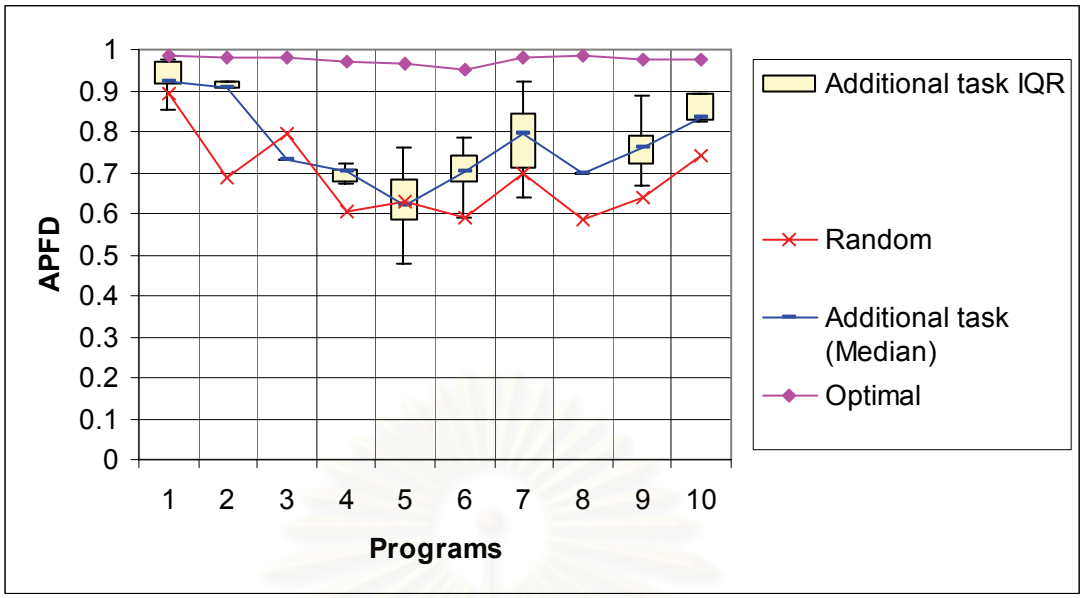
[†] Boxplot คือ แผนภูมิที่ใช้แสดงการกระจายของข้อมูล โดยเส้นตรงกลางของกล่องแสดงถึงค่ามัธยฐาน (Median) ขอบบนและล่างของกล่องแสดงค่าควอร์ไทล์ที่ 1 และ 3 เส้นที่ต่อออกจากกล่องแสดงค่ามากที่สุดและน้อยที่สุดที่อยู่ภายในระยะ 1.5 เท่าของพิสัยระหว่างควอร์ไทล์ (Interquartile range : IQR : ระยะห่างระหว่างควอร์ไทล์ที่ 1 ถึงควอร์ไทล์ที่ 3) สำหรับค่านอกเหนือจากนี้ถือเป็นค่าผิดปกติ (outliers)



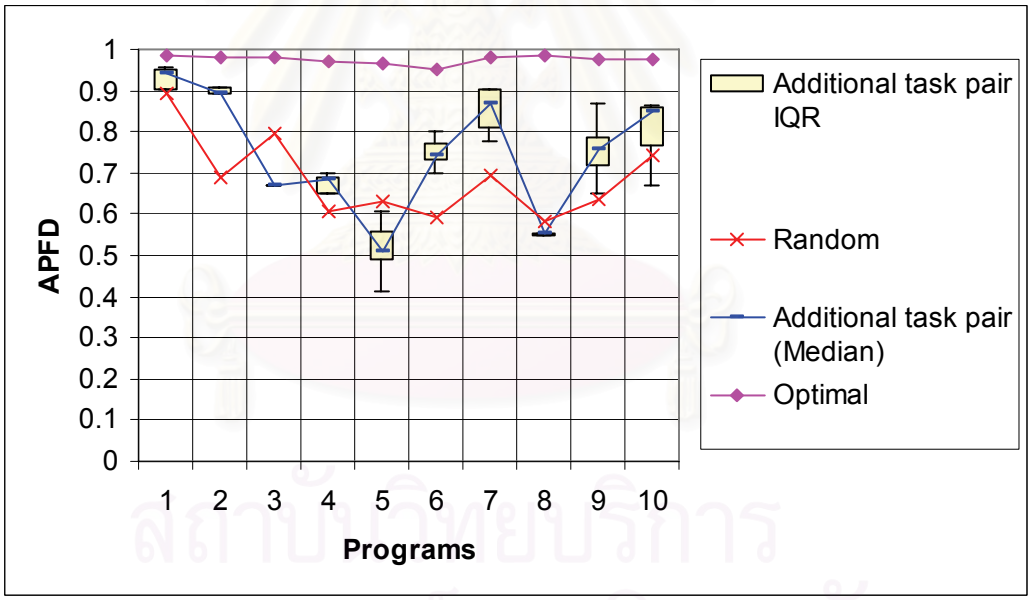
รูปที่ จ-2 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับเหตุการณ์แบบเพิ่ม



รูปที่ จ-3 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างเหตุการณ์แบบเพิ่ม



รูปที่ จ-4 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับภารกิจแบบเพิ่ม



รูปที่ จ-5 แผนภูมิแสดงการแจกแจงของค่า APFD สำหรับการจัดลำดับความสำคัญกรณีทดสอบโดยใช้เกณฑ์ความครอบคลุมระดับความสัมพันธ์ระหว่างภารกิจแบบเพิ่ม

ภาคผนวก ฉ

คู่มือการใช้งานเครื่องมือจัดลำดับความสำคัญกรณีทดสอบ สำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้

เครื่องมือจัดลำดับความสำคัญกรณีทดสอบสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI Test Case Prioritizer) เป็นเครื่องมือที่ใช้จัดลำดับความสำคัญกรณีทดสอบที่พัฒนาขึ้นโดยมีเป้าหมายและลักษณะการทำงานตามวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้ สำหรับรายละเอียดเกี่ยวกับการใช้งานเครื่องมือมีดังนี้

1. ความต้องการพื้นฐานของเครื่องมือ

1.1 ด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลใช้ความถี่สัญญาณนาฬิกาตั้งแต่ 400 เมกกะเฮิรตซ์ (MHz) ขึ้นไป
- หน่อยความจำหลัก (RAM) ขนาดตั้งแต่ 128 เมกกะไบต์ (MB) ขึ้นไป
- เมาส์ และคีย์บอร์ด

1.2 ด้านซอฟต์แวร์

- ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ตั้งแต่รุ่น 2000 หรือเอ็กซ์พี ขึ้นไป

2. รูปแบบข้อมูล

เครื่องมือแบ่งการทำงานกับข้อมูลออกเป็น 2 ประเภท ได้แก่ ข้อมูลกรณีทดสอบซึ่งมีทั้งการรับเข้าและส่งออกจากเครื่องมือ และข้อมูลค่า APFD ที่ได้จากการทดสอบการจัดลำดับความสำคัญกรณีทดสอบซึ่งเป็นข้อมูลส่งออกจากเครื่องมือเพียงทิศทางเดียว สำหรับรายละเอียดเกี่ยวกับรูปแบบข้อมูลและแฟ้มข้อมูลที่รองรับการทำงานทั้งสองแบบดังกล่าว มีดังนี้

2.1 ข้อมูลกรณีทดสอบ

กรณีทดสอบประกอบด้วยข้อมูล ดังนี้

- รหัสประจำตัวกรณีทดสอบ (ID) มีลักษณะเป็นตัวอักษรหรือตัวเลขต่อเนื่องกัน ใช้ระบุตัวตนของกรณีทดสอบ
- ลำดับข้อมูลทดสอบ (Test sequence) ประกอบด้วยสถานะเริ่มต้นตามด้วยลำดับของเหตุการณ์หรือภารกิจ ใช้เครื่องหมายจุลภาค (,) เป็นตัวคั่น

- ข้อผิดพลาดที่พบ (Error) ข้อผิดพลาดที่กรณีทดสอบนั้นค้นพบ สำหรับกรณีทดสอบที่พบข้อผิดพลาดมากกว่า 1 กรณี ให้ใช้เครื่องหมายจุลภาค (,) เป็นตัวคั่น ในกรณีที่ไม่พบข้อผิดพลาดหรือไม่มีข้อมูลให้เว้นว่างไว้หรือใช้เครื่องหมายยัติภังค์ (-) แทน
- ลำดับความสำคัญ (Priority) ลำดับความสำคัญที่ได้มาจากการจัดลำดับความสำคัญกรณีทดสอบเป็นค่าตัวเลข กรณีที่ไม่มีข้อมูลนี้ให้เว้นว่างไว้หรือใช้เครื่องหมายยัติภังค์ (-) แทน
- ลำดับการทดสอบ (Order) ลำดับการทดสอบที่ได้จากการเรียงลำดับความสำคัญของกรณีทดสอบทั้งหมดในชุดทดสอบ กรณีที่มีความสำคัญเท่ากันจะเรียงตามลำดับในชุดทดสอบ ไม่มีการอ่านหรือบันทึกข้อมูลนี้

สำหรับสถานะเริ่มต้น เหตุการณ์ ภารกิจหรือข้อผิดพลาดให้แทนด้วยชื่อเฉพาะที่ประกอบด้วยตัวอักษร ตัวเลข เครื่องหมาย ( ) “ _ หรือช่องว่าง โดยไม่ซ้ำกับสถานะเริ่มต้น เหตุการณ์ ภารกิจหรือข้อผิดพลาดอื่น

สำหรับทิศทางการนำเข้าและส่งออกข้อมูลกรณีทดสอบของเครื่องมือแสดงดังตาราง ฉ-1

ตารางที่ ฉ-1 ทิศทางการนำเข้าและส่งออกข้อมูลกรณีทดสอบของเครื่องมือ

	รหัสประจำกรณีทดสอบ (ID)	ลำดับข้อมูลทดสอบ (Test sequence)	ข้อผิดพลาดที่พบ (Error)	ลำดับความสำคัญ (Priority)	ลำดับการทดสอบ (Order)
นำเข้า (Input)	✓	✓	*	*	-
ส่งออก (Output)	✓	✓	✓	✓	-

หมายเหตุ เครื่องหมาย ✓ แทนข้อมูลที่จำเป็น * แทนข้อมูลที่จะมีหรือไม่ก็ได้ และ – แทนไม่ต้องมีข้อมูลดังกล่าว

สำหรับการข้อมูลกรณีทดสอบทั้งหมดนั้นจัดเก็บในแฟ้มข้อมูล โดยแฟ้มข้อมูล 1 แฟ้มถือเป็น 1 ชุดทดสอบ โดยประเภทของแฟ้มข้อมูลที่ใช้จัดเก็บข้อมูลกรณีทดสอบเป็นไปได้ 2 ประเภท ได้แก่ แฟ้มข้อความ (Text file) และแฟ้มเอ็กซ์เอ็มแอล โดยมีลักษณะการจัดเก็บข้อมูลดังนี้

สำหรับแฟ้มข้อความจะจัดเก็บข้อมูลแต่ละส่วนเรียงตามลำดับโดยใช้แท็บเป็นตัวคั่นและแยกกรณีทดสอบออกจากกันโดยการขึ้นบรรทัดใหม่

ส่วนแฟ้มเอ็กซ์เอ็มแอลนั้นจะประกอบด้วยแท็กและคุณสมบัติ (Attributes) ต่างๆ ดังแสดงในตาราง ฉ-2 และตัวอย่างข้อมูลกรณีทดสอบในแฟ้มเอ็กซ์เอ็มแอลแสดงในรูป ฉ-1

ตารางที่ ฉ-2 แท็กและคุณสมบัติต่างๆที่ใช้ในการกำหนดข้อมูลกรณีทดสอบ

แท็ก	คุณสมบัติ	หน้าที่
<Testcase> </Testcase>	name priority	ใช้แสดงจุดเริ่มและจุดสิ้นสุดของกรณีทดสอบ โดยมีคุณสมบัติ name ใช้ระบุชื่อหรือรหัสประจำตัวของกรณีทดสอบ และ priority ใช้ระบุลำดับความสำคัญของกรณีทดสอบ
<Precondition> </Precondition>		ใช้ระบุสถานะตั้งต้นของกรณีทดสอบนั้นๆ
<TOperator> <TOperator>		กำหนดเหตุการณ์หรือหน้าที่หนึ่งๆในกรณีทดสอบนั้นๆ โดยมีลำดับการทำงานตามลำดับที่ปรากฏในกรณีทดสอบ
<Defect> </Defect>		ใช้ระบุข้อผิดพลาดแต่ละกรณีที่กรณีทดสอบค้นพบ

```
<Testcase name="T1" priority="1">
        <Precondition>  InitializeProgram       </Precondition>
        <TOperator>     PressButton("OK")       </TOperator>
        <TOperator>     CloseWindow(Main)       </TOperator>
        <Defect>        E1                      </Defect>
        <Defect>        E2                      </Defect>
</Testcase>
```

รูปที่ ฉ-1 ตัวอย่างการจัดเก็บข้อมูลกรณีทดสอบในแฟ้มเอ็กซ์เอ็มแอล

2.2 ข้อมูลค่า APFD

กรณีที่มีการทำการทดสอบผลการจัดลำดับความสำคัญกรณีทดสอบแบบต่อเนื่อง ผลลัพธ์การจัดลำดับความสำคัญกรณีทดสอบที่วัดผ่านค่า APFD จะถูกจัดเก็บในแฟ้มข้อมูล ซึ่งสามารถเลือกประเภทของแฟ้มที่ใช้จัดเก็บได้ 2 แบบ ได้แก่

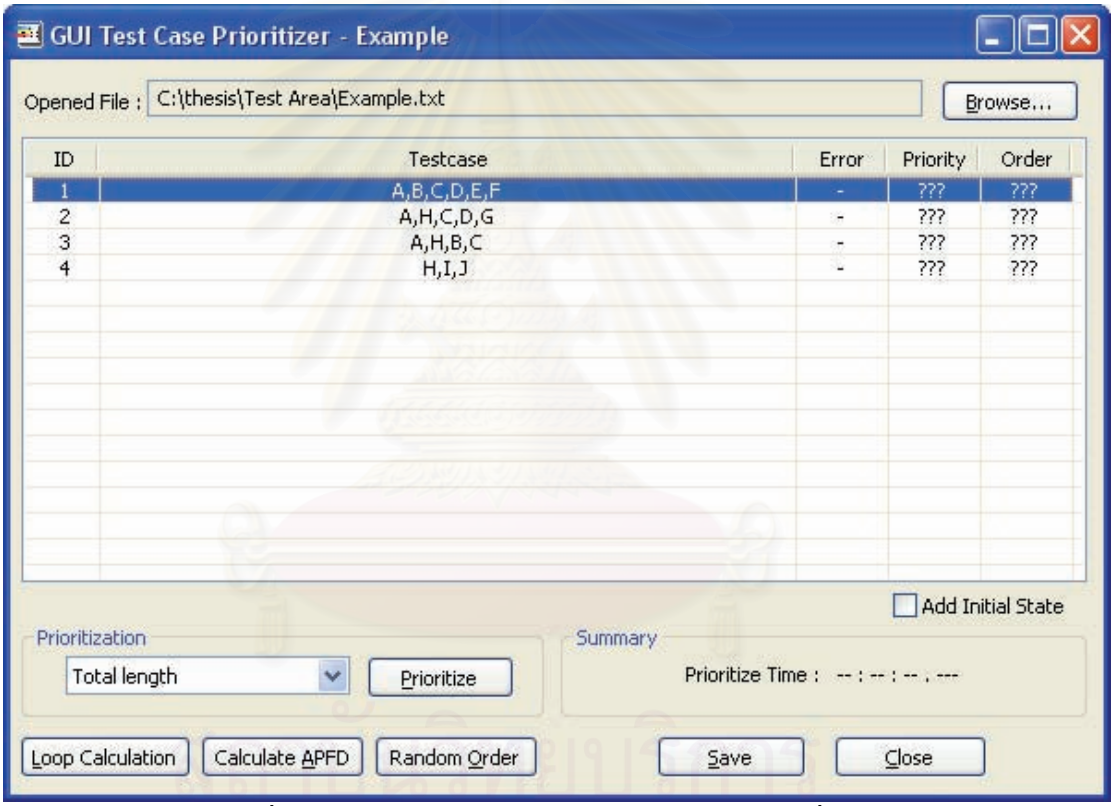
แฟ้มข้อความ (Text file) ซึ่งค่า APFD ถูกจัดเก็บโดยใช้แท็บเป็นตัวคั่น และแบ่งการทดสอบแต่ละครั้งด้วยการขึ้นบรรทัดใหม่ สามารถกำหนดให้มีการระบุชื่อวิธีการจัดลำดับ

ความสำคัญกรณีทดสอบที่ใช้เป็นชื่อคอลัมน์หรือไม่ก็ได้ หากไม่ระบุจะเรียงลำดับตามวิธีการที่กำหนดให้ทำการทดสอบ

แฟ้มข้อความที่ใช้จุลภาคเป็นตัวคั่น (CSV file) มีข้อกำหนดและลักษณะข้อมูลเช่นเดียวกับแฟ้มที่ใช้แท็บเป็นตัวคั่น ต่างกันที่ใช้เครื่องหมายจุลภาคแทนแท็บในการแบ่งระหว่างคอลัมน์

3. วิธีการใช้งาน

3.1 การจัดลำดับความสำคัญกรณีทดสอบ



รูปที่ ฉ-2 ส่วนต่อประสานกราฟิกกับผู้ใช้หลักของเครื่องมือ

ขั้นตอนใช้เครื่องมือในการจัดลำดับความสำคัญกรณีทดสอบมีดังนี้

1. เมื่อเริ่มทำงานเครื่องมือจะแสดงส่วนต่อประสานกับผู้ใช้ดังรูป ฉ-2
2. เลือกแฟ้มข้อมูลชุดทดสอบที่ต้องการจัดลำดับความสำคัญ โดยใช้ปุ่ม Browse…
3. ข้อมูลกรณีทดสอบทั้งหมดในแฟ้มจะแสดงผลในตาราง

4. กรณีที่ข้อมูลกรณีทดสอบไม่มีข้อมูลสถานะตั้งต้นให้เลือก Add Initial State เพื่อสร้างสถานะตั้งต้นจำลอง (Dummy) ให้กรณีทดสอบ โดยใช้สมมุติฐานว่ากรณีทดสอบทุกกรณีมีสถานะตั้งต้นเหมือนกันหมด

5. เลือกวิธีการจัดลำดับความสำคัญกรณีทดสอบโดยใช้คอมโบบ็อกซ์ภายใต้หัวข้อ Prioritization

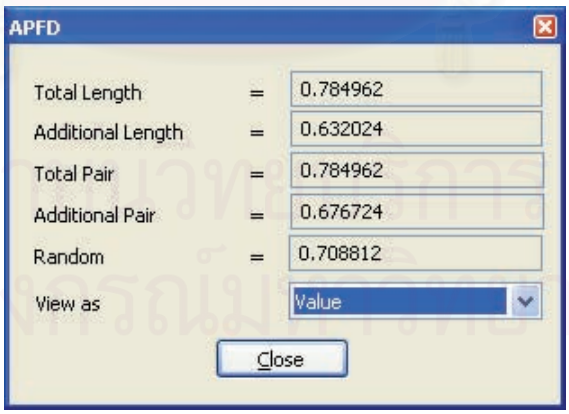
6. กดปุ่ม Prioritize เพื่อเริ่มการจัดลำดับความสำคัญกรณีทดสอบ

7. ค่าความสำคัญของกรณีทดสอบแต่ละกรณีจะถูกแสดงในคอลัมน์ Priority ในตาราง และลำดับการทดสอบจะแสดงในคอลัมน์ Order เวลาที่ใช้ในการจัดลำดับความสำคัญแสดงในหัวข้อ Summary

8. หากต้องการให้มีการสลับตำแหน่งกรณีทดสอบแบบสุ่ม สามารถกดปุ่ม Random Order

9. กรณีที่ต้องการบันทึกข้อมูลกรณีทดสอบที่ได้รับการจัดลำดับความสำคัญแล้ว สามารถกดปุ่ม Save

10. หากต้องการตรวจสอบค่า APFD ของวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบต่างๆ ให้กดปุ่ม Calculate APFD ได้เลย โดยไม่ต้องจัดลำดับความสำคัญกรณีทดสอบก่อน เครื่องมือจะทำการจัดลำดับความสำคัญและคำนวณโดยอัตโนมัติและไม่ส่งผลกระทบต่อข้อมูลกรณีทดสอบที่กำลังทำงานอยู่ ดังแสดงในรูป ฉ-3 ซึ่งสามารถเลือกรูปแบบการแสดงผลได้ 2 แบบ ได้แก่ แบบค่า (value) และแบบสูตรคำนวณ (formula)



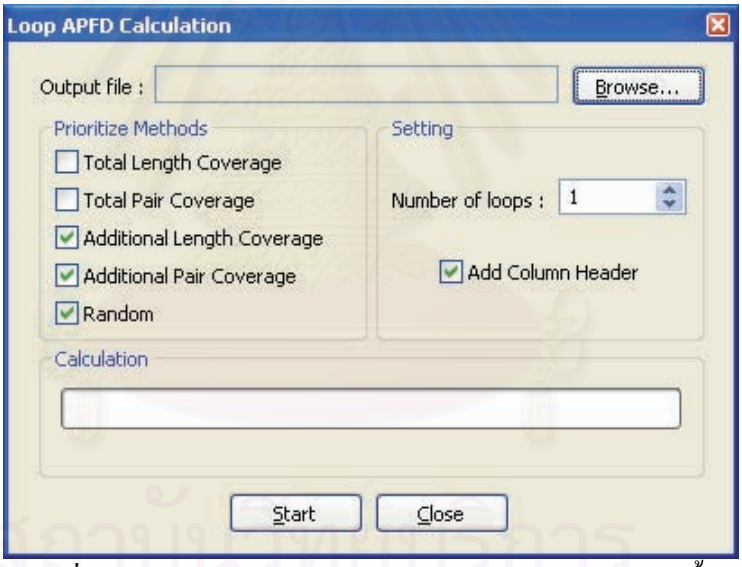
รูปที่ ฉ-3 ส่วนแสดงผลค่า APFD

11. เมื่อจบการทำงานให้กดปุ่ม Close หรือปิดหน้าต่าง

3.2 การทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบต่อเนื่อง

ขั้นตอนใช้เครื่องมือทดสอบวิธีการจัดลำดับความสำคัญกรณีทดสอบแบบต่อเนื่องเพื่อเก็บค่า APFD มีดังนี้

1. เมื่อเริ่มทำงานเครื่องมือจะแสดงส่วนต่อประสานกับผู้ใช้หลัก
2. เลือกแฟ้มข้อมูลชุดทดสอบที่ต้องการจัดลำดับความสำคัญ โดยใช้ปุ่ม Browse…
3. ข้อมูลกรณีทดสอบทั้งหมดในแฟ้มจะแสดงผลในตาราง
4. กรณีที่ข้อมูลกรณีทดสอบไม่มีข้อมูลสถานะตั้งต้นให้เลือก Add Initial State เพื่อสร้างสถานะตั้งต้นจำลอง (Dummy) ให้กรณีทดสอบ
5. กดปุ่ม Loop Calculation เพื่อเริ่มการทดสอบเก็บข้อมูล
6. เครื่องมือจะแสดงผลดังรูป ฉ-4 ให้เลือกแฟ้มที่ใช้เก็บข้อมูลโดยกดปุ่ม Browse… เลือกวิธีการจัดลำดับความสำคัญกรณีทดสอบที่ต้องการทดสอบ โดยเลือกเช็คบ็อกซ์หน้าวิธีการที่ต้องการ และกำหนดจำนวนรอบการทำซ้ำในช่องกรอกข้อความ Number of loops



รูปที่ ฉ-4 ส่วนต่อประสานกราฟิกกับผู้ใช้ของส่วนทดสอบซ้ำ

7. ถ้าต้องการให้บันทึกชื่อวิธีการจัดลำดับความสำคัญกรณีทดสอบเป็นชื่อคอลัมน์ในบรรทัดแรกให้เลือกเช็คบ็อกซ์ Add Header Column
8. เริ่มการทำงานโดยกดปุ่ม Start
9. หากต้องการหยุดระหว่างขั้นตอนการทำงานให้กดปุ่ม Stop
10. เมื่อจบการทำงานกดปุ่ม Close หรือปิดหน้าต่าง

ประวัติผู้เขียนวิทยานิพนธ์

นายพิศักดิ์ คุรุเสถียร เกิดวันที่ 5 กรกฎาคม พ.ศ. 2523 ที่อำเภอโพธาราม จังหวัดราชบุรี สำเร็จการศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปี พ.ศ. 2545 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2546

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย