บทที่ 5

การดำเนินการวิจัยและผลการวิเคราะห์ข้อมูล

ในบทนี้จะกล่าวถึง ผลการวิเคราะห์ข้อมูลที่ได้จากการทดลอง เพื่อนำมาตอบวัตถุประสงค์ของงานวิจัยที่กล่าวมาข้างต้น ซึ่งได้แก่ การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล (Run – Time) ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน (Association Relationship) และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (Generalization Relationship) และการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน (Functional Requirement) ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์เจเนอรัลไลเซชัน ซึ่งจะประกอบด้วยการตรวจสอบการแจกแจงของข้อมูล และการวิเคราะห์ข้อมูลในเชิงสถิติ

5.1 การจัดเตรียมข้อมูลเพื่อวิเคราะห์ผล

หลังจากผู้วิจัยได้กำหนดแผนแบบการทดลอง (Experimental Design) และจัดทำเครื่องมือที่ใช้ในการทดลองและประมวลผล เพื่อตอบวัตถุประสงค์ของงานวิจัยตามที่กล่าวมาข้างต้นเรียบร้อยแล้ว ผู้วิจัยได้ดำเนินการทดลองตามแผนแบบการทดลอง กล่าวคือจัดเก็บข้อมูลจากหน่วยตัวอย่างเพื่อประมวลผล และนำข้อมูลที่ได้มาวิเคราะห์ผล โดยงานวิจัยนี้สามารถวิเคราะห์ข้อมูลคือ (1) เพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล และ (2) เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ ดังนั้น สามารถพิจารณาการจัดเตรียมข้อมูลเพื่อวิเคราะห์ผล ได้เป็น 2 ส่วน ดังนี้

5.1.1 การจัดเตรียมข้อมูลเพื่อวิเคราะห์ผลการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล

จากวัตถุประสงค์ของงานวิจัย การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล สามารถกำหนดแผนแบบการทดลอง และจัดทำเครื่องมือเพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล ซึ่งได้แก่ เครื่องมือที่ใช้สำหรับวัดประสิทธิภาพขณะประมวลผลของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน จากนิยามประสิทธิภาพของโครงสร้างคลาสที่กล่าวถึงในบทที่ 3 ดังนั้นประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์จะเกิดขึ้นจากจำนวนการส่งข้อความหรือเมสเสจ

(Message Calling) ระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ (ชาลี วรกุลพิพัฒน์ และเทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2544) และวัดจากระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ จากนั้นผู้วิจัยได้ดำเนินการทดลองตามแผนแบบการทดลอง กล่าวคือ จัดเตรียมหน่วยตัวอย่างที่เป็นซอฟต์แวร์ธุรกิจที่ออกโดยใช้แนวคิดเชิงวัตถุ (Object – Oriented Paradigm) จำนวนทั้งหมด 5 หน่วยตัวอย่าง ซึ่งประกอบด้วย

หน่วยตัวอย่างที่ 1: ซอฟต์แวร์เอทีเอ็มออนไลน์ (Online ATM System)

หน่วยตัวอย่างที่ 2: ซอฟต์แวร์ช่วยสอนออโตเมตา (Learning Software - Autometa)

หน่วยตัวอย่างที่ 3: ซอฟต์แวร์อนุมัติบัตรเครดิต (Credit Card Approval System)

หน่วยตัวอย่างที่ 4: ซอฟต์แวร์จัดการเรียนการสอน (Short Course Management System)

หน่วยตัวอย่างที่ 5: ซอฟต์แวร์หน้าร้านขายหนังสือ (Vijay Bookshop System Ver. 1.04)

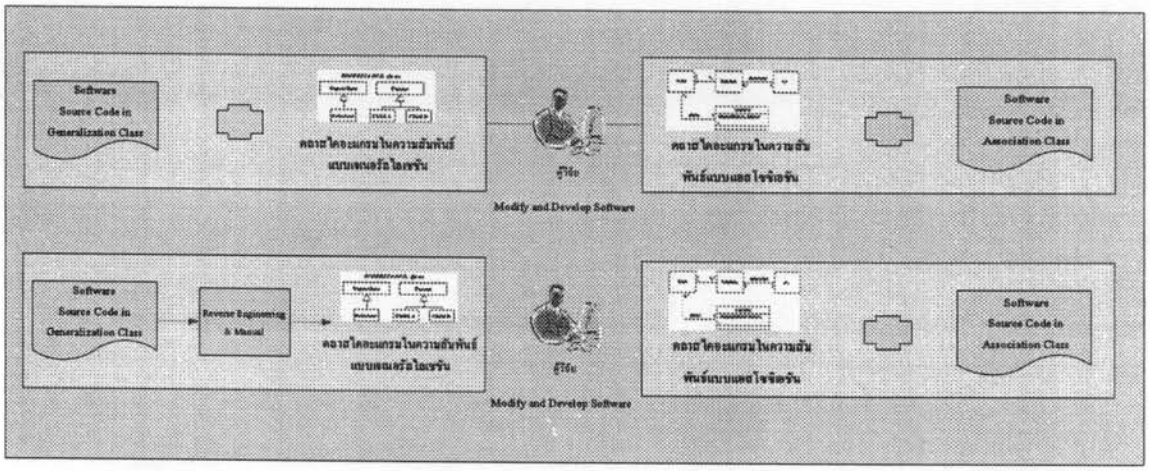
โดยมีข้อกำหนดว่าแต่ละหน่วยตัวอย่างจะต้องมีจำนวนคลาสการทำงานในเชิงธุรกิจ (Business Class Diagram) อย่างน้อย 5 คลาส และเป็นซอฟต์แวร์ที่มีการพัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันที่มีการสืบทอดคุณสมบัติ (Inheritance) จากซูปเปอร์คลาส (Super Class) และมีซับคลาส (Subclass) ที่รับการถ่ายทอดคุณสมบัติจากซูปเปอร์คลาสอย่างน้อยหนึ่งซับคลาสในโครงสร้างคลาสของซอฟต์แวร์ ซึ่งลักษณะของหน่วยตัวอย่างที่ได้ แสดงรายละเอียดได้ดังตารางที่ 5-1

หน่วยตัวอย่าง / ลักษณะของหน่วยตัวอย่าง	หน่วยตัวอย่างที่ 1	หน่วยตัวอย่างที่ 2	หน่วยตัวอย่างที่ 3	หน่วยตัวอย่างที่ 4	หน่วยตัวอย่างที่ 5
โครงสร้างของหน่วยตัวอย่าง	ซอฟต์แวร์ในโครงสร้างคลาสเจเนอรัลไลเซชัน รวมถึงเอกสารแสดงการออกแบบคลาสไดอะแกรม	ซอฟต์แวร์โครงสร้างคลาสเจเนอรัลไลเซชัน รวมถึงเอกสารแสดงการออกแบบคลาสไดอะแกรม	ซอฟต์แวร์ในโครงสร้างคลาสเจเนอรัลไลเซชัน	ซอฟต์แวร์ในโครงสร้างคลาสเจเนอรัลไลเซชัน	ซอฟต์แวร์ในโครงสร้างคลาสเจเนอรัลไลเซชัน
ผู้พัฒนาซอฟต์แวร์	Russell C. B. Professor and Programmer Gordon College	นายศักดิ์ชัย พ่วงปรีชา Programmer	Jeshurun, R. F. Programmer in Philippines	Jackson, T. Programmer in Australia	Sachin, S. T. Programmer in India
แหล่งที่มาของหน่วยตัวอย่าง	http://www.math-cs.gordon.edu	นายศักดิ์ชัย พ่วงปรีชา	http://www.planet-source-code.com	http://www.planet-source-code.com	http://www.programmersheaven.com
จำนวนคลาสการทำงานของซอฟต์แวร์หน่วยตัวอย่าง	22	10	13	11	14
ระดับความลึกของการสืบทอดคุณสมบัติ	2	2	2	2	2
จำนวนซูปเปอร์คลาส	1	1	1	1	1
จำนวนซับคลาส	4	3	1	6	2
จำนวนเมธธอดที่สืบทอดคุณสมบัติ	7	6	10	7	5
จำนวนแอททริบิวต์ที่สืบทอดคุณสมบัติ	16	2	5	8	3

ตารางที่ 5-1 แสดงลักษณะและที่มาของหน่วยตัวอย่างที่ใช้เปรียบเทียบประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์ขณะประมวลผล 5 หน่วยตัวอย่าง

จากตารางที่ 5-1 ผู้วิจัยกำหนดให้ลักษณะของหน่วยตัวอย่างที่ได้มาเป็นซอฟต์แวร์ที่พัฒนาภายใต้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ซึ่งหน่วยตัวอย่างที่ได้มาสามารถแบ่งออกเป็น 2 ลักษณะด้วยกันคือ (1) ซอฟต์แวร์ที่พัฒนาภายใต้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยไม่มีเอกสารแสดงการออกแบบคลาสไดอะแกรม และ (2) ซอฟต์แวร์พัฒนาภายใต้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน รวมถึงเอกสารแสดงการออกแบบคลาสไดอะแกรมของซอฟต์แวร์ในโครงสร้างคลาสความสัมพันธ์แบบเจเนอรัลไลเซชัน

จากนั้น ผู้วิจัยนำหน่วยตัวอย่างมาเปลี่ยนแปลงให้อยู่ในลักษณะของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันทั้ง 5 หน่วยตัวอย่างดังรูปที่ 5-1



รูปที่ 5-1 การเปลี่ยนโครงสร้างคลาสแบบเจเนอรัลไลเซชันเป็นโครงสร้างคลาสแบบแอสโซซิเอชัน

จากรูปที่ 5-1 สามารถอธิบาย ได้ดังนี้

- เมื่อหน่วยตัวอย่างที่ได้อยู่ในลักษณะของซอฟต์แวร์ที่พัฒนาภายใต้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยไม่มีเอกสารแสดงการออกแบบคลาสไดอะแกรม ผู้วิจัยจะรีเวิร์สเอนจิเนียร์ริง (Reverse - Engineering) โดยใช้เครื่องมือวิสชวลแพระไดม์ (Visual Paradigm Tool) เพื่อให้ได้โครงสร้างความสัมพันธ์ของซอฟต์แวร์ในรูปของคลาสไดอะแกรมในความสัมพันธ์แบบเจเนอรัลไลเซชันที่มีความน่าเชื่อถือ จากนั้นออกแบบคลาสไดอะแกรมในความสัมพันธ์แบบแอสโซซิเอชัน ตามหลักการการเปลี่ยนแปลงโครงสร้างคลาส และพัฒนาเป็นซอฟต์แวร์ภายใต้โครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน

- เมื่อหน่วยตัวอย่างที่ได้อยู่ในลักษณะของซอฟต์แวร์พัฒนาภายใต้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน รวมถึงเอกสารแสดงการออกแบบคลาสไดอะแกรมของซอฟต์แวร์ในโครงสร้างคลาสความสัมพันธ์แบบเจเนอรัลไลเซชัน ผู้วิจัยจะนำคลาสไดอะแกรมมาวิเคราะห์และออกแบบคลาสไดอะแกรมในสัมพันธ์แบบแอสโซซิเอชัน ตามหลักการการเปลี่ยนแปลงโครงสร้างคลาส และพัฒนาเป็นซอฟต์แวร์ภายใต้โครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน

จากหลักการเปลี่ยนแปลงโครงสร้างคลาสที่กล่าวมาในบทที่ 2 ผู้วิจัยจะอ้างอิงหลักการเปลี่ยนแปลงโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันให้เป็นโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน ดังนี้

1. กรณีที่ซอฟต์แวร์หน่วยตัวอย่างพัฒนาด้วยโครงคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยที่ซูปเปอร์คลาสเป็นแอบสแทรคท์คลาส (Abstract Class) เปลี่ยนแปลงได้โดย
   - นำเอาความสัมพันธ์ระหว่างคลาสที่มีการสืบทอดคุณสมบัติกันออก
   - แอททริบิวต์และเมธธอดของแอบสแทรคท์คลาสจะนำมาเพิ่มเข้าที่คลาสที่เป็นซับคลาสที่มีการสืบทอดคุณสมบัติ
   - และทุก ๆ แอบสแทรคท์คลาส ในโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน จะถูกนำออกทั้งหมดแล้วเปลี่ยนความสัมพันธ์ระหว่างคลาสให้เป็นโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน
2. กรณีที่ซอฟต์แวร์หน่วยตัวอย่างพัฒนาด้วยโครงคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยที่ซูปเปอร์คลาสเป็นอิมพลีเมนต์คลาส (Implement Class) เปลี่ยนแปลงได้โดย
   - นำเอาความสัมพันธ์ระหว่างคลาสที่มีการสืบทอดคุณสมบัติกันออก
   - นำแอททริบิวต์และเมธธอดของอิมพลีเมนต์คลาสที่มีการถ่ายทอดคุณสมบัติให้ซับคลาส ลงในแต่ละคลาสที่เป็นซับคลาส
   - และให้คงแอททริบิวต์และเมธธอดการทำงานที่ซูปเปอร์คลาส (อิมพลีเมนต์คลาส) กำหนดสิทธิ์ไว้ให้เป็นการทำงานของตัวเองไว้ในคลาส และนำความสัมพันธ์แบบเจเนอรัลไลเซชันออกจากโครงสร้างคลาส แล้วเปลี่ยนความสัมพันธ์ระหว่างคลาสมาเป็นโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน

ซึ่งผู้วิจัยเลือกพิจารณาปรับเปลี่ยนโครงสร้างของคลาสที่เกี่ยวข้องกับคลาสการทำงานของซอฟต์แวร์เท่านั้น โดยไม่ทำกระบวนการปรับเปลี่ยนโครงสร้างคลาสที่มีส่วนติดต่อประสานงานกับผู้ใช้ (User Interface) และคลาสที่เรียกการทำงานของจาวาคลาสไลบรารี (Java Class Libraries) หรือที่เรียกกันว่าแอพพลิเคชันโปรแกรมเมอร์อินเทอร์เฟซ (Application Programmer Interfaces - APIs) เพราะส่วนนี้เป็นส่วนที่โปรแกรมภาษาจาวา (Java Programming Language) และชุดโปรแกรมสำหรับพัฒนาโปรแกรมภาษาจาวา (Integrated Development Environment - IDE) มีบริการไว้ให้เรียกใช้งานอยู่แล้ว (วีระศักดิ์ ซึงถาวร, 2549)

หมายเหตุ เอกสารแสดงรายละเอียดแผนภาพคลาสไดอะแกรมของหน่วยตัวอย่างทั้งโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันและโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน แสดงในเอกสารรายชื่อหน่วยตัวอย่าง ภาคผนวก ก

เมื่อปรับเปลี่ยนเป็นโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันเรียบร้อยแล้ว ผู้วิจัยได้ดำเนินการเปลี่ยนแปลงโครงสร้างคลาสของหน่วยตัวอย่างให้เป็นซอฟต์แวร์ซอร์สโค้ด โดยซอฟต์แวร์ซอร์สโค้ดที่ผู้วิจัยพัฒนาจากหน่วยตัวอย่างนั้นเป็นซอร์สโค้ดประเภทโปรแกรมประยุกต์แบบวินโดว์ (Windows Application Program) เพื่อจัดเก็บข้อมูลจากหน่วยตัวอย่าง โดยหน่วยตัวอย่างที่นำเข้าเครื่องมือนั้น จะต้องถูกแปลงให้เป็นดอทจาร์ (.JAR) ซึ่งทำได้โดยการเอคซพอร์ท (Export) จากชุดโปรแกรมสำหรับพัฒนาภาษาโปรแกรมจาวาอิคลิพซ์ (Eclipse)

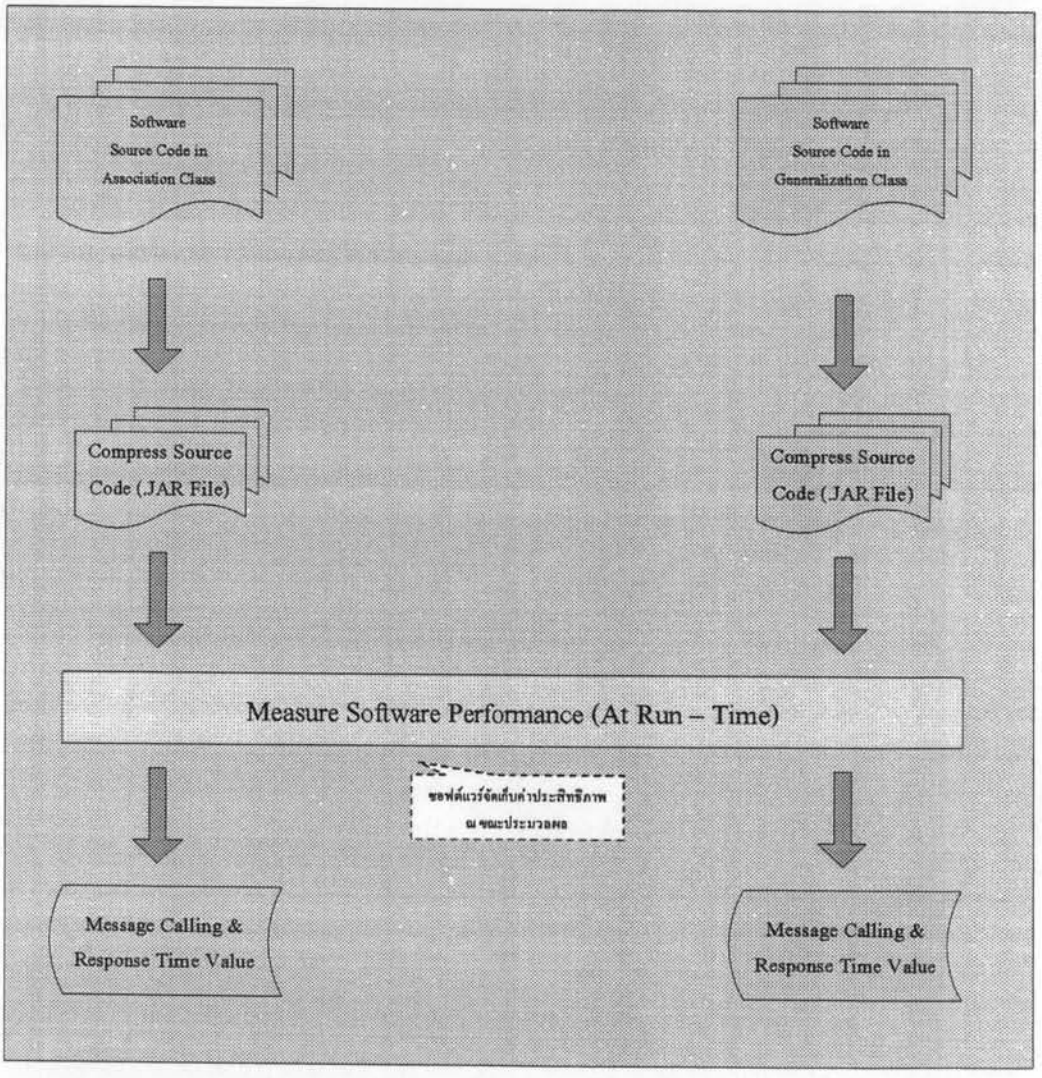
เนื่องจากโปรแกรมภาษาจาวาที่ใช้เป็นหน่วยตัวอย่างประกอบด้วยคลาสจำนวนมาก ดังนั้น เพื่อให้สะดวกแก่การดำเนินการวิจัย ผู้วิจัยจึงจำเป็นต้องบีบอัดไฟล์ดอทคลาส (.CLASS) เป็นไฟล์ดอทจาร์ (.JAR) เพื่อให้หน่วยตัวอย่างมีขนาดเล็กลง และสามารถบีบอัดจำนวนคลาสการทำงานหลาย ๆ คลาสเก็บลงในไฟล์เพียงไฟล์เดียวได้ จึงสะดวกในการจัดการกับคลาสจำนวนมาก และง่ายในการนำส่งข้อมูลเข้าสู่เครื่องมือที่ใช้สำหรับประมวลผลการวิจัย นอกจากนี้ โปรแกรมภาษาจาวายังกำหนดให้คลาสโหลดเดอร์ (Class Loaders) ซึ่งเป็นโปรแกรมจาวาทำหน้าที่ในการโหลดคลาส สามารถโหลดคลาสที่ถูกบีบอัดอยู่ในไฟล์ออกมาใช้งานได้ ทำให้ไม่ต้องขยายไฟล์ดอทจาร์ ออกมาเพื่อจะใช้งานคลาสในไฟล์นั้น (กิตติ ภักดีวัฒนะกุล, 2545)

นำเข้าข้อมูลโดยการโหลดหน่วยตัวอย่างมาผ่านเครื่องมือที่ใช้สำหรับวัดประสิทธิภาพขณะประมวลผล สำหรับช่วยในการอ่านกระบวนการทำงานของซอฟต์แวร์หน่วยตัวอย่าง โดยการประมวลผลซอฟต์แวร์หน่วยตัวอย่างนั้นทำได้โดยการประมวลผลคำสั่งจากแบทไฟล์ (Batch File) ที่รวมคำสั่งการประมวลผลจากไฟล์ดอทจาร์ ดังรูปที่ 5-2

```
java -jar Sample.jar
```

รูปที่ 5-2 คำสั่งแบทไฟล์เพื่อการประมวลผลซอฟต์แวร์หน่วยตัวอย่าง

จากนั้น จัดเก็บรวบรวมข้อมูลตามขั้นตอนในการจัดเก็บค่าประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์ขณะประมวลผลแสดงได้ดังรูปที่ 5-3 และนำข้อมูลที่ได้มาวิเคราะห์ผลเชิงสถิติเพื่อตอบวัตถุประสงค์ที่กำหนดไว้ข้างต้น



รูปที่ 5-3 การจัดเก็บค่าประสิทธิภาพของโครงสร้างคลาสขณะประมวลผล

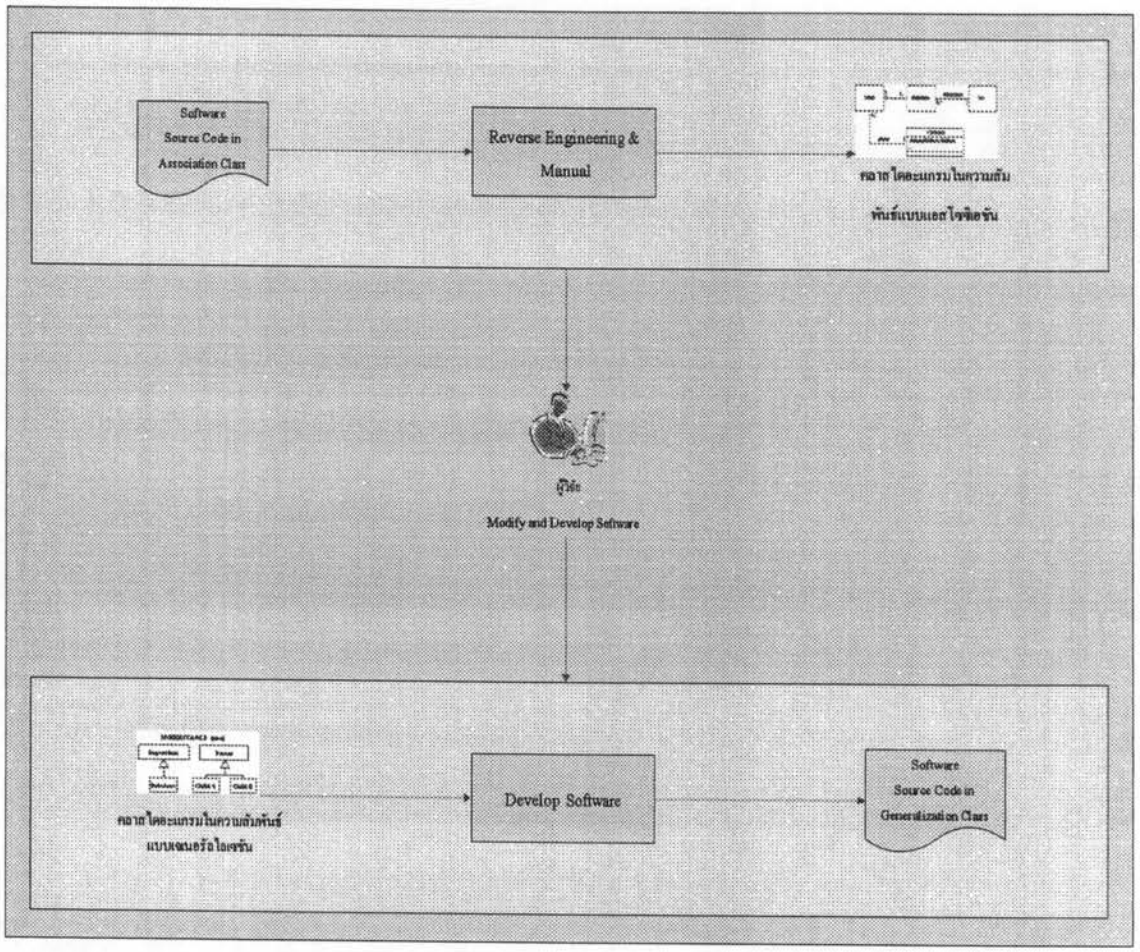
5.1.2 การจัดเตรียมข้อมูลเพื่อวิเคราะห์ผลการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์

จากวัตถุประสงค์ของงานวิจัย การเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน (Functional Requirement) สามารถกำหนดแผนแบบการทดลอง และจัดทำเครื่องมือเพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยคำนวณจากจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ความต้องการเชิงฟังก์ชัน จากนั้นผู้วิจัยได้ดำเนินการทดลองตามแผนแบบการทดลอง กล่าวคือ จัดเตรียมซอฟต์แวร์กรณีศึกษา (Software Case Study) ซึ่งกำหนดให้เป็นซอฟต์แวร์ระบบธุรกิจที่พัฒนาด้วยแนวคิดเชิงวัตถุ โดยมีข้อกำหนดว่าจะต้องมีจำนวนคลาสการทำงานในเชิงธุรกิจ (Business Class Diagram) อย่างน้อย 5 คลาส ผู้วิจัยได้เลือกระบบการจัดการการสายการบินและการบริหารการบิน (Airway and Flight Management System) ขึ้นมาเป็นซอฟต์แวร์กรณีศึกษา ซึ่งเป็นซอฟต์แวร์ประเภทเว็บแอพพลิเคชัน (Web Application Programming Language)

โดยซอฟต์แวร์กรณีศึกษาที่นำมาใช้ในงานวิจัยนั้น มีลักษณะการทำงานในเชิงธุรกิจ (Business Process) โดยรวม ดังนี้

- การดำเนินการแบบฟรอนต์เอนด์ (Front End) โดยมีการให้บริการอำนวยความสะดวกแก่ลูกค้าในการจัดการเกี่ยวกับตั๋วโดยสาร เช่น การจองตั๋วโดยสารออนไลน์ การเลื่อนเที่ยวบินโดยสาร การเช็คอินตั๋วโดยสาร เป็นต้น
- การบริหารการบิน เช่น การกำหนดเที่ยวบินโดยสาร เที่ยวบินลำเลียงสินค้า แสดงเที่ยวบินในการบินแต่ละวัน เป็นต้น

ซึ่งซอฟต์แวร์กรณีศึกษาที่ได้มาอยู่ในรูปของซอฟต์แวร์ที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน ดังนั้นผู้วิจัยได้ทำรีเวิร์สเอนจิเนียร์ริง (Reverse - Engineering) โดยใช้เครื่องมือวิสชวลแพระไดม์ (Visual Paradigm Tool) เพื่อให้ได้โครงสร้างความสัมพันธ์ของซอฟต์แวร์ในรูปของคลาสไดอะแกรมในความสัมพันธ์แบบแอสโซซิเอชันที่มีความน่าเชื่อถือ จากนั้นนำคลาสไดอะแกรมในความสัมพันธ์แบบแอสโซซิเอชันมาวิเคราะห์และเปลี่ยนแปลงให้เป็นคลาสไดอะแกรมในความสัมพันธ์แบบเจเนอรัลไลเซชันตามหลักการเปลี่ยนแปลงโครงสร้างคลาสที่ได้กล่าวมาแล้วในบทที่ 2 แล้วผู้วิจัยนำคลาสไดอะแกรมในความสัมพันธ์แบบเจเนอรัลไลเซชันมาพัฒนาเป็นซอฟต์แวร์ซอร์สโค้ดที่พัฒนาภายโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ดังรูปที่ 5-4



รูปที่ 5-4 การเปลี่ยนแปลงโครงสร้างคลาสของซอฟต์แวร์กรณีศึกษาให้อยู่ในโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและเจเนอรัลไลเซชัน

ซึ่งงานวิจัยนี้ต้องการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน เพื่อพิจารณาจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการ ดังนั้นหน่วยตัวอย่างที่ใช้ในงานวิจัยนี้ได้แก่ ความต้องการเชิงฟังก์ชัน จำนวน 30 หน่วยตัวอย่าง

โดยหน่วยตัวอย่างที่เป็นความต้องการเชิงฟังก์ชันทั้ง 30 หน่วยตัวอย่าง มาจากการที่สายการบินต้องการเพิ่มการดำเนินการในส่วนของแบ็กเอนด์ (Back End) ซึ่งเป็นความต้องการเชิงฟังก์ชันที่เป็นความต้องการของซอฟต์แวร์กรณีศึกษาที่ต้องการให้ฟังก์ชันการดำเนินงานสามารถใช้งานจริงในระบบธุรกิจสายการบินทั้ง 30 หน่วยตัวอย่าง ซึ่งจะครอบคลุมการทำงาน ดังนี้

- การจัดการเกี่ยวกับสินค้า (Cargo Application) เช่น การโหลดสินค้าขึ้นเครื่องบิน การคิดอัตราภาษีสินค้า แสดงสถานะของสินค้า การออกใบรายการสินค้า เป็นต้น

- การจัดการเกี่ยวกับการบัญชี (Account Application) เช่น การให้เครดิตผู้โดยสาร การจัดการเกี่ยวกับการเงิน เป็นต้น
- การบริหารการบิน (Flight Application) เช่น การตรวจสอบสถานการณ์บิน เป็นต้น
- การจัดการเกี่ยวกับตัวแทนจำหน่ายตั๋วเครื่องบินโดยสาร การจัดการพนักงาน

ซึ่งรายละเอียดของความต้องการเชิงฟังก์ชันทั้ง 30 หน่วยตัวอย่าง แสดงได้ดังตารางที่ 5-2

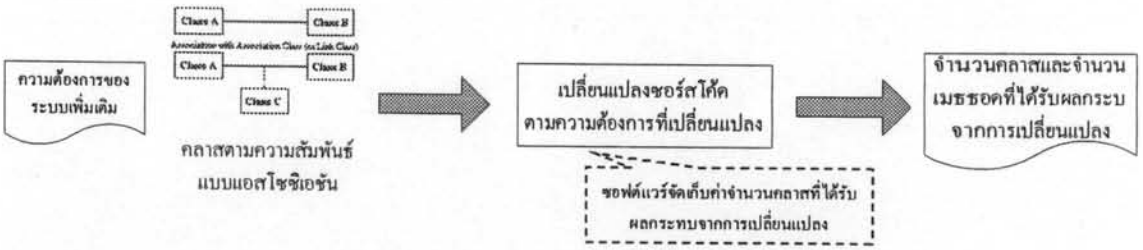
ตารางที่ 5-2 ความต้องการเชิงฟังก์ชันของซอฟต์แวร์กรณีศึกษาที่ใช้เป็นหน่วยตัวอย่าง

ลำดับความต้องการของซอฟต์แวร์	ความต้องการเชิงฟังก์ชัน (Functional Requirement) ที่เป็นหน่วยตัวอย่างของซอฟต์แวร์กรณีศึกษา
1	Travel Agents Definition
2	Locate Agents Definition
3	Sign Up A New Frequent Flyer Account
4	Define New Air Waybill
5	Search For Air Waybill
6	Sign Up New Customer Account For Cargo
7	Search/Edit Customer Account
8	Cargo Manifest
9	Load Cargo on Flight
10	Cargo Flight Release
11	Delivery Air Waybill
12	Print Delivery Order
13	Flight Status View/Update
14	Define Blocked Flights / Holidays
15	Weight & Balance Report
16	Move passengers between flights
17	List Schedule
18	Add/Change Reservation Panel Routes
19	Aircraft Definition
20	City Code Definition

ตารางที่ 5-2 ความต้องการเชิงฟังก์ชันของซอฟต์แวร์กรณีศึกษาที่ใช้เป็นหน่วยตัวอย่าง (ต่อ)

ลำดับความต้องการของซอฟต์แวร์	ความต้องการเชิงฟังก์ชัน (Functional Requirement) ที่เป็นหน่วยตัวอย่างของซอฟต์แวร์กรณีศึกษา
21	Leg Definition
22	Flight Definition
23	Update Flight Schedule
24	Accounting Reports
25	Edit Passenger Credit
26	Cash Drawer Report
27	Cargo Invoice Tool
28	Update Travel Agent Cash Payment As Paid
29	Refunded Report
30	Define /Update User Group

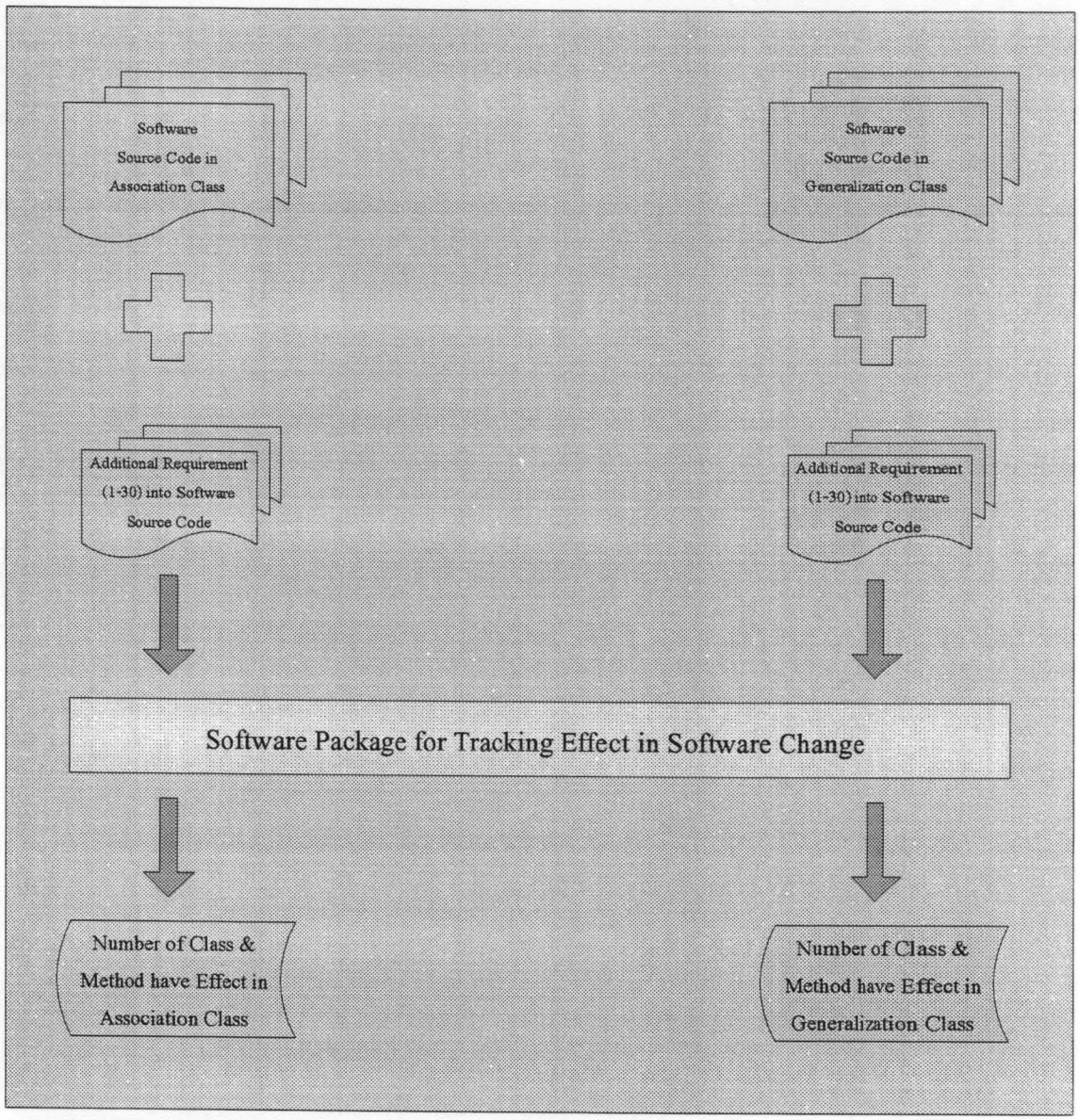
เมื่อเรียบร้อยแล้วผู้วิจัยได้แก้ไขโครงสร้างของคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างของคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยการออกแบบความต้องการเชิงฟังก์ชันที่เป็นหน่วยตัวอย่างเพิ่มเข้ามาในซอฟต์แวร์กรณีศึกษา ดังรูปที่ 5-5



รูปที่ 5-5 การเพิ่มหน่วยตัวอย่างเพื่อจัดเก็บคลาสและเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์

รูปที่ 5-5 การเพิ่มหน่วยตัวอย่างเพื่อจัดเก็บคลาสและเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์ (ต่อ)

จากนั้น จัดเก็บข้อมูลจากหน่วยตัวอย่าง ด้วยการนำหน่วยตัวอย่างมาผ่านเครื่องมือที่ใช้สำหรับวัดจำนวนคลาสและจำนวนเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ความต้องการเชิงฟังก์ชัน และคำนวณหาจำนวนคลาสและจำนวนเมธอดที่ถูกกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์แล้วดำเนินการจัดเก็บรวบรวมข้อมูล ดังรูปที่ 5-6 และนำข้อมูลที่ได้มาวิเคราะห์ผลเชิงสถิติ เพื่อตอบวัตถุประสงค์ที่กำหนดไว้ข้างต้น



รูปที่ 5-6 การจัดเก็บจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชัน

5.2 ผลการวิเคราะห์ข้อมูล

เนื่องด้วยงานวิจัยนี้เป็นการวิจัยเชิงทดลอง (Experimental Research) เมื่อผู้วิจัยได้ดำเนินการตามแผนแบบการทดลองที่กำหนดไว้และจัดเก็บข้อมูลจากหน่วยตัวอย่างเรียบร้อยแล้ว ผู้วิจัยได้รวบรวมข้อมูลที่ได้จากการทดลองเพื่อนำมาวิเคราะห์ข้อมูล โดยงานวิจัยนี้สามารถแยกการวิเคราะห์ข้อมูลออกได้เป็น 2 ส่วน คือ (1) การวิเคราะห์ข้อมูลเพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล และ (2) การวิเคราะห์ข้อมูลเพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ ซึ่งสามารถอธิบายรายละเอียดได้ดังนี้

5.2.1 ผลการวิเคราะห์ข้อมูลเพื่อการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล

จากแบบแผนการทดลองที่กำหนดให้นำหน่วยตัวอย่างมาผ่านเครื่องมือที่ใช้สำหรับวัดประสิทธิภาพขณะประมวลผล สำหรับช่วยในการอ่านกระบวนการทำงานของซอฟต์แวร์หน่วยตัวอย่างและหาค่าประสิทธิภาพของซอฟต์แวร์ขณะประมวลผลอย่างอัตโนมัติ เพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผลระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยมีหน่วยตัวอย่างที่ใช้ทำการทดลองจำนวน 5 หน่วยตัวอย่าง ดังที่กล่าวมาแล้วในหัวข้อที่ 5.1.1 ซึ่งเมื่อเปลี่ยนแปลงโครงสร้างคลาสของหน่วยตัวอย่าง แต่ละซอฟต์แวร์หน่วยตัวอย่างจะมีจำนวนคลาสการทำงานทางธุรกิจในโครงสร้างความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ดังตารางที่ 5-3

ตารางที่ 5-3 จำนวนคลาสในการทำงานทางธุรกิจของซอฟต์แวร์หน่วยตัวอย่างในโครงสร้างคลาสความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

รายชื่อหน่วยตัวอย่าง	จำนวนคลาสการทำงานทางธุรกิจของโครงสร้างคลาสในความสัมพันธ์แอสโซซิเอชัน	จำนวนคลาสการทำงานทางธุรกิจของโครงสร้างคลาสในความสัมพันธ์เจเนอรัลไลเซชัน
หน่วยตัวอย่างที่ 1	22	22
หน่วยตัวอย่างที่ 2	9	10
หน่วยตัวอย่างที่ 3	12	13
หน่วยตัวอย่างที่ 4	11	11
หน่วยตัวอย่างที่ 5	13	14

ซึ่งการจัดเก็บข้อมูลสำหรับงานวิจัยนี้ ดำเนินการโดยซอฟต์แวร์หน่วยตัวอย่างแต่ละหน่วย ผู้วิจัยจะประมวลผลทรานแซคชันจำนวน 3 แบบ ซึ่งทรานแซคชันแต่ละแบบมีลักษณะของการทำรายการ ดังนี้

- **ทรานแซคชันแบบที่ 1** คือ การทำรายการซอฟต์แวร์หน่วยตัวอย่างโดยยังไม่มีการเปลี่ยนแปลงข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธธอด) ภายในโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (ซูปเปอร์คลาสและซับคลาสที่มีการสืบทอดหรือถ่ายทอดคุณสมบัติกัน)
- **ทรานแซคชันแบบที่ 2** คือ การทำรายการซอฟต์แวร์หน่วยตัวอย่างเพื่อให้มีการเปลี่ยนแปลงข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธธอด) ภายในคลาส โดยกำหนดให้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลเซชันมีการเปลี่ยนแปลงข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธธอด) ที่กระทบกับซับคลาสใดซับคลาสหนึ่งที่มีการสืบทอดคุณสมบัติในโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลเซชัน
- **ทรานแซคชันแบบที่ 3** คือ การทำรายการซอฟต์แวร์หน่วยตัวอย่างเพื่อให้มีการเปลี่ยนแปลงข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธธอด) ภายในคลาส โดยกำหนดให้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลเซชันมีการเปลี่ยนแปลงข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธธอด) ที่กระทบกับหลาย ๆ ซับคลาสที่มีการสืบทอดคุณสมบัติในโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลเซชัน

เพื่อต้องการตรวจสอบว่าระดับความลึกของการสืบทอดคุณสมบัติมีผลกับประสิทธิภาพของซอฟต์แวร์หรือไม่

หมายเหตุ การประมวลผลทรานแซคชันในแต่ละแบบของซอฟต์แวร์หน่วยตัวอย่างทั้ง 5 หน่วยตัวอย่าง แสดงรายละเอียดในภาคผนวก ข

เมื่อกำหนดให้หน่วยตัวอย่างทั้งหมดผ่านเครื่องมือสำหรับวัดประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์ขณะประมวลผล เพื่อจัดเก็บข้อมูลจำนวนการส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์หน่วยตัวอย่าง ซึ่งหมายรวมถึงคลาสที่มีส่วนติดต่อประสานงานกับผู้ใช้และคลาสที่เรียกการทำงานของจาวาคลาสไลบรารีด้วย ดังนั้นเมื่อประมวลผลและจัดเก็บข้อมูลจากหน่วยตัวอย่างทุกหน่วยเรียบร้อยแล้ว สามารถแสดงผลลัพธ์ของจำนวนการส่งข้อความหรือเมสเสจ นอกจากนี้ยังได้ผลลัพธ์จากการจัดเก็บประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์โดยวัดจากระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ ได้ดังตารางที่ 5-4 และตารางที่ 5-5

ตารางที่ 5-4 จำนวนการส่งข้อความหรือเมสเสจขณะประมวลผลและระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสแบบแอสโซซิเอชัน (P_x) และ (T_x)

รายชื่อหน่วยตัวอย่าง	จำนวนการส่งข้อความ / เมสเสจ (Number of Message Calling) : P_x	ระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ (Response Time - Msec) : T_x
หน่วยตัวอย่างที่ 1 : ซอฟต์แวร์เอทีเอ็มออนไลน์		
ทรานแซคชันที่ 1	160	19,426
ทรานแซคชันที่ 2	414	126,113
ทรานแซคชันที่ 3	1,578	640,552
หน่วยตัวอย่างที่ 2 : ซอฟต์แวร์ช่วยสอนออโตเมตา		
ทรานแซคชันที่ 1	72	3,557
ทรานแซคชันที่ 2	999	1,535
ทรานแซคชันที่ 3	5,596	6,065
หน่วยตัวอย่างที่ 3 : ซอฟต์แวร์อนุมัติบัตรเครดิต		
ทรานแซคชันที่ 1	459	14,061
ทรานแซคชันที่ 2	939	27,812
ทรานแซคชันที่ 3	1,904	54,918
หน่วยตัวอย่างที่ 4 : ซอฟต์แวร์จัดการเรียนการสอน		
ทรานแซคชันที่ 1	15	600
ทรานแซคชันที่ 2	33	1,000
ทรานแซคชันที่ 3	82	1,593
หน่วยตัวอย่างที่ 5 : ซอฟต์แวร์หน้าร้านขายหนังสือ		
ทรานแซคชันที่ 1	115	23,920
ทรานแซคชันที่ 2	577	10,078
ทรานแซคชันที่ 3	631	571,779

ตารางที่ 5-5 จำนวนการส่งข้อความหรือเมสเสจขณะประมวลผลและระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสแบบเจเนอรัลไลเซชัน (P_y) และ (T_y)

รายชื่อหน่วยตัวอย่าง	จำนวนการส่งข้อความ / เมสเสจ (Number of Message Calling) : P_y	ระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ (Response Time - Msec) : T_y
หน่วยตัวอย่างที่ 1 : ซอฟต์แวร์เอทีเอ็มออนไลน์		
ทรานแซคชันที่ 1	163	22,440
ทรานแซคชันที่ 2	415	147,617
ทรานแซคชันที่ 3	1,584	712,258
หน่วยตัวอย่างที่ 2 : ซอฟต์แวร์ช่วยสอนออโตเมตา		
ทรานแซคชันที่ 1	74	5,599
ทรานแซคชันที่ 2	1,014	1,650
ทรานแซคชันที่ 3	5,648	6,113
หน่วยตัวอย่างที่ 3 : ซอฟต์แวร์อนุมัติบัตรเครดิต		
ทรานแซคชันที่ 1	474	16,916
ทรานแซคชันที่ 2	988	31,186
ทรานแซคชันที่ 3	2,029	62,082
หน่วยตัวอย่างที่ 4 : ซอฟต์แวร์จัดการเรียนการสอน		
ทรานแซคชันที่ 1	15	610
ทรานแซคชันที่ 2	36	1,200
ทรานแซคชันที่ 3	86	1,774
หน่วยตัวอย่างที่ 5 : ซอฟต์แวร์หน้าร้านขายหนังสือ		
ทรานแซคชันที่ 1	124	67,766
ทรานแซคชันที่ 2	585	11,075
ทรานแซคชันที่ 3	674	727,828

จากตารางที่ 5-4 และตารางที่ 5-5 สามารถนำผลลัพธ์ที่ได้จากเครื่องมือมาหาค่าความแตกต่างของจำนวนการส่งข้อความหรือเมสเสจขณะประมวลผลระหว่างโครงสร้างคลาสแอสโซซิเอชัน (P_x) และโครงสร้างคลาสเจเนอรัลไลเซชัน (P_y) ได้ดังตารางที่ 5-6 และหาค่าความแตกต่างของระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ระหว่างโครงสร้างคลาสแอสโซซิเอชัน (T_x) และโครงสร้างคลาสเจเนอรัลไลเซชัน (T_y) ได้ดังตารางที่ 5-7

ตารางที่ 5-6 แสดงความแตกต่างของจำนวนการส่งข้อความ / เมสเสจขณะประมวลผลระหว่างโครงสร้างคลาสแอสโซซิเอชัน (P_x) และโครงสร้างคลาสเจเนอรัลไลเซชัน (P_y)

รายชื่อหน่วยตัวอย่าง	จำนวนการส่งข้อความ / เมสเสจ (Number of Message Calling) : P_y	จำนวนการส่งข้อความ / เมสเสจ (Number of Message Calling) : P_x	ค่าความแตกต่างของจำนวนการส่งข้อความ / เมสเสจระหว่าง (P_y - P_x)
หน่วยตัวอย่างที่ 1 : ซอฟต์แวร์เอทีเอ็มออนไลน์			
ทรานแซคชันที่ 1	163	160	3
ทรานแซคชันที่ 2	415	414	1
ทรานแซคชันที่ 3	1,584	1,578	6
หน่วยตัวอย่างที่ 2 : ซอฟต์แวร์ช่วยสอนออโตเมตา			
ทรานแซคชันที่ 1	74	72	2
ทรานแซคชันที่ 2	1,014	999	15
ทรานแซคชันที่ 3	5,648	5,596	52
หน่วยตัวอย่างที่ 3 : ซอฟต์แวร์อนุมัติบัตรเครดิต			
ทรานแซคชันที่ 1	474	459	15
ทรานแซคชันที่ 2	988	939	49
ทรานแซคชันที่ 3	2,029	1,904	125
หน่วยตัวอย่างที่ 4 : ซอฟต์แวร์จัดการเรียนการสอน			
ทรานแซคชันที่ 1	15	15	0
ทรานแซคชันที่ 2	36	33	3
ทรานแซคชันที่ 3	86	82	4

ตารางที่ 5-6 แสดงความแตกต่างของจำนวนการส่งข้อความ / เมสเสจขณะประมวลผลระหว่างโครงสร้างคลาสแอสโซซิเอชัน (P_x) และโครงสร้างคลาสเจเนอรัลไลเซชัน (P_y) (ต่อ)

รายชื่อหน่วยตัวอย่าง	จำนวนการส่งข้อความ / เมสเสจ (Number of Message Calling) : P_y	จำนวนการส่งข้อความ / เมสเสจ (Number of Message Calling) : P_x	ค่าความแตกต่างของจำนวนการส่งข้อความ / เมสเสจระหว่าง ($P_y - P_x$)
หน่วยตัวอย่างที่ 5 : ซอฟต์แวร์หน้าร้านขายหนังสือ			
ทรานแซคชันที่ 1	124	115	9
ทรานแซคชันที่ 2	585	577	8
ทรานแซคชันที่ 3	674	631	43

จากตารางที่ 5-6 ความแตกต่างของจำนวนการส่งข้อความหรือเมสเสจขณะประมวลผลซอฟต์แวร์ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยวัดจากจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์หน่วยตัวอย่าง พบว่าจำนวนผลความแตกต่างระหว่างโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีค่ามากกว่าโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันในทุก ๆ หน่วยตัวอย่าง

ตารางที่ 5-7 แสดงความแตกต่างของระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ระหว่างโครงสร้างคลาสแอสโซซิเอชัน (T_x) และโครงสร้างคลาสเจเนอรัลไลเซชัน (T_y)

รายชื่อหน่วยตัวอย่าง	ระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ (Response Time - Msec) : T_y	ระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ (Response Time - Msec) : T_x	ค่าความแตกต่างของระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ระหว่าง (T_y - T_x)
หน่วยตัวอย่างที่ 1 : ซอฟต์แวร์เอทีเอ็มออนไลน์			
ทรานแซคชันที่ 1	22,440	19,426	3,014
ทรานแซคชันที่ 2	147,617	126,113	21,504
ทรานแซคชันที่ 3	712,258	640,552	71,706
หน่วยตัวอย่างที่ 2 : ซอฟต์แวร์ช่วยสอนออโตเมตา			
ทรานแซคชันที่ 1	5,599	3,557	2,042
ทรานแซคชันที่ 2	1,650	1,535	115
ทรานแซคชันที่ 3	6,113	6,065	48
หน่วยตัวอย่างที่ 3 : ซอฟต์แวร์อนุมัติบัตรเครดิต			
ทรานแซคชันที่ 1	16,916	14,061	2,855
ทรานแซคชันที่ 2	31,186	27,812	3,374
ทรานแซคชันที่ 3	62,082	54,918	7,164
หน่วยตัวอย่างที่ 4 : ซอฟต์แวร์จัดการเรียนการสอน			
ทรานแซคชันที่ 1	610	600	10
ทรานแซคชันที่ 2	1,200	1,000	200
ทรานแซคชันที่ 3	1,774	1,593	181
หน่วยตัวอย่างที่ 5 : ซอฟต์แวร์หน้าร้านขายหนังสือ			
ทรานแซคชันที่ 1	67,766	23,920	43,846
ทรานแซคชันที่ 2	11,075	10,078	997
ทรานแซคชันที่ 3	727,828	571,779	156,049

จากตารางที่ 5-7 ความแตกต่างของของระยะเวลาที่หน่วยตัวอย่างถูกประมวลผลโดยสมบูรณ์ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยวัดจากระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ พบว่าจำนวนผลความแตกต่างระหว่างโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีค่ามากกว่าโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันในทุก ๆ หน่วยตัวอย่าง

5.2.2 การตรวจสอบการแจกแจงข้อมูลและทดสอบสมมติฐานเพื่อการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล

จากตารางที่ 5-6 และตารางที่ 5-7 ผู้วิจัยดำเนินการตรวจสอบการแจกแจงของข้อมูลจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์และระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ที่ได้ว่าเป็นการแจกแจงปกติหรือไม่ ถ้าข้อมูลมีการแจกแจงแบบปกติผู้วิจัยจะทดสอบสมมติฐานแบบอิงพารามิเตอร์ (Parametric Test) แต่ถ้าข้อมูลไม่มีการแจกแจงแบบปกติ ผู้วิจัยจะทดสอบสมมติฐานแบบไม่อิงกับพารามิเตอร์ (Non Parametric Test) (ศิริชัย พงษ์วิชัย, 2546) ดังนั้น ผู้วิจัยได้กำหนดสมมติฐานสำหรับการทดสอบการแจกแจงข้อมูลได้ดังนี้

1. H_0: ความแตกต่างของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติ

   H_1: ความแตกต่างของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

2. H_0: ความแตกต่างของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติ

   H_1: ความแตกต่างของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและ

โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

ในการตรวจสอบการแจกแจงของข้อมูลเชิงปริมาณว่าเป็นการแจกแจงแบบปกติโดยใช้สถิติทดสอบนั้น มีสถิติที่ใช้ทดสอบ คือ Kolmogorov-Smirnov สำหรับหน่วยตัวอย่างที่มากกว่า 50 หน่วยตัวอย่าง และ Shapiro-Wilk สำหรับหน่วยตัวอย่างที่น้อยกว่า 50 หน่วยตัวอย่าง (ศิริชัย พงษ์วิชัย, 2546) สำหรับงานวิจัยนี้มีหน่วยตัวอย่างที่ใช้ในงานวิจัยจำนวน 15 หน่วยตัวอย่าง ซึ่งน้อยกว่า 50 หน่วยตัวอย่าง ดังนั้น จึงใช้เทคนิค Shapiro-Wilk ในการตรวจสอบการแจกแจงข้อมูล โดยจะเลือกทดสอบสมมติฐานแบบอิงพารามิเตอร์ (Parametric Test) เมื่อค่า Sig. (Significance) ของการทดสอบมีค่ามากกว่าหรือเท่ากับระดับนัยสำคัญที่กำหนด โดยงานวิจัยนี้กำหนดระดับนัยสำคัญเท่ากับ 0.05 ให้ยอมรับสมมติฐาน H_0 ซึ่งถือว่าข้อมูลมีการแจกแจงแบบปกติ ดังนั้นจึงทดสอบสมมติฐานแบบอิงพารามิเตอร์ ซึ่งใช้วิธีการทดสอบสมมติฐานเกี่ยวกับผลต่างระหว่างค่าเฉลี่ย 2 ประชากรแบบจับคู่ (Paired t-test) เนื่องจากการทดสอบสมมติฐานนี้เป็นการหาค่าความแตกต่างของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน และการหาค่าความแตกต่างของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

ในทางตรงกันข้ามจะเลือกทดสอบสมมติฐานแบบไม่อิงพารามิเตอร์ (Non Parametric Test) เมื่อมีการปฏิเสธสมมติฐานที่ H_0 โดยที่ค่า Sig. ของการทดสอบมีค่าน้อยกว่าระดับนัยสำคัญที่กำหนด โดยงานวิจัยนี้กำหนดระดับนัยสำคัญเท่ากับ 0.05 ดังนั้น จึงทดสอบสมมติฐานแบบไม่อิงพารามิเตอร์ ผู้วิจัยเลือกใช้วิธีการทดสอบสมมติฐานเกี่ยวกับการแจกแจงของข้อมูล 2 ชุดที่สัมพันธ์กัน (Two-Related-Samples Tests) โดยเลือกใช้วิธีการทดสอบประเภทวิลคอกสัน ไซน์-แรงค์ เทส (Wilcoxon signed-rank test) เนื่องจากการทดสอบสมมติฐานนี้เป็นการหาค่าความแตกต่างของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน และการหาค่าความแตกต่างของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (ศิริชัย พงษ์วิชัย, 2546)

จากการทดสอบการแจกแจงข้อมูลสามารถแสดงรายละเอียดของข้อมูลได้ดังนี้

1. ข้อมูลสำหรับการทดสอบสมมติฐานที่ 1 : ความแตกต่างของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติหรือไม่ ดังตารางที่ 5-8

ตารางที่ 5-8 แสดงการทดสอบทางสถิติตามผลการทดลองว่าจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์มีการแจกแจงแบบปกติหรือไม่

	Shapiro-Wilk		
	Statistic	df	Sig.
Different of Message (Generalization and Association)	.675	15	.000

a Lilliefors Significance Correction

จากตารางที่ 5-8 จะเห็นได้ว่าค่า Sig. ของการทดสอบน้อยกว่าระดับนัยสำคัญที่กำหนด ดังนั้นจึงจำเป็นต้องปฏิเสธสมมติฐาน H_0 ดังนั้น ค่าความแตกต่างของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

2. ข้อมูลสำหรับการทดสอบสมมติฐานที่ 2 : ความแตกต่างของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติหรือไม่ ดังตารางที่ ตารางที่ 5-9

ตารางที่ 5-9 แสดงการทดสอบทางสถิติตามผลการทดลองว่าระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์มีการแจกแจงแบบปกติหรือไม่

	Shapiro-Wilk		
	Statistic	df	Sig.
Different of Response Time (Generalization and Association)	.568	15	.000

a Lilliefors Significance Correction

จากตารางที่ 5-9 จะเห็นได้ว่าค่า Sig. ของการทดสอบน้อยกว่าระดับนัยสำคัญที่กำหนด ดังนั้นจึงจำเป็นต้องปฏิเสธสมมติฐาน H_0 ดังนั้น ค่าความแตกต่างของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

จากข้อมูลที่เก็บรวบรวมได้จากการทำการทดลอง ผู้วิจัยต้องการเปรียบเทียบลักษณะของโครงสร้างคลาสที่แตกต่างกันมีผลต่อประสิทธิภาพของซอฟต์แวร์ขณะประมวลผลอย่างไร

กำหนดให้ P_x คือ ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันของซอฟต์แวร์ขณะประมวลผล โดยวัดจากจำนวนการส่งข้อความหรือเมสเสจ

P_y คือ ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันของซอฟต์แวร์ขณะประมวลผล โดยวัดจากจำนวนการส่งข้อความหรือเมสเสจ

T_x คือ ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันของซอฟต์แวร์ขณะประมวลผล โดยวัดจากระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์

T_y คือ ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันของซอฟต์แวร์ขณะประมวลผล โดยวัดจากระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์

จากตัวแปรที่ผู้วิจัยกำหนด สามารถนำมาตั้งสมมติฐานเพื่อหาความสัมพันธ์ระหว่างตัวแปรเพื่อนำมาตอบวัตถุประสงค์ของงานวิจัยภายใต้ขอบเขตของปัญหาที่กำหนด คือ การเปรียบเทียบ

ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ได้ดังนี้

1. $H_0 : P_x$ และ P_y มีค่าไม่แตกต่างกัน
   $H_1 : P_x$ และ P_y มีค่าแตกต่างกัน
2. $H_0 : T_x$ และ T_y มีค่าไม่แตกต่างกัน
   $H_1 : T_x$ และ T_y มีค่าแตกต่างกัน

ซึ่งจากการตรวจสอบการแจกแจงข้อมูล พบว่าจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์และระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ที่ได้ว่าเป็นไม่มีการแจกแจงแบบปกติ ผู้วิจัยจึงเลือกการทดสอบสมมติฐานเกี่ยวกับการแจกแจงของข้อมูล 2 ชุดที่สัมพันธ์กัน (Two-Related-Samples Tests) โดยจะใช้โปรแกรม SPSS เวอร์ชัน 14 ในการทดสอบข้อมูล

1 การทดสอบสมมติฐานของจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสที่แตกต่างกัน

เนื่องจากการตรวจสอบการแจกแจงข้อมูลไม่มีการแจกแจงแบบปกติ สำหรับสมมติฐานนี้เป็นการกำหนดสมมติฐานแบบสองทาง เมื่อนำข้อมูลที่ได้จากการทดลองมาทดสอบทางสถิติ โดยเลือกใช้วิธีการทดสอบประเภทวิลคอกสัน ไซน์-แรงค์ เทส โดยผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ ดังตารางที่ 5-10

ตารางที่ 5-10 ผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ของจำนวนการรับส่งข้อความหรือเมสเสจ

Wilcoxon Signed Ranks Test

		N	Mean Rank	Sum of Ranks
Px - Py	Negative Ranks	14(a)	7.50	105.00
	Positive Ranks	0(b)	.00	.00
	Ties	1(c)		
	Total	15		

(a) Px < Py , (b) Px > Py , (c) Px = Py

ตารางที่ 5-10 ผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ของจำนวนการรับส่งข้อความหรือเมสเสจ (ต่อ)

Test Statistics (b)

	Px - Py
Z	-3.297(a)
Asymp. Sig. (2-tailed)	.001

(a) Based on positive ranks.

(b) Wilcoxon Signed Ranks Test

ดังนั้น จะปฏิเสธสมมติฐาน H_0 เมื่อค่า Asymp Sig (2-tailed) ที่คำนวณมีค่าน้อยกว่าระดับนัยสำคัญ (0.05) ในที่นี้เนื่องจากค่า Asymp Sig (2-tailed) คือ .001 มีค่าน้อยกกว่าระดับนัยสำคัญ (0.05) ผู้วิจัยจึงตัดสินใจปฏิเสธสมมติฐาน H_0 : P_x และ P_y มีค่าไม่แตกต่างกัน กัน และยอมรับสมมติฐาน H_1 : P_x และ P_y มีค่าแตกต่างกัน ดังนั้นสรุปได้ว่า จำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีค่าแตกต่างกัน โดยสามารถวิเคราะห์ข้อมูลได้ดังนี้ เมื่อนำโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันไปพัฒนาซอฟต์แวร์แล้ว พบว่าจำนวนการส่งข้อความหรือเมสเสจของซอฟต์แวร์ขณะประมวลผลจะน้อยกว่าซอฟต์แวร์ที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน จึงถือได้ว่าซอฟต์แวร์ที่สร้างด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันมีความซับซ้อนต่ำกว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ดังนั้น ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันจึงดีกว่าประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

2. การทดสอบสมมติฐานของระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสที่แตกต่างกัน

เนื่องจากการตรวจสอบการแจกแจงข้อมูลไม่มีการแจกแจงแบบปกติ สำหรับสมมติฐานนี้เป็นการกำหนดสมมติฐานแบบสองทาง เมื่อนำข้อมูลที่ได้จากการทดลองมาทดสอบทางสถิติ โดย

เลือกใช้วิธีการทดสอบประเภทวิลคอกสัน ไซน์-แรงค์ เทส โดยผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ ดังตารางที่ 5-11

ตารางที่ 5-11 ผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ของระยะเวลาที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์

Wilcoxon Signed Ranks Test

		N	Mean Rank	Sum of Ranks
Tx - Ty	Negative Ranks	15(a)	8.00	120.00
	Positive Ranks	0(b)	.00	.00
	Ties	0(c)		
	Total	15		

a Tx < Ty

b Tx > Ty

c Tx = Ty

Test Statistics(b)

	Tx - Ty
Z	-3.408(a)
Asymp. Sig. (2-tailed)	.001

a Based on positive ranks.

b Wilcoxon Signed Ranks Test

ดังนั้น จะปฏิเสธสมมติฐาน H_0 เมื่อค่า Asymp Sig (2-tailed) ที่คำนวณมีค่าน้อยกว่าระดับนัยสำคัญ (0.05) ในที่นี้เนื่องจากค่า Asymp Sig (2-tailed) คือ .001 มีค่าน้อยกกว่าระดับนัยสำคัญ (0.05) ผู้วิจัยจึงตัดสินใจปฏิเสธสมมติฐาน H_0 : T_x และ T_y มีค่าไม่แตกต่างกัน กัน และยอมรับสมมติฐาน H_1 : T_x และ T_y มีค่าแตกต่างกัน ดังนั้นสรุปได้ว่า ระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน

และระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีค่าแตกต่างกัน โดยสามารถวิเคราะห์ได้ดังนี้ เมื่อประมวลผลการทำงานของซอฟต์แวร์ ในการทำงานที่ทรานแซคชันเดียวกันและได้ผลลัพธ์ของการทำทรานแซคชันนั้นเหมือนกัน ระยะเวลาในการประมวลผลซอฟต์แวร์ที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันใช้ระยะเวลาในการประมวลผลเร็วกว่าซอฟต์แวร์ที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ดังนั้น ประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันจึงดีกว่าประสิทธิภาพของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

5.2.3 ผลการวิเคราะห์ข้อมูลเพื่อการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์

จากแบบแผนการทดลองที่กำหนดให้นำหน่วยตัวอย่างมาผ่านเครื่องมือที่ใช้สำหรับวัดจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ความต้องการเชิงฟังก์ชัน และคำนวณหาจำนวนคลาสและจำนวนเมธธอดที่ถูกกระทบจากการเปลี่ยนแปลงความต้องการ เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ โดยมีหน่วยตัวอย่างที่เป็นความต้องการของซอฟต์แวร์เชิงฟังก์ชันสำหรับใช้เปลี่ยนแปลงซอฟต์แวร์กรณีศึกษาจำนวน 30 หน่วยตัวอย่าง ดังที่กล่าวมาแล้วในหัวข้อที่ 5.1.2

เมื่อกำหนดให้หน่วยตัวอย่างทั้งหมดผ่านเครื่องมือสำหรับที่ใช้สำหรับวัดจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ ดังนั้น เมื่อประมวลผลและจัดเก็บข้อมูลจากหน่วยตัวอย่างทุกหน่วยเรียบร้อยแล้ว ได้ผลลัพธ์ของจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์

จากงานวิจัยของ Li และ Offutt (1996) ได้รวบรวมรูปแบบของการเปลี่ยนแปลงที่เกิดขึ้นในซอฟต์แวร์เชิงวัตถุ ดังนั้นผู้วิจัยได้นำรูปแบบของการเปลี่ยนแปลงมาวิเคราะห์ว่าการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์กรณีศึกษา มีลักษณะของการเปลี่ยนแปลงตามคลาสที่ถูกกระทบว่ามีผลกับคลาสยังไง สามารถวิเคราะห์ได้ดังตารางที่ 5-12

ผลกระทบจากการเปลี่ยนแปลง / ความต้องการเชิงฟังก์ชันที่	โครงสร้างคลาสแบบแอสโซซิเอชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ	โครงสร้างคลาสแบบเจเนอรัลไลเซชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ
1			✓		✓				✓	✓	✓	
2			✓						✓	✓		
3			✓		✓				✓	✓	✓	
4			✓						✓			
5		✓	✓					✓	✓			
6			✓		✓				✓	✓	✓	
7			✓		✓				✓	✓	✓	
8			✓						✓			

ตารางที่ 5-12 แสดงการวิเคราะห์ผลกระทบจากการเปลี่ยนแปลงตามความต้องการเชิงฟังก์ชัน

ความต้องการเชิงฟังก์ชันที่ / ผลกระทบจากการเปลี่ยนแปลง	โครงสร้างคลาสแบบแอสโซซิเอชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ	โครงสร้างคลาสแบบเจนเนอรัลไลเซชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ
9		✓	✓					✓	✓			
10			✓		✓				✓		✓	
11		✓	✓					✓	✓			
12		✓	✓					✓	✓			
13			✓						✓			
14			✓		✓				✓		✓	
15			✓		✓				✓	✓	✓	
16			✓		✓				✓	✓	✓	

ตารางที่ 5-12 แสดงการวิเคราะห์ผลกระทบจากการเปลี่ยนแปลงตามความต้องการเชิงฟังก์ชัน (ต่อ)

ความต้องการเชิงฟังก์ชันที่ / ผลกระทบจากการเปลี่ยนแปลง	โครงสร้างคลาสแบบแอสโซซิเอชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ	โครงสร้างคลาสแบบเจเนอรัลไลเซชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ
17			✓		✓				✓		✓	
18			✓						✓			
19		✓	✓					✓	✓			
20		✓	✓					✓	✓			
21		✓	✓					✓	✓			
22		✓	✓					✓	✓			
23			✓		✓				✓		✓	
24			✓		✓				✓		✓	

ตารางที่ 5-12 แสดงการวิเคราะห์ผลกระทบจากการเปลี่ยนแปลงตามความต้องการเชิงฟังก์ชัน (ต่อ)

ความต้องการเชิงฟังก์ชันที่ / ผลกระทบจากการเปลี่ยนแปลง	โครงสร้างคลาสแบบแอสโซซิเอชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ	โครงสร้างคลาสแบบเจนเนอรัลไลเซชัน	การเปลี่ยนแปลงแบบกระทบทั้งหมด	การเปลี่ยนแปลงที่กระทบกับคลาสปัจจุบัน	การเปลี่ยนแปลงที่กระทบกับคลาสลูก	การเปลี่ยนแปลงกระทบกับคลาสไคลเอนต์	การเปลี่ยนแปลงที่ไม่มีผลกระทบ
25			✓		✓				✓	✓	✓	
26			✓						✓			
27		✓	✓					✓	✓			
28		✓	✓					✓	✓	✓		
29			✓		✓				✓		✓	
30			✓						✓	✓		
รวม		**10**	**30**	**0**	**13**	**0**		**10**	**30**	**10**	**13**	**0**

ตารางที่ 5-12 แสดงการวิเคราะห์ผลกระทบจากการเปลี่ยนแปลงตามความต้องการเชิงฟังก์ชัน (ต่อ)

หมายเหตุ

- ในทุก ๆ การเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ พบว่าทุก ๆ การเปลี่ยนแปลงมีผลกระทบกับคลาสปัจจุบัน เนื่องจากคลาสปัจจุบันคือคลาสที่ผู้วิจัยเพิ่มความต้องการเชิงฟังก์ชัน
- ในทุก ๆ การเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน ทุก ๆ การเปลี่ยนแปลงจะไม่มีผลกระทบกับคลาสลูก เนื่องจากในความสัมพันธ์แบบแอสโซซิเอชันไม่มีคุณสมบัติของสืบการสืบทอดคุณสมบัติจากซูปเปอร์คลาสไปยังซับคลาส
- ในทุก ๆ การเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ พบว่าทุก ๆ การเปลี่ยนแปลงจะไม่พบการเปลี่ยนแปลงที่ไม่มีผลกระทบกับคลาสใด ๆ เนื่องจากความต้องการเชิงฟังชันที่ใช้เป็นหน่วยตัวอย่างเป็นฟังก์ชันที่มีการใช้งานจริง และมีผลการทำงานเชื่อมโยงกับคลาสอื่น ๆ รวมถึงมีการเรียกใช้งานเมธธอดและแอททริบิวต์ในคลาสอื่น ๆ ด้วย

จากแผนแบบการทดลองที่กำหนดให้หน่วยตัวอย่างผ่านเครื่องมือที่ใช้สำหรับวัดจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ พบว่าจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ แสดงได้ดังตารางที่ 5-13

ตารางที่ 5-13 แสดงจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์

จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลง / ความต้องการเชิงฟังก์ชันที่	จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบแอสโซซิเอชัน	จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบเจเนอรัลไลเซชัน
1	5	7
2	6	7
3	1	2
4	7	7
5	12	12
6	1	2
7	2	3
8	1	1
9	5	5
10	4	4
11	3	3
12	3	3
13	1	1
14	1	1
15	28	29
16	4	5
17	3	3
18	1	1

ตารางที่ 5-13 แสดงจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ (ต่อ)

จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลง / ความต้องการเชิงฟังก์ชันที่	จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบแอสโซซิเอชัน	จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบเจเนอรัลไลเซชัน
19	7	7
20	4	4
21	4	4
22	6	6
23	2	2
24	3	3
25	1	3
26	4	4
27	7	7
28	2	5
29	4	4
30	1	2

และจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ แสดงได้ดังตารางที่ 5-14

ตารางที่ 5-14 แสดงจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์

จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลง / ความต้องการเชิงฟังก์ชันที่	จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบแอสโซซิเอชัน	จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบเจเนอรัลไลเซชัน
1	71	73
2	86	87
3	35	36
4	89	89
5	110	110
6	51	52
7	77	78
8	2	2
9	59	59
10	49	49
11	41	41
12	58	58
13	59	59
14	51	51
15	178	178
16	69	70
17	54	54
18	60	60

ตารางที่ 5-14 แสดงจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ (ต่อ)

จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลง / ความต้องการเชิงฟังก์ชันที่	จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบแอสโซซิเอชัน	จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงของโครงสร้างคลาสแบบเจเนอรัลไลเซชัน
19	67	67
20	57	57
21	68	68
22	92	92
23	42	42
24	42	42
25	41	42
26	68	68
27	59	59
28	49	50
29	57	57
30	50	51

5.2.4 การตรวจสอบการแจกแจงข้อมูลและทดสอบสมมติฐานเพื่อการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์

ผู้วิจัยตรวจสอบการแจกแจงของข้อมูลจำนวนคลาสและจำนวนเมธธอดที่ได้ว่าเป็นการแจกแจงปกติหรือไม่ ถ้าข้อมูลมีการแจกแจงแบบปกติผู้วิจัยจะทดสอบสมมติฐานแบบอิงพารามิเตอร์ (Parametric Test) แต่ถ้าข้อมูลไม่มีการแจกแจงแบบปกติ ผู้วิจัยจะทดสอบสมมติฐาน

แบบไม่อิงกับพารามิเตอร์ (Non Parametric Test) (ศิริชัย พงษ์วิชัย, 2546) ดังนั้น ผู้วิจัยได้กำหนดสมมติฐานสำหรับการทดสอบการแจกแจงข้อมูลได้ดังนี้

1. H_0: ความแตกต่างของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติ

   H_1: ความแตกต่างของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

2. H_0: ความแตกต่างของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติ

   H_1: ความแตกต่างของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

ในการตรวจสอบการแจกแจงของข้อมูลเชิงปริมาณว่าเป็นการแจกแจงแบบปกติโดยใช้สถิติทดสอบนั้น มีสถิติที่ใช้ทดสอบ คือ Kolmogorov-Smirnov สำหรับหน่วยตัวอย่างที่มากกว่า 50 หน่วยตัวอย่าง และ Shapiro-Wilk สำหรับหน่วยตัวอย่างที่น้อยกว่า 50 หน่วยตัวอย่าง (ศิริชัย พงษ์วิชัย, 2546) สำหรับงานวิจัยนี้มีหน่วยตัวอย่างที่ใช้ในงานวิจัยจำนวน 30 หน่วยตัวอย่าง ซึ่งน้อยกว่า 50 หน่วยตัวอย่าง ดังนั้น จึงใช้เทคนิค Shapiro-Wilk ในการตรวจสอบการแจกแจงข้อมูล โดยจะเลือกทดสอบสมมติฐานแบบอิงพารามิเตอร์ (Parametric Test) เมื่อค่า Sig. (Significance) ของการทดสอบมีค่ามากกว่าหรือเท่ากับระดับนัยสำคัญที่กำหนด โดยงานวิจัยนี้กำหนดระดับนัยสำคัญเท่ากับ 0.05 ให้ยอมรับสมมติฐาน H_0 ซึ่งถือว่าข้อมูลมีการแจกแจงแบบปกติ ดังนั้นจึงทดสอบสมมติฐานแบบอิงพารามิเตอร์ ซึ่งใช้วิธีการทดสอบสมมติฐานเกี่ยวกับผลต่างระหว่างค่าเฉลี่ย 2 ประชากรแบบจับคู่ (Paired t-test) เนื่องจากการทดสอบสมมติฐานนี้เป็นการหาค่าความแตกต่างของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซ

ชัน และการหาค่าความแตกต่างของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

ในทางตรงกันข้ามจะเลือกทดสอบสมมติฐานแบบไม่อิงพารามิเตอร์ (Non Parametric Test) เมื่อมีการปฏิเสธสมมติฐานที่ H_0 โดยที่ค่า Sig. ของการทดสอบมีค่าน้อยกว่าระดับนัยสำคัญที่กำหนด โดยงานวิจัยนี้กำหนดระดับนัยสำคัญเท่ากับ 0.05 ดังนั้น จึงทดสอบสมมติฐานแบบไม่อิงพารามิเตอร์ ผู้วิจัยเลือกใช้วิธีการทดสอบสมมติฐานเกี่ยวกับการแจกแจงของข้อมูล 2 ชุดที่สัมพันธ์กัน (Two-Related-Samples Tests) โดยเลือกใช้วิธีการทดสอบประเภทวิลคอกสัน ไซน์-แรงค์ เทส (Wilcoxon signed-rank test) เนื่องจากการทดสอบสมมติฐานนี้เป็นการหาค่าความแตกต่างของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน และการหาค่าความแตกต่างของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (ศิริชัย พงษ์วิชัย, 2546)

จากการทดสอบการแจกแจงข้อมูลสามารถแสดงรายละเอียดของข้อมูลได้ดังนี้

1. ข้อมูลสำหรับการทดสอบสมมติฐานที่ 1 : ความแตกต่างของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติหรือไม่ ดังตารางที่ 5-15

ตารางที่ 5-15 แสดงการทดสอบทางสถิติตามผลการทดลองว่าจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงมีการแจกแจงแบบปกติหรือไม่

	Shapiro-Wilk		
	Statistic	df	Sig.
Association Class	.596	30	.000
Generalization Class	.590	30	.000

a Lilliefors Significance Correction

จากตารางที่ 5-15 จะเห็นได้ว่าค่า Sig. ของการทดสอบน้อยกว่าระดับนัยสำคัญที่กำหนด ดังนั้นจึงจำเป็นต้องปฏิเสธสมมติฐาน H_0 ดังนั้น ค่าความแตกต่างของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

2. ข้อมูลสำหรับการทดสอบสมมติฐานที่ 2 : ความแตกต่างของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน มีการแจกแจงแบบปกติหรือไม่ ดังตารางที่ 5-16

ตารางที่ 5-16 แสดงการทดสอบทางสถิติตามผลการทดลองว่าจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงมีการแจกแจงแบบปกติหรือไม่

	Shapiro-Wilk		
	Statistic	df	Sig.
Association Class	.816	30	.000
Generalization Class	.817	30	.000

a Lilliefors Significance Correction

จากตารางที่ 5-16 จะเห็นได้ว่าค่า Sig. ของการทดสอบน้อยกว่าระดับนัยสำคัญที่กำหนด ดังนั้นจึงจำเป็นต้องปฏิเสธสมมติฐาน H_0 ดังนั้น ค่าความแตกต่างของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ไม่มีการแจกแจงแบบปกติ

โดยงานวิจัยนี้สนใจการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันระหว่างโครงสร้างของคลาสในความสัมพันธ์แบบแอสโซซิเอชันและแบบเจเนอรัลไลเซชัน ว่ามีผลกับคุณภาพของซอฟต์แวร์ในเชิงของการแก้ไข ปรับปรุงและการบำรุงรักษาซอฟต์แวร์

กำหนดให้ C_x คือ จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน

C_y คือ จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

M_x คือ จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน

M_y คือ จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

จากตัวแปรที่ผู้วิจัยกำหนด สามารถนำมาตั้งสมมติฐานเพื่อหาความสัมพันธ์ระหว่างตัวแปรเพื่อนำมาตอบวัตถุประสงค์ของงานวิจัยภายใต้ขอบเขตของปัญหาที่กำหนด ได้แก่ การเปรียบเทียบจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันระหว่างโครงสร้างของคลาสในความสัมพันธ์แบบแอสโซซิเอชันและแบบเจเนอรัลไลเซชัน

1. $H_0 : C_x$ และ C_y มีค่าไม่แตกต่างกัน

   $H_1 : C_x$ และ C_y มีค่าแตกต่างกัน

2. $H_0 : M_x$ และ M_y มีค่าไม่แตกต่างกัน

   $H_1 : M_x$ และ M_y มีค่าแตกต่างกัน

ซึ่งจากการตรวจสอบการแจกแจงข้อมูล พบว่าจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงไม่มีการแจกแจงแบบปกติ ผู้วิจัยจึงเลือกการทดสอบสมมติฐานเกี่ยวกับการแจกแจงของข้อมูล 2 ชุดที่สัมพันธ์กัน (Two-Related-Samples Tests) โดยจะใช้โปรแกรม SPSS เวอร์ชัน 14 ในการทดสอบข้อมูล

1 การทดสอบสมมติฐานจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสที่แตกต่างกัน

เนื่องจากการตรวจสอบการแจกแจงข้อมูลไม่มีการแจกแจงแบบปกติ สำหรับสมมติฐานนี้เป็นการกำหนดสมมติฐานแบบสองทาง เมื่อนำข้อมูลที่ได้จากการทดลองมาทดสอบทางสถิติ โดยเลือกใช้วิธีการทดสอบประเภทวิลคอกสัน ไซน์-แรงค์ เทส โดยผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ ดังตารางที่ 5-17

ตารางที่ 5-17 ผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ของจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลง

Wilcoxon Signed Ranks Test

		N	Mean Rank	Sum of Ranks
Generalization - Association	Negative Ranks	0(a)	.00	.00
	Positive Ranks	10(b)	5.50	55.00
	Ties	20(c)		
	Total	30		

(a) Generalization < Association

(b) Generalization > Association

(c) Generalization = Association

Test Statistics(b)

	Generalization - Association
Z	-2.913(a)
Asymp. Sig. (2-tailed)	.004

a Based on negative ranks.

b Wilcoxon Signed Ranks Test

ดังนั้น จะปฏิเสธสมมติฐาน H_0 เมื่อค่า Asymp Sig (2-tailed) ที่คำนวณมีค่าน้อยกว่าระดับนัยสำคัญ (0.05) ในที่นี้เนื่องจากค่า Asymp Sig (2-tailed) คือ .004 มีค่าน้อยกกว่าระดับนัยสำคัญ (0.05) ผู้วิจัยจึงตัดสินใจปฏิเสธสมมติฐาน H_0 : C_x และ C_y มีค่าไม่แตกต่างกัน และยอมรับสมมติฐาน H_1 : C_x และ C_y มีค่าแตกต่างกัน ดังนั้นสรุปได้ว่า จำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีค่าแตกต่างกัน โดยสามารถวิเคราะห์ได้ดังนี้ เมื่อมีการแก้ไข ปรับปรุง เปลี่ยนแปลงและการบำรุงรักษาซอฟต์แวร์ จำนวนคลาสที่ได้รับผลกระทบจาก

การเปลี่ยนแปลงความต้องการของซอฟต์แวร์ที่ออกแบบด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันจะได้รับผลกระทบจากการเปลี่ยนแปลงน้อยกว่าซอฟต์แวร์ที่ออกแบบด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

2 การทดสอบสมมติฐานจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสที่แตกต่างกัน

เนื่องจากการตรวจสอบการแจกแจงข้อมูลไม่มีการแจกแจงแบบปกติ สำหรับสมมติฐานนี้เป็นการกำหนดสมมติฐานแบบสองทาง เมื่อนำข้อมูลที่ได้จากการทดลองมาทดสอบทางสถิติ โดยเลือกใช้วิธีการทดสอบประเภทวิลคอกสัน ไซน์-แรงค์ เทส โดยผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ ดังตารางที่ 5-18

ตารางที่ 5-18 ผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลง

Wilcoxon Signed Ranks Test

		N	Mean Rank	Sum of Ranks
Association - Generalization	Negative Ranks	9(a)	5.00	45.00
	Positive Ranks	0(b)	.00	.00
	Ties	21(c)		
	Total	30		

(a) Association < Generalization

(b) Association > Generalization

(c) Association = Generalization

ตารางที่ 5-18 ผลลัพธ์ทางสถิติที่ได้จากโปรแกรม SPSS สำหรับการทดสอบค่าเฉลี่ยของ 2 กลุ่มตัวอย่างแบบไม่อิงพารามิเตอร์ของจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลง (ต่อ)

Test Statistics(b)

	Association - Generalization
Z	-2.887(a)
Asymp. Sig. (2-tailed)	.004

a Based on positive ranks.

b Wilcoxon Signed Ranks Test

ดังนั้น จะปฏิเสธสมมติฐาน H_0 เมื่อค่า Asymp Sig (2-tailed) ที่คำนวณมีค่าน้อยกว่าระดับนัยสำคัญ (0.05) ในที่นี้เนื่องจากค่า Asymp Sig (2-tailed) คือ .004 มีค่าน้อยกกว่าระดับนัยสำคัญ (0.05) ผู้วิจัยจึงตัดสินใจปฏิเสธสมมติฐาน H_0 : M_x และ M_y มีค่าไม่แตกต่างกัน และยอมรับสมมติฐาน H_1 : M_x และ M_y มีค่าแตกต่างกัน ดังนั้นสรุปได้ว่า จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและจำนวนที่ได้เมธธอดรับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีค่าแตกต่างกัน โดยสามารถวิเคราะห์ได้ดังนี้ เมื่อมีการแก้ไข ปรับปรุง เปลี่ยนแปลงและการบำรุงรักษาซอฟต์แวร์ จำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์ที่ออกแบบด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันจะได้รับผลกระทบจากการเปลี่ยนแปลงน้อยกว่าซอฟต์แวร์ที่ออกแบบด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

จากสมมติฐานทั้งสองสมมติฐาน สามารถวิเคราะห์ได้ว่าซอฟต์แวร์ที่ออกแบบด้วยโครงสร้างคลาสในความแบบแอสโซซิเอชันมีความสามารถในการบำรุงรักษาซอฟต์แวร์ได้ดีกว่าซอฟต์แวร์ที่ออกแบบด้วยโครงสร้างคลาสในความแบบเจเนอรัลไลเซชัน เนื่องจากเมื่อมีการบำรุงรักษาซอฟต์แวร์จะมีผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์น้อยกว่า