บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าวถึง การสรุปผลการวิเคราะห์ข้อมูลเพื่อตอบวัตถุประสงค์ของงานวิจัย การอภิปรายประเด็นต่าง ๆ ที่เกิดขึ้นในงานวิจัย การนำงานวิจัยนี้ไปใช้ประโยชน์ในเชิงทฤษฎีและเชิงประยุกต์ ข้อจำกัดของงานวิจัยและข้อเสนอแนะเพื่อเป็นประโยชน์ในการนำงานวิจัยไปศึกษาต่อในอนาคต

6.1 การทดลองและลักษณะของหน่วยตัวอย่าง

งานวิจัยนี้เป็นการวิจัยเชิงทดลอง (Experimental Research) โดยการทดลองต้องการตอบวัตถุประสงค์ของงานวิจัย คือ (1) เพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล (Run – Time) ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน (Association Relationship) และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (Generalization Relationship) และ (2) เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน (Functional Requirement) ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ดังนั้นผู้วิจัยจึงได้แบ่งหน่วยตัวอย่างที่ใช้ในงานวิจัยออกเป็น 2 กลุ่ม คือ

- กลุ่มที่ 1 คือ หน่วยตัวอย่างที่ใช้เพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล ซึ่งผู้วิจัยกำหนดให้หน่วยตัวอย่างเป็นซอฟต์แวร์ระบบที่ออกแบบโดยใช้หลักการเชิงวัตถุจำนวน 5 หน่วยตัวอย่าง
- กลุ่มที่ 2 คือ หน่วยตัวอย่างที่ใช้เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน โดยผู้วิจัยเลือกซอฟต์แวร์กรณีศึกษาที่ออกแบบโดยใช้หลักการเชิงวัตถุจำนวน 1 ซอฟต์แวร์ และกำหนดให้หน่วยตัวอย่างที่ใช้ในการทดลองเป็นความต้องการเชิงฟังก์ชันของซอฟต์แวร์ที่เลือกจำนวน 30 หน่วยตัวอย่าง

โดยหน่วยตัวอย่างทั้ง 2 กลุ่ม มีข้อกำหนดว่าแต่ละซอฟต์แวร์เชิงวัตถุจะต้องมีจำนวนคลาสการทำงานในระบบอย่างน้อย 5 คลาสขึ้นไปและแต่ละซอฟต์แวร์เชิงวัตถุสามารถนำมาปรับโครงสร้างคลาสให้อยู่ในลักษณะโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันได้ เมื่อผ่านกระบวนการปรับเปลี่ยน

โครงสร้างคลาสและประมวลผลเรียบร้อยแล้ว ข้อมูลจะถูกนำมาวิเคราะห์ผลทางสถิติเพื่อนำมาสรุปผลตามสมมติฐานที่ตั้งไว้

6.2 บทสรุป

งานวิจัยนี้เป็นการเปรียบเทียบคุณภาพของซอฟต์แวร์เชิงวัตถุที่ออกแบบซอฟต์แวร์ด้วยโครงสร้างคลาสที่แตกต่างกัน 2 ลักษณะ ได้แก่ การออกแบบซอฟต์แวร์ด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และการออกแบบซอฟต์แวร์ด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยงานวิจัยนี้ผู้วิจัยได้แบ่งการศึกษาออกเป็น 2 กลุ่ม คือ

- การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล (Run – Time) โดยวิเคราะห์ผลการศึกษาจากจำนวนการส่งข้อความหรือเมสเสจ (Message Calling) ระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์ (ชาลี วรกุลพิพัฒน์ และ เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2544) และวัดจากระยะเวลาการทำงานที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์
- การเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน (Functional Requirement) โดยวิเคราะห์ผลการศึกษาจากการเพิ่มความต้องการเชิงฟังก์ชันเข้าสู่ซอฟต์แวร์ที่เป็นกรณีศึกษา จากนั้นพิจารณาจากจำนวนคลาส และจำนวนเมธธอดที่ถูกกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์

โดยผลการทดสอบงานวิจัยนี้สามารถนำมาวิเคราะห์ความสัมพันธ์ระหว่างโครงสร้างคลาสกับคุณภาพของซอฟต์แวร์ คือ ประสิทธิภาพของซอฟต์แวร์ (Software Performance) และความสามารถในการบำรุงรักษาซอฟต์แวร์ (Software Maintainability)

สำหรับ (1) การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันและ (2) การเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชันระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน สามารถสรุปผลการเปรียบเทียบได้ดังนี้

6.2.1 การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

จากการวิเคราะห์ผลของการเปรียบเทียบประสิทธิภาพของโครงสร้างคลาสขณะประมวลผลซอฟต์แวร์ ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยวัดจากจำนวนการรับส่งข้อความหรือเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์และระยะเวลาการทำทรานแซคชันที่ซอฟต์แวร์ถูกประมวลผลโดยสมบูรณ์ ปรากฏว่า ซอฟต์แวร์ที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันมีประสิทธิภาพขณะประมวลผลดีกว่าซอฟต์แวร์ที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน เนื่องจากโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันมีความซับซ้อนในการทำงานน้อยกว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (พิจารณาจากจำนวนการส่งข้อความหรือเมสเสจ) และมีความเร็วในการประมวลผลซอฟต์แวร์ที่เร็วกว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยผลจากงานวิจัยนี้ถือเป็นองค์ประกอบหนึ่งในการตัดสินใจให้ผู้ออกแบบและพัฒนาซอฟต์แวร์ใช้เป็นปัจจัยเพื่อเลือกลักษณะของโครงสร้างคลาสในการออกแบบและพัฒนาซอฟต์แวร์ได้

6.2.2 การเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชันระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

งานวิจัยนี้สนใจการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันระหว่างโครงสร้างของคลาสในความสัมพันธ์แบบแอสโซซิเอชันและแบบเจเนอรัลไลเซชัน ว่ามีผลกับคุณภาพของซอฟต์แวร์ในเชิงของการแก้ไข ปรับปรุงและการบำรุงรักษาซอฟต์แวร์ (Maintainability)

สำหรับการเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและแบบเจเนอรัลไลเซชัน สามารถสรุปผลได้ดังนี้

- ผลการเปรียบเทียบจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ปรากฏว่าจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน มีจำนวนน้อยกว่าจำนวนคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลง

ความต้องการเชิงฟังก์ของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ซึ่งเป็นไปตามที่ผู้วิจัยคาดหวังไว้ก่อนที่จะทำการทดลอง เนื่องจาก จำนวนผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์เกิดจากการที่ในโครงสร้างคลาสมีการเชื่อมต่อความสัมพันธ์ภายในคลาส และมีการเชื่อมต่อความสัมพันธ์ไปยังคลาสอื่น ๆ ซึ่งยังหมายรวมถึงการเรียกใช้เมธธอดหรือแอททริบิวต์ในคลาสอื่น ๆ ด้วย

- ผลการเปรียบเทียบจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ปรากฏว่าจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน มีจำนวนน้อยกว่าจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ซึ่งเป็นไปตามที่ผู้วิจัยคาดหวังไว้ก่อนที่จะทำการทดลอง

เนื่องจาก จำนวนผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์เกิดจากการที่ในโครงสร้างคลาสมีการเชื่อมต่อความสัมพันธ์ภายในคลาส และมีการเชื่อมต่อความสัมพันธ์ไปยังคลาสอื่น ๆ ซึ่งหมายรวมถึงการเรียกใช้เมธธอดหรือแอททริบิวต์ในคลาสอื่น ๆ ด้วย ดังนั้นหากโครงสร้างคลาสมีความสัมพันธ์เชื่อมต่อระหว่างคลาสอื่น ๆ รวมถึงเมธธอดและแอททริบิวต์ในคลาสอื่น ๆ มากจะเป็นผลให้การแก้ไข ปรับปรุงและการบำรุงรักษาซอฟต์แวร์ทำได้ยาก

นอกจากนี้ พบว่าการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์ที่มีผลกระทบกับคลาสลูกตามที่ผู้วิจัยได้พิจารณาจากงานวิจัยของ Li และ Offutt (1996) ซึ่งเป็นการเปลี่ยนแปลงที่มีผลกับคลาสที่มีการสืบทอดคุณสมบัติ แสดงให้เห็นผลลัพธ์ความแตกต่างของจำนวนคลาสและจำนวนเมธธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน แสดงให้เห็นว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมีจำนวนผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์มากกว่าโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน ทำให้การแก้ไข ปรับปรุงและการบำรุงรักษาซอฟต์แวร์ทำได้ยากกว่า หรือมีผลกระทบจากการเปลี่ยนแปลงมากกว่า

6.3 การนำงานวิจัยไปประยุกต์ใช้ (Contribution)

งานวิจัยนี้สามารถนำไปเป็นประโยชน์ในเชิงทฤษฎีแล้ว ยังสามารถนำไปประยุกต์ใช้ได้ดังต่อไปนี้

6.3.1 การนำงานวิจัยไปใช้ในเชิงทฤษฎี (Theoretical Contribution)

งานวิจัยนี้เป็นการต่อยอดองค์ความรู้ในการพัฒนาซอฟต์แวร์ให้มีคุณภาพมากยิ่งขึ้น เนื่องจากผู้วิจัยได้ทบทวนวรรณกรรมในอดีต (Literature Review) แล้วพบว่าการสืบทอดคุณสมบัติของคลาสมีผลระยะเวลาในการบำรุงรักษา (Maintainability) ซอฟต์แวร์เชิงวัตถุ (Daly และคณะ, 1995) โดยไม่ได้กล่าวถึงประสิทธิภาพของซอฟต์แวร์และผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน ซึ่งประเด็นนี้มีส่วนสนับสนุนว่าลักษณะการออกแบบโครงสร้างคลาสที่แตกต่างกันน่าจะมีผลต่อประสิทธิภาพของซอฟต์แวร์ และการบำรุงรักษาซอฟต์แวร์ เพื่อนำข้อมูลเหล่านี้ไปใช้ประโยชน์ในการศึกษาต่อไป โดยอาจจะพิจารณาครอบคลุมถึงคุณภาพของซอฟต์แวร์ในด้านอื่น ๆ อีก เช่น พิจารณาประสิทธิภาพของซอฟต์แวร์ในเทอมที่ครอบคลุมกับทุก ๆ กระบวนการในการพัฒนาซอฟต์แวร์ ไม่ใช่การพิจารณาประสิทธิภาพของซอฟต์แวร์ในเทอมของการอิมพลีเมนต์เฟส (Implementation Phase) การรักษาความปลอดภัยของซอฟต์แวร์ เป็นต้น เพื่อเป็นแนวทางให้กับผู้ที่สนใจในเรื่องการออกแบบ พัฒนาซอฟต์แวร์เชิงวัตถุและคุณภาพของซอฟต์แวร์ได้นำข้อมูลงานวิจัยนี้ไปใช้ในการศึกษาต่อไป

6.3.2 การนำงานวิจัยไปใช้ในเชิงประยุกต์ (Practical Contribution)

โดยผลของการศึกษางานวิจัยนี้ สนใจประเด็นของประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล และผลกระทบจากการเปลี่ยนแปลงความต้องการเชิงฟังก์ชันของซอฟต์แวร์โดยการออกแบบโดยใช้คุณลักษณะของโครงสร้างคลาสที่แตกต่างกัน คือ โครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ซึ่งผลของการทำการทดลองงานวิจัยนี้แสดงให้เห็นว่าโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันมีประสิทธิภาพดีกว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันทั้งในด้านของจำนวนการรับส่งเมสเสจระหว่างคลาสทั้งหมดที่เป็นองค์ประกอบของซอฟต์แวร์และด้านระยะเวลาที่ประมวลผลซอฟต์แวร์โดยสมบูรณ์ และโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันมีความสามารถในการบำรุงรักษาซอฟต์แวร์ได้ดีกว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน เนื่องจากเมื่อมีการเปลี่ยนแปลงความต้องการของซอฟต์แวร์โครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันจะได้รับผลกระทบจากการเปลี่ยนแปลงน้อยกว่าโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

จากผลสรุปดังกล่าวสามารถนำผลสรุปจากงานวิจัย ไปใช้แนะนำผู้ออกแบบซอฟต์แวร์เชิงวัตถุหรือนักพัฒนาซอฟต์แวร์เชิงวัตถุ ว่าประเด็นไหนในหลักการออกแบบและพัฒนาเชิงวัตถุที่ควรให้ความสำคัญ เช่น การออกแบบและพัฒนาซอฟต์แวร์เชิงวัตถุควรพิจารณาโมเดลทางธุรกิจให้เข้าใจชัดเจน เพื่อปรับเข้าสู่การออกแบบและพัฒนาหน้าที่การทำงานของแต่ละคลาสและเมธธอด และสามารถสร้างความสัมพันธ์ระหว่างคลาสได้ชัดเจนยิ่งขึ้น

6.4 ข้อจำกัดและข้อเสนอแนะของงานวิจัย

งานวิจัยนี้มีข้อจำกัดบางประการ ซึ่งผู้วิจัยขอสรุปและเสนอแนะแนวทาง ดังนี้

1. ขนาดของโครงสร้างคลาส เนื่องจากผู้วิจัยกำหนดได้กำหนดให้โครงสร้างคลาสที่นำมาใช้เป็นหน่วยตัวอย่างว่าจะต้องมีจำนวนโครงสร้างคลาสในระบบอย่างน้อยจำนวน 5 คลาสขึ้นไป แต่โครงสร้างคลาสที่นำมาใช้ในงานวิจัยมีจำนวนไม่เกิน 25 คลาส ดังนั้นถ้านำงานวิจัยไปเปรียบเทียบที่จำนวนโครงสร้างคลาสมากกว่า 25 คลาส อาจจะทำให้ผลการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผลคลาดเคลื่อนได้
2. รูปแบบของการเขียนซอร์สโค้ดของหน่วยตัวอย่าง เนื่องจากผู้วิจัยได้นำซอร์สโค้ดหน่วยตัวอย่างมาจากเว็บไซต์โอเพนซอร์ส (Open Source) โดยแต่ละหน่วยตัวอย่างที่เลือกอาจจะมีรูปแบบและหลักการการพัฒนาซอร์สโค้ดที่แตกต่างกันออกไป ซึ่งอาจจะมีผลทำให้ผลการเปรียบเทียบประสิทธิภาพขณะประมวลผลคลาดเคลื่อนได้
3. เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์สำหรับใช้ประมวลผลประสิทธิภาพและผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์พัฒนาภายใต้ชุดโปรแกรมสำหรับพัฒนาโปรแกรมภาษาจาวาอีคลิพซ์ (Eclipse) ดังนั้น หากนำซอฟต์แวร์นี้ไปประมวลผลในชุดโปรแกรมสำหรับพัฒนาโปรแกรมภาษาจาวาอื่นอาจให้ผลลัพธ์ในการประมวลผลที่ต่างกันออกไป เนื่องจากเครื่องมือแต่ละชนิดจะมีจาวาคลาสไลบรารี (Java Class Library) ที่ให้บริการแตกต่างกัน
4. การนำผลการทดลองนี้ไปใช้ให้คำนึงด้วยว่า หน่วยตัวอย่างที่ใช้ในงานวิจัยนี้ส่วนหนึ่งเป็นซอฟต์แวร์โอเพนซอร์ส ดังนั้นการนำข้อมูลการวิจัยไปขยายผลในองค์กรที่พัฒนาซอฟต์แวร์ทางธุรกิจ อาจจะได้ผลลัพธ์ที่แตกต่างออกไปจากงานวิจัย

6.5 แนวทางการวิจัยในอนาคต

แนวทางการทำงานวิจัยในอนาคต มีดังนี้

1. พิจารณาคุณภาพของซอฟต์แวร์ในด้านอื่น ๆ เนื่องจากการออกแบบโครงสร้างของคลาสอาจจะมีผลกับคุณภาพของซอฟต์แวร์ที่แตกต่างกัน เช่น การรักษาความปลอดภัยของซอฟต์แวร์ การนำซอฟต์แวร์กลับมาใช้ใหม่ เป็นต้น
2. เพิ่มการพิจารณาประสิทธิภาพของซอฟต์แวร์ของโครงสร้างคลาสการทำงานในเชิงธุรกิจ (Business Class Diagram) ที่มีความซับซ้อน (Complexity) มากขึ้น โดยการเพิ่มระดับชั้นความลึกในการสืบทอดคุณสมบัติและความซับซ้อนในการสืบทอดคุณสมบัติ (Multiple Inheritance) เช่น กำหนดให้หนึ่งซับคลาสมีการสืบทอดคุณสมบัติมาจากหลายซูปเปอร์คลาส เป็นต้น และกำหนดจำนวนกลุ่มของการสืบทอดคุณสมบัติของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันมากกว่าหนึ่งกลุ่ม