

บทที่ 1

บทนำ

ในบทนี้จะขอกล่าวถึง ความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของงานวิจัยและขั้นตอนการดำเนินการวิจัย ประโยชน์ที่คาดว่าจะได้รับ เพื่อนำไปสู่การดำเนินการวิจัยในบทต่อ ๆ ไป

1.1 ความเป็นมาและความสำคัญของปัญหา

ตลอดระยะเวลาหลายปีที่ผ่านมาองค์กรธุรกิจได้มีการนำเอาระบบสารสนเทศเชิงธุรกิจเข้ามาเป็นส่วนหนึ่งของแผนกลยุทธ์ที่องค์กรให้ความสำคัญ เนื่องจากการประยุกต์ใช้เทคโนโลยีสารสนเทศ (Information Technology : IT) มีส่วนเสริมสร้างความสำเร็จการแข่งขันและพัฒนาประสิทธิภาพในการบริหารงานในองค์กร รวมถึงกำหนดทิศทางการพัฒนาองค์กร ดังนั้นจึงมีการนำเอาซอฟต์แวร์มาประยุกต์ใช้กับงานด้านต่าง ๆ เพื่อรองรับการทำงานในระบบธุรกิจ เนื่องด้วยซอฟต์แวร์เป็นส่วนประกอบหนึ่งที่ทำให้ระบบสารสนเทศขับเคลื่อนไปในทิศทางที่องค์กรต้องการได้ (กรมเศรษฐกิจพาณิชย์, 2545)

การบริหารองค์กรธุรกิจมีการเปลี่ยนแปลงภายใต้ภาวะการแข่งขันกันทางธุรกิจระหว่างองค์กร และนโยบายของรัฐบาลที่ต้องการพัฒนาเศรษฐกิจและสังคมโดยใช้เทคโนโลยีสารสนเทศและการสื่อสารที่ทันสมัยเป็นแรงขับเคลื่อน เช่น การยกระดับประสิทธิภาพการผลิตโดยใช้เทคโนโลยีสารสนเทศ การขยายฐานการตลาดโดยใช้เทคโนโลยีสารสนเทศ เป็นต้น (แผนแม่บทเทคโนโลยีสารสนเทศและการสื่อสารของประเทศไทย, 2545-2549) ซึ่งนโยบายเหล่านี้ส่งผลกระทบกับการเปลี่ยนแปลงสารสนเทศทางธุรกิจตามมาทำให้มีการพัฒนาและปรับปรุงซอฟต์แวร์ธุรกิจอย่างต่อเนื่อง

โดยซอฟต์แวร์ที่พัฒนาขึ้นมาแล้ว จะเข้าสู่วัฏจักรของการนำไปใช้งาน การบำรุงรักษาหรือการปรับปรุงแก้ไข และการนำส่วนของซอฟต์แวร์กลับมาใช้ใหม่ โดยการบำรุงรักษาซอฟต์แวร์อาจเกิดจากการที่พบข้อผิดพลาดของซอฟต์แวร์หลงเหลืออยู่หรือข้อกำหนดของเงื่อนไขในซอฟต์แวร์ (Software Requirement Specification) มีการเปลี่ยนแปลงไป เช่น การเปลี่ยนแปลงการจัดเก็บภาษี การปรับอัตราค่าเงินเดือนพนักงาน เป็นต้น หรือสภาวะแวดล้อมในการทำงานของซอฟต์แวร์เปลี่ยนแปลงไป เพื่อให้ได้คุณภาพของซอฟต์แวร์ตามต้องการ

ดังนั้น ขั้นตอนการเปลี่ยนแปลงซอฟต์แวร์ (Software Change) จึงเป็นขั้นตอนหนึ่งที่มีความสำคัญอย่างยิ่งที่จะทำให้ได้มาซึ่งซอฟต์แวร์ที่มีคุณภาพ (Martin, 1983) ซึ่งคุณภาพที่องค์กรส่วนใหญ่ให้ความสำคัญ ได้แก่ ความสามารถในการบำรุงรักษาซอฟต์แวร์ (Maintainability) และความสามารถในการนำซอฟต์แวร์ที่มีอยู่กลับมาใช้ใหม่ (Reusability) (Young, 2000) แต่ปัญหาสำคัญของการบำรุงรักษาซอฟต์แวร์อยู่ที่ซอฟต์แวร์หรือโปรแกรมไม่ได้มีรูปแบบหรือโครงสร้างที่ดีพอสำหรับการแก้ไขหรือยากที่จะนำซอฟต์แวร์กลับมาใช้ใหม่ ทำให้บางครั้งต้องมีการพัฒนาซอฟต์แวร์ขึ้นมาใหม่ทั้งหมด จากสาเหตุดังกล่าวทำให้การพัฒนาซอฟต์แวร์ได้เปลี่ยนแปลงเข้าสู่การใช้เทคโนโลยีเชิงวัตถุ

จากแนวคิดเชิงวัตถุ ทำให้การพัฒนาซอฟต์แวร์แนวใหม่หันมาสนับสนุนการพัฒนาโดยอาศัยแนวคิดนี้ เนื่องจากแนวคิดเชิงวัตถุสนับสนุนการพัฒนาซอฟต์แวร์ที่ซับซ้อน สนับสนุนการนำกลับมาใช้งานซ้ำ และสามารถปรับปรุงแก้ไขและบำรุงรักษาได้ง่าย เพราะแต่ละคลาสหรือออบเจกต์ที่กำหนดขึ้นนั้นจะมีความสมบูรณ์อยู่ในตัวเอง ซึ่งประกอบด้วยข้อมูลและการดำเนินการภายในแต่ละคลาสหรือออบเจกต์นั้น (ชาติ วรกุลพิพัฒน์ และ เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2544) ทำให้การพัฒนาทำได้ง่ายขึ้นและรองรับการเปลี่ยนแปลงในภายหลังได้ง่าย แต่การใช้แนวคิดเชิงวัตถุก็ไม่ได้หมายความว่า จะมีประสิทธิภาพหรือความคล่องตัวในการเปลี่ยนแปลง เนื่องจากการเปลี่ยนแปลงที่ออบเจกต์หนึ่งอาจจะมีผลกระทบกับการทำงานของออบเจกต์อื่นได้ เช่น การเปลี่ยนแปลงฟังก์ชันการทำงานอาจทำให้ข้อมูล (Data) หรือแอททริบิวต์และกระบวนการทำงาน (Process) หรือเมธอด (Method) เปลี่ยนแปลงไป ตัวอย่างเช่น การแก้ไขและการเพิ่มข้อมูลต่าง ๆ อาจกระทบกับกระบวนการรายงานผลที่อาจจะต้องปรับเปลี่ยนรูปแบบการรายงานผลตามมาเป็นต้น

กระบวนการในการรองรับการเปลี่ยนแปลงนี้ จึงเป็นส่วนหนึ่งของการบำรุงรักษาซอฟต์แวร์ (Software Maintenance) ซึ่งขั้นตอนในการบำรุงรักษาซอฟต์แวร์ประกอบด้วย การแก้ไขข้อผิดพลาดที่เกิดขึ้น (Corrective Maintenance) การเพิ่มประสิทธิภาพในการบำรุงรักษาซอฟต์แวร์ด้วยการปรับปรุงระบบให้ดีขึ้น (Perfective Maintenance) การปรับเปลี่ยนซอฟต์แวร์ให้เข้ากับสภาพแวดล้อมใหม่ (Adaptive Maintenance) และการป้องกันข้อผิดพลาดที่อาจจะตามมาในอนาคต (Preventive Maintenance) นอกจากนี้ยังคำนึงถึงการนำเอาซอฟต์แวร์กลับมาใช้ใหม่ ซึ่งจะทำให้คุณภาพของซอฟต์แวร์สูงขึ้น และมีต้นทุนในการบำรุงรักษาซอฟต์แวร์ลดลง (Sommerville, 2001)

แต่เนื่องด้วยการพัฒนาซอฟต์แวร์ในปัจจุบัน เป็นการพัฒนาซอฟต์แวร์เชิงวัตถุ ดังนั้นคลาสจึงเป็นหน่วยพื้นฐานสำคัญในการพัฒนาซอฟต์แวร์ (Pressman, 2001) เนื่องจากคลาสสามารถระบุ

ถึงแอททริบิวต์ คุณสมบัติต่าง ๆ ของออบเจกต์ รวมถึงความสัมพันธ์ระหว่างออบเจกต์ได้ดี และสามารถบอกถึงคุณภาพของซอฟต์แวร์ได้ และจากเหตุผลดังที่ได้กล่าวมาแล้วข้างต้น เมื่อมีการเปลี่ยนแปลงที่เกิดขึ้นในซอฟต์แวร์เชิงวัตถุ คลาสที่ออกแบบต่าง โครงสร้างหรือต่างลักษณะกัน ประสิทธิภาพของการแก้ไขหรือภาระงาน รวมทั้งต้นทุนที่แต่ละคลาสต้องรองรับการเปลี่ยนแปลง อาจจะมากน้อยแตกต่างกัน

ทำให้ผู้วิจัยเกิดแนวคิดในการเปรียบเทียบผลกระทบของการเปลี่ยนแปลงซอฟต์แวร์ที่ ออกแบบด้วยโครงสร้างคลาสที่ต่างกัน โดยการพิจารณาถึงความเกี่ยวข้องกันระหว่างการออกแบบ โครงสร้างของคลาสและคุณภาพของซอฟต์แวร์ เพื่อวิเคราะห์ถึงผลกระทบและประสิทธิภาพของ ซอฟต์แวร์ขณะประมวลผลระหว่างการเปลี่ยนแปลงโครงสร้างคลาส การเปลี่ยนแปลงของแอททริ บิวต์และเมธอด เพื่อพิจารณาถึงผลกระทบที่อาจจะเกิดขึ้นจากการเปลี่ยนแปลง และนำเสนอ โครงสร้างของคลาสที่เหมาะสมเพื่อรองรับการเปลี่ยนแปลงซอฟต์แวร์ทางธุรกิจ

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อเปรียบเทียบประสิทธิภาพของโครงสร้างคลาสดขณะประมวลผล (Run – Time) ระหว่างซอฟต์แวร์เชิงวัตถุที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน (Association Relationship) และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน (Generalization Relationship)
- 1.2.2 เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน (Functional Requirement) ระหว่างซอฟต์แวร์เชิงวัตถุที่พัฒนาด้วยโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

1.3 ขอบเขตของการวิจัย

- 1.3.1 การเปลี่ยนแปลงโครงสร้างของคลาส งานวิจัยนี้พิจารณาเฉพาะการเปลี่ยนแปลงโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน
- 1.3.2 การเปลี่ยนแปลงโครงสร้างของคลาสให้อยู่ในรูปแบบของโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันเพื่อการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ จะพิจารณาปรับเปลี่ยนโครงสร้างของคลาสที่เกี่ยวข้องกับคลาสการทำงานของซอฟต์แวร์

ระบบ (Business Class) เท่านั้น โดยจะไม่พิจารณารวมถึงคลาสที่มีการเรียกใช้งานไลบรารีต่าง ๆ (Class Libraries) และส่วนติดต่อประสานงานกับผู้ใช้ (User Interface)

- 1.3.3 การเปลี่ยนแปลงความต้องการของซอฟต์แวร์ จะพิจารณาที่ความต้องการเชิงฟังก์ชัน (Functional Requirement) เท่านั้น
- 1.3.4 การเปลี่ยนแปลงคุณลักษณะของคลาสตามความต้องการเชิงฟังก์ชันของซอฟต์แวร์ งานวิจัยนี้พิจารณาเฉพาะลักษณะของการเปลี่ยนแปลงที่เกิดขึ้นกับการเพิ่ม การลบ การแก้ไขแอททริบิวต์และเมธอดในคลาส การเปลี่ยนแปลงที่คลาสพิจารณาที่การเพิ่มคลาสและลบคลาส
- 1.3.5 หน่วยตัวอย่างที่เลือกมาใช้ในงานวิจัย จะต้องเป็นคลาสที่มีความหลากหลายทางธุรกิจและจะมีจำนวนคลาสการทำงานภายในระบบอย่างน้อย 5 คลาส

1.4 ขั้นตอนการดำเนินการวิจัย

ในงานวิจัยนี้ได้พิจารณาขั้นตอนการดำเนินงานวิจัยออกเป็น 2 ส่วน คือ (1) การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล และ (2) การเปรียบเทียบจำนวนคลาสและจำนวนเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ ซึ่งมีขั้นตอนการดำเนินงานวิจัย ดังนี้

1.4.1 การเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล

โดยขั้นตอนการดำเนินการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ มีดังนี้

- เลือกหน่วยตัวอย่างที่เป็นซอฟต์แวร์ระบบธุรกิจที่ออกโดยใช้หลักการเชิงวัตถุ สำหรับใช้ในงานวิจัย โดยกำหนดจำนวนหน่วยตัวอย่าง 5 หน่วยตัวอย่าง ซึ่งแต่ละหน่วยตัวอย่างที่เลือก จะต้องสามารถปรับเปลี่ยนโครงสร้างของคลาสให้เป็นโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันได้ โดยมีจำนวนคลาสการทำงานเชิงธุรกิจในระบบอย่างน้อยจำนวน 5 คลาส และแต่ละหน่วยตัวอย่างต้องพัฒนาภายใต้ภาษาโปรแกรมเดียวกัน
- นำหน่วยตัวอย่างจำนวน 5 หน่วยตัวอย่าง มาเปลี่ยนแปลงโครงสร้างคลาสให้เป็นโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชัน และโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน

- ออกแบบและพัฒนาเครื่องมือสำหรับอ่านกระบวนการทำงานของซอฟต์แวร์หน่วยตัวอย่างและหาค่าประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล
- จากนั้นนำหน่วยตัวอย่างที่ถูกพัฒนาเป็นซอฟต์แวร์ที่อยู่ในรูปแบบโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันไปประมวลผล เพื่อจัดเก็บค่าประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล
- นำข้อมูลที่จัดเก็บได้มาวิเคราะห์ประสิทธิภาพของซอฟต์แวร์ขณะประมวลผลระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน
- สรุปผลของการวิจัย

1.4.2 การเปรียบเทียบจำนวนคลาสและจำนวนเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์

โดยขั้นตอนการดำเนินการเปรียบเทียบจำนวนคลาสและจำนวนเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน มีดังนี้

- ศึกษารูปแบบและลักษณะการเปลี่ยนแปลงของซอฟต์แวร์เชิงวัตถุ
- เลือกซอฟต์แวร์กรณีศึกษาสำหรับใช้ในงานวิจัย โดยกำหนดให้เป็นซอฟต์แวร์ทางธุรกิจ ที่พัฒนาโดยใช้หลักการเชิงวัตถุ
- นำซอฟต์แวร์กรณีศึกษาที่เลือก มาปรับเปลี่ยนให้อยู่ในรูปแบบของโครงสร้างในความสัมพันธ์แบบแอสโซซิเอชันและโครงสร้างในความสัมพันธ์แบบเจเนอรัลไลเซชัน
- กำหนดความต้องการเชิงฟังก์ชัน เพื่อใช้ในการพิจารณาจำนวนคลาสและจำนวนเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการ ซึ่งความต้องการที่กำหนดจะอยู่ภายใต้ลักษณะการเปลี่ยนแปลงของซอฟต์แวร์เชิงวัตถุ
- ออกแบบและพัฒนาเครื่องมือสำหรับจัดเก็บจำนวนคลาสและจำนวนเมธอดที่ได้รับผลกระทบจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์
- ทำการทดลองโดยการปรับปรุง แก๊ซซอฟต์แวร์กรณีศึกษา โดยกำหนดหน่วยตัวอย่างเพื่อใช้ในการเปลี่ยนแปลงซอฟต์แวร์ในรูปแบบของความต้องการเชิงฟังก์ชัน (เพื่อให้ซอฟต์แวร์กรณีศึกษามีการเปลี่ยนแปลง) แล้วจัดเก็บผลการเปลี่ยนแปลงของจำนวนคลาสที่เกิดขึ้น

- นำข้อมูลที่จัดเก็บได้มาวิเคราะห์เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชันระหว่างโครงสร้างคลาสในความสัมพันธ์แบบแอตโซซิเอชันและโครงสร้างคลาสในความสัมพันธ์แบบเอนอรัลไลเซชัน
- สรุปผลของการวิจัย และจัดทำเอกสารประกอบการวิจัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 สามารถบอกถึงความสัมพันธ์ ผลกระทบและประสิทธิภาพของซอฟต์แวร์ที่อาจเกิดขึ้นจากการออกแบบโครงสร้างคลาสที่ต่างกันกับคุณภาพของซอฟต์แวร์ เพื่อลดความเสี่ยงอันเกิดจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์ และลดความเสี่ยงจากการบำรุงรักษาซอฟต์แวร์
- 1.5.2 สามารถนำผลสรุปที่ได้จากการวิจัย มาเป็นแนวทางในการเลือกโครงสร้างคลาสที่เหมาะสมในการออกแบบและพัฒนาซอฟต์แวร์ทางธุรกิจ เพื่อรองรับการเปลี่ยนแปลงความต้องการที่อาจเกิดขึ้น
- 1.5.3 เพื่อให้องค์กรธุรกิจลดต้นทุนและค่าใช้จ่ายในการบำรุงรักษาซอฟต์แวร์ ที่เกิดจากผลกระทบในการเปลี่ยนแปลงซอฟต์แวร์และสามารถลดความเสี่ยงที่ซอฟต์แวร์จะถูกกระทบโดยไม่ได้ตั้งใจจากการเปลี่ยนแปลงความต้องการของซอฟต์แวร์