

บทที่ 1

บทนำ



## ความเป็นมาของปัญหา

โดยทั่วไปปัญหาการพัฒนาาระบบเก่าที่ผ่านมา จะพัฒนาระบบย่อยสำหรับแต่ละส่วนขององค์กรโดยไม่ได้คำนึงถึงภาพของระบบโดยรวม จึงมีเพิ่มข้อมูลและโปรแกรมของแต่ละระบบย่อยทำให้เกิดความซ้ำซ้อนของข้อมูล ตัวอย่างเช่น ในองค์กรขนาดกลางอาจมีหลายฝ่าย แต่ละฝ่ายก็อาจมีแผนกจัดซื้อของตัวเอง ในยุคก่อนที่ไม่นำคอมพิวเตอร์มาใช้ อาจจำเป็นต้องมีการเก็บข้อมูลซ้ำ ๆ กันบนกระดาษไว้ตามฝ่ายต่าง ๆ เพื่อเกิดความสะดวกในการใช้งาน การเกิดความซ้ำซ้อนของข้อมูล ก็ไม่ก่อให้เกิดปัญหามากนัก อาจจะเป็นวิธีที่เหมาะสมในสมัยนั้นก็ได้อีก แต่ในยุคที่นำคอมพิวเตอร์มาใช้ ควรจะใช้ให้เกิดประโยชน์สูงสุดตามที่ควรจะเป็น จึงควรจัดทำโปรแกรมจัดซื้อให้เหมือนกันและใช้งานร่วมกันได้ ถ้าจัดให้ในแต่ละแผนกจัดซื้อมีโปรแกรมของตนเอง ซึ่งต้องดูแลแทนที่จะเป็นโปรแกรมเดียว นอกจากนี้ก็จะมีเพิ่มข้อมูลที่มีโครงสร้างต่างกันอยู่ในแต่ละแผนกจัดซื้อ ความเข้ากันไม่ได้ของโปรแกรมและเพิ่มข้อมูล จะทำให้เกิดปัญหาที่จะนำข้อมูลทั้งหมดมาใช้เพื่อการบริหาร

ระบบสารสนเทศในช่วงแรก ๆ มักจะได้รับการพัฒนาโดยไม่คำนึงถึงระบบต่าง ๆ ที่เกี่ยวข้อง ดังนั้นการใช้ระบบเหล่านั้นทำงานร่วมกันจึงเป็นเรื่องยาก การทำให้ระบบทำงานประสานกันอย่างดีจะต้องการออกแบบระบบสารสนเทศแบบใหม่ ในยุคแรก ๆ ของระบบสารสนเทศ ในแต่ละระบบจะมีขั้นตอนการทำงานซึ่งผู้พัฒนาระบบนั้น ๆ เข้าใจเป็นอย่างดี แต่เข้าใจขั้นตอนของระบบอื่นน้อยมาก แต่ละระบบจะมีเพิ่มข้อมูลของระบบเอง เพิ่มเหล่านี้มีลักษณะเฉพาะตัวและสัมพันธ์โดยตรงกับระบบนั้น ๆ แต่ในการปฏิบัติงานโดยทั่วไป มีความจำเป็นต้องส่งผ่านข้อมูลระหว่างระบบ การบริหารต้องใช้ข้อมูลจากระบบต่าง ๆ ร่วมกัน ซึ่งข้อมูลเหล่านี้มักเข้ากันไม่ค่อยได้ และข้อเสียอีกอย่างก็คือ โครงสร้างเพิ่มข้อมูลมักถูกเปลี่ยนแปลงบ่อย ๆ และจะมีผลต่อเนื่องกับโปรแกรมที่ใช้ข้อมูลเหล่านั้น

เมื่อแต่ละระบบมีเพิ่มข้อมูลและโปรแกรมเป็นของตัวเอง ทำให้เกิดปัญหาการส่งผ่านข้อมูลระหว่างระบบ ดังนั้นวิธีการแก้ปัญหาก็คือการออกแบบข้อมูลแบบรวมศูนย์ หน้าที่การออกแบบจำลองข้อมูล (Data Model) เป็นของผู้ดูแลระบบสารสนเทศ ซึ่งแบบจำลองข้อมูลนี้จะสัมพันธ์กับทุกระบบ เมื่อแบบจำลองข้อมูลถูกแยกออกเป็นเพิ่มข้อมูล ซึ่งถูกใช้ร่วมกันระหว่างระบบต่าง ๆ และข้อมูลต่าง ๆ ถูกเก็บไว้ในระบบฐานข้อมูล การทำแบบจำลองข้อมูลเป็นปัจจัยพื้นฐานที่นำไปสู่ความสำเร็จ ถึงแม้ว่าโครงสร้างของเพิ่มข้อมูลจะซับซ้อน แต่การไหลของข้อมูลจะง่ายขึ้น ข้อมูลจะมีความถูกต้องตรงกันและแม่นยำ การเรียกใช้ข้อมูลจากระบบต่าง ๆ สามารถทำได้ และการเปลี่ยนขั้นตอนหรือวิธีการคำนวณทำได้ง่ายขึ้นเช่นกัน กระจายที่ใช้ส่งผ่านข้อมูลจะน้อยลง ขั้นตอนของระบบต่าง ๆ จะเก็บไว้ที่ส่วนกลาง และจะต้องมีการวิเคราะห์ออกแบบระบบแบบใหม่

เนื่องจากความต้องการของฝ่ายบริหาร อาจทำให้มีการแก้ไขโครงสร้างข้อมูลบ่อย ๆ ซึ่งการเก็บข้อมูลด้วยโครงสร้างเพิ่มข้อมูลแบบเดิม เช่น เพิ่มข้อมูลในภาษาโคบอล โปรแกรมที่เกี่ยวข้องกับเพิ่มข้อมูลนั้นจะต้องถูกแก้ไขโครงสร้างตาม และทำการรวบรวมใหม่ซึ่งเสียเวลา ระบบฐานข้อมูลสมัยใหม่เมื่อมีการแก้ไขโครงสร้างข้อมูล โปรแกรมที่เกี่ยวข้องก็จะไม่ต้องแก้ไข เรียกว่า ความเป็นอิสระจากโครงสร้างข้อมูล

ถึงแม้ว่าจะใช้ระบบฐานข้อมูลแล้ว การออกแบบข้อมูลหากไม่ได้ทำข้อมูลให้เป็นบรรทัดฐาน (Normalize) มีการแก้ไขความสัมพันธ์ระหว่างเพิ่มข้อมูล ย่อมมีผลกระทบกับโปรแกรมต้องแก้ไขตามด้วย ผู้ออกแบบระบบจึงควรจัดความสัมพันธ์ของเพิ่มข้อมูลให้ดี เพื่อให้เกิดความรวดเร็ว ความถูกต้อง ลดความซับซ้อนของข้อมูล และลดปริมาณการบำรุงรักษาโปรแกรมให้น้อยลง

จากปัญหาที่กล่าวมา จึงเกิดแนวความคิดในการพัฒนาซอฟต์แวร์ที่เรียกว่า วิศวกรรมซอฟต์แวร์ (Software Engineering) Fritz Bauer (Pressman,1992) ได้ให้คำนิยามของ วิศวกรรมซอฟต์แวร์ ไว้ว่า หมายถึง การสร้างและการใช้วิธีทางวิศวกรรมเพื่อสร้างซอฟต์แวร์ ที่มีความคุ้มค่าทางเศรษฐศาสตร์ และซอฟต์แวร์ที่สร้างขึ้นมามีความเชื่อถือได้ และทำงานอย่างมีประสิทธิภาพ

องค์ประกอบของ วิศวกรรมซอฟต์แวร์ ที่สำคัญมีอยู่ด้วยกัน 3 ส่วน

1. วิธีการทางวิศวกรรมซอฟต์แวร์ หมายถึงขั้นตอนในการพัฒนาซอฟต์แวร์ ตัวอย่างเช่น การวางแผน การศึกษาความต้องการของผู้ใช้ การวิเคราะห์และออกแบบ การเขียนโปรแกรม และการ

จัดทำเอกสารประกอบ ซึ่งขั้นตอนต่าง ๆ เหล่านี้จะนิยมใช้แผนภาพ หรือการเขียนบรรยายอย่างมีรูปแบบ เพราะจะทำให้ง่ายต่อการสื่อความหมายในกลุ่มผู้พัฒนาซอฟต์แวร์ และผู้ใช้ซอฟต์แวร์

2. เครื่องมือทางวิศวกรรมซอฟต์แวร์ หมายถึง เครื่องมือชนิดอัตโนมัติ หรือกึ่งอัตโนมัติที่ใช้ในระหว่างขั้นตอนการพัฒนาซอฟต์แวร์ และถ้าหากมีการนำเครื่องมือเหล่านี้ มาพัฒนาเป็นโปรแกรมคอมพิวเตอร์ โปรแกรมเหล่านี้จะถูกเรียกว่าเคส (CASE ย่อจาก Computer Aided Software Engineering)

3. วิธีปฏิบัติทางวิศวกรรมซอฟต์แวร์ หมายถึงการบอกรายละเอียดวิธีปฏิบัติ ในแต่ละขั้นตอนของการพัฒนาซอฟต์แวร์ ซึ่งมีประโยชน์ในการบริหารและควบคุมการพัฒนาซอฟต์แวร์ ให้เป็นไปตามระยะเวลา ค่าใช้จ่าย และความต้องการของผู้ใช้

เนื่องจากปัจจุบันมีการใช้เคสกันอย่างแพร่หลาย เพื่อช่วยในการพัฒนาซอฟต์แวร์ ซึ่งประโยชน์ที่ได้รับจากเคสก็คือ การพัฒนาระบบเป็นไปอย่างรวดเร็ว ลดข้อผิดพลาดในการทำงาน ลดต้นทุนการพัฒนาโปรแกรม ลดแรงงานของโปรแกรมเมอร์

แม้ว่าจะได้มีการพัฒนาเคสขึ้นมาเพื่อช่วยพัฒนาระบบมากแล้วก็ตาม แต่เคสส่วนใหญ่จะยึดวิธีการอื่น ๆ ในการสร้างระบบข้อมูลเพื่อการบริหาร เช่น วัฏจักรการพัฒนากระบวน (SDLC : System Development Life Cycle) หรือ การวิเคราะห์เชิงวัตถุ (Object Oriented) ยังไม่ปรากฏว่ามีการพัฒนาเคส เพื่อการปรับปรุงโครงสร้างขององค์กร (Reengineering) โดยอาศัยวิธีการวางแผนกลยุทธ์สารสนเทศ (Information Strategic Planning ย่อว่า ISP) อย่างจริงจัง ทั้งที่โครงสร้างปัจจุบันของบริษัทเป็นปัญหาต่อความอยู่รอดของบริษัทเอง ในหลาย ๆ บริษัท โครงการวิจัยนี้เป็นโครงการวิจัยเพื่อสร้างเคสขึ้นมาสำหรับ การปรับปรุงโครงสร้างขององค์กร เพื่อให้ผู้เกี่ยวข้องได้รับข้อมูลสารสนเทศที่สำคัญในการวัดผลการดำเนินงานของบริษัทอย่างรวดเร็ว ทั้งนี้จะได้ปรับเปลี่ยนวิธีการดำเนินงานของบริษัทได้ทันต่อเหตุการณ์ เนื่องจากในสภาวะปัจจุบันและแนวโน้มในอนาคต จะมีการแข่งขันทางธุรกิจอย่างสูง และความก้าวหน้าของเทคโนโลยีที่รวดเร็ว ดังนั้นโครงการพัฒนาเคสเพื่อการปรับโครงสร้างองค์กร จึงเป็นโครงการที่เป็นประโยชน์ต่อการดำเนินงานของบริษัททั้งในปัจจุบันและอนาคต

### วัตถุประสงค์ของวิทยานิพนธ์

1. เพื่อศึกษาหลักการจัดลำดับชั้นข้อมูล (Information Class) ตามหลักการของการวางแผนกลยุทธ์สารสนเทศ
2. เพื่อศึกษาขั้นตอนการสร้างตารางความสัมพันธ์เบื้องต้น ระหว่างหน้าที่ในหน่วยย่อยขององค์กรและข้อมูล
3. เพื่อศึกษาขั้นตอนและทฤษฎีที่ใช้ในการคำนวณและจัดเรียงตารางความสัมพันธ์ของหน้าที่และข้อมูล เพื่อให้เหมาะสมต่อการปรับปรุงโครงสร้างองค์กร (Reengineering)
4. เพื่อพัฒนาเครื่องมือเอส เพื่อใช้ปรับปรุงโครงสร้างองค์กร (Reengineering)

### ขอบเขตของวิทยานิพนธ์

1. พัฒนาโปรแกรมโดยใช้ภาษาวิซวลเบสิก (Visual Basic) ภายใต้ระบบปฏิบัติการวินโดวส์ (Windows) บนเครื่องไมโครคอมพิวเตอร์ ใช้ไมโครโปรเซสเซอร์ตั้งแต่ 80486 ขึ้นไป โดยมีหน่วยความจำอย่างน้อย 4 MB
2. พัฒนาโปรแกรมในส่วนของการวิเคราะห์ความสัมพันธ์ของข้อมูล เพื่อใช้ในการจัดลำดับใหม่
3. พัฒนาโปรแกรมในส่วนของการลำดับหน้าที่ในหน่วยย่อยขององค์กรและการรวมกลุ่ม (Grouping) เพื่อให้เกิดความเหมาะสมในการทำกิจกรรมขององค์กรตามทฤษฎีการวางแผนกลยุทธ์สารสนเทศ
4. โปรแกรมที่พัฒนาขึ้นนี้ ในส่วนการจัดเรียงข้อมูลจะต้องสามารถทำงานได้โดยอัตโนมัติ และเป็นระบบแมนนวล เพื่อให้ผู้ใช้โปรแกรมสามารถจัดเรียงโดยอาศัยประสบการณ์ของผู้ใช้เอง

### ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาขั้นตอนการพัฒนาาระบบสารสนเทศ โดยใช้วิธีการวางแผนกลยุทธ์สารสนเทศ (Information Strategic Planning)
2. ศึกษาวิธีการวิเคราะห์และจัดเก็บข้อมูล เพื่อให้ได้มาซึ่งตารางแสดงความสัมพันธ์ระหว่างข้อมูลและหน้าที่

3. ศึกษาขั้นตอนและทฤษฎีที่ลำดับความสัมพันธ์ของข้อมูลและหน้าที่ตามทฤษฎีการวางแผนกลยุทธ์สารสนเทศ
4. ศึกษาการเขียนโปรแกรมวิซวลเบสิก (Visual Basic) ภายใต้ระบบปฏิบัติการวินโดวส์
5. ออกแบบวิธีการทำงานของโปรแกรม ตลอดจนการติดต่อระหว่างผู้ใช้โปรแกรม (User Interface) เพื่อให้เกิดความคล่องตัวระหว่างการทำงาน
6. พัฒนาโปรแกรม เพื่อให้สามารถนำขั้นตอนและทฤษฎีในการเรียงลำดับความสำคัญระหว่างข้อมูลและหน้าที่
7. ทดลองและปรับปรุงให้โปรแกรมดียิ่งขึ้น
8. สรุปผลการทำวิทยานิพนธ์และจัดทำรายงาน

#### ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้เกิดการเรียนรู้ขั้นตอนการพัฒนาระบบสารสนเทศเพื่อการบริหารได้ดียิ่งขึ้นตามทฤษฎีการวางแผนกลยุทธ์สารสนเทศ
2. ทำให้เกิดความรู้ ความเข้าใจในการพัฒนาระบบ ที่นำไปสู่การปรับปรุงกิจการภายในองค์กร เพื่อเพิ่มประสิทธิภาพการทำงานและทำงานตรงเป้าหมายขององค์กรได้ดียิ่งขึ้น
3. เรียนรู้การเขียนโปรแกรมด้วยวิซวลเบสิกภายใต้ระบบวินโดวส์ ตลอดจนเครื่องมือ (Toolkits) ต่าง ๆ เพื่อการพัฒนาโปรแกรม
4. เป็นแนวทางในการพัฒนา CASE Tool ต่อไป

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย