

บทที่ 1

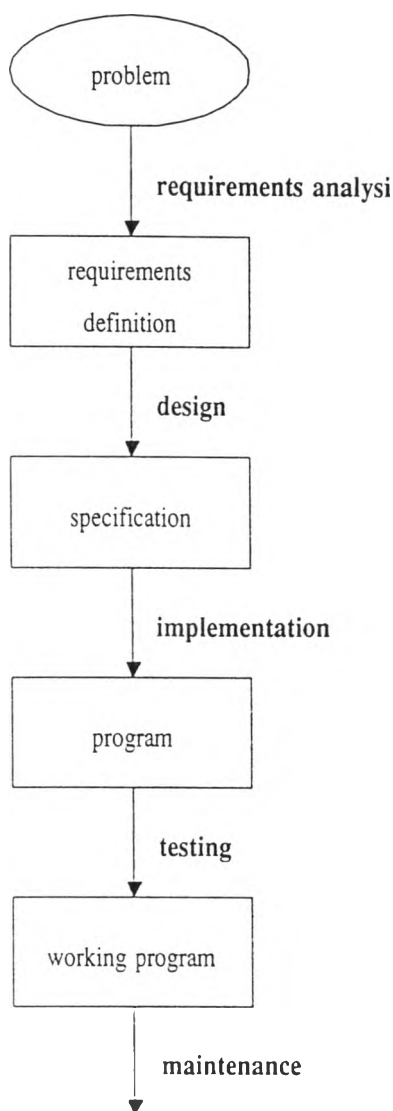
บทนำ



ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันธุรกิจประเภทต่าง ๆ มีการแข่งขันกันมากองค์กรต่าง ๆ ได้มีการนำระบบคอมพิวเตอร์เข้ามาช่วยสนับสนุนงานขององค์กร เพื่อช่วยในการบริหาร และการจัดการในด้านต่าง ๆ ซึ่งการนำเอาระบบคอมพิวเตอร์เข้ามาช่วยนี้จำเป็นต้องใช้ซอฟต์แวร์ที่มีคุณภาพสามารถรองรับระบบงานขององค์กรได้อย่างมีประสิทธิภาพ ไม่ว่าจะเป็นในด้านความถูกต้องของข้อมูล ความรวดเร็วในการตอบสนองการใช้งาน ความปลอดภัยของข้อมูล ความยืดหยุ่นในการแก้ไขเปลี่ยนแปลง ความสะดวกในการใช้งานและการบำรุงรักษา และการจัดทำเอกสารต่าง ๆ ที่เกี่ยวข้อง เป็นต้น ซึ่งแต่ละองค์กรอาจจะหาซื้อซอฟต์แวร์เข้ามาใช้หรือทำการพัฒนาซอฟต์แวร์ขึ้นมาเองก็ตาม แต่การที่จะได้มาซึ่งซอฟต์แวร์ที่มีประสิทธิภาพนั้น การพัฒนาซอฟต์แวร์ดังกล่าวต้องมีประสิทธิภาพด้วยเช่นกัน

ในการพัฒนาซอฟต์แวร์นั้นมีหลักการต่าง ๆ คล้ายคลึงกับหลักการทางวิศวกรรมทั่วไป กล่าวคือ ต้องมีการวิเคราะห์หาความต้องการแล้วจึงทำการกำหนดความต้องการเหล่านั้นออกมาให้ชัดเจน เพื่อนำมาใช้ในการออกแบบให้เหมาะสม และ ตรงกับความต้องการที่ได้กำหนดไว้ จากนั้นจึงเริ่มทำการพัฒนาให้เกิดผลตามที่ได้ทำการออกแบบเอาไว้ ขั้นตอนต่าง ๆ ที่ใช้เป็นพื้นฐานในการพัฒนาซอฟต์แวร์สามารถแสดงไว้ดังรูปที่ 1.1 ได้แก่ขั้นตอนการวิเคราะห์หาความต้องการ (requirements analysis) ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ การกำหนดความต้องการ (requirements definition) ขั้นตอนการออกแบบ (design) ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ ข้อกำหนดคุณลักษณะ (specifications) ขั้นตอนการทำให้เกิดผล (implementation) ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ โปรแกรมต่าง ๆ (program) ขั้นตอนการตรวจสอบ (testing) ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ โปรแกรมที่ถูกต้องพร้อมใช้งาน (working program) และขั้นตอนการบำรุงรักษา (maintenance)



รูปที่ 1.1 แสดงขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาซอฟต์แวร์

จากขั้นตอนต่าง ๆ ที่กล่าวมานี้จะเห็นได้ว่าเราสามารถที่จะนำเอาขั้นตอนเหล่านี้มาใช้เป็นแนวทางในการพัฒนาซอฟต์แวร์ได้ แต่ว่าในทางปฏิบัติจริง ๆ แล้วในแต่ละขั้นตอนของการพัฒนาซอฟต์แวร์นั้นยังมีปัจจัยอีกหลาย ๆ อย่างที่เกี่ยวข้องซึ่งยังไม่มีมีการกล่าวถึงในแต่ละขั้นตอน ได้แก่ การศึกษาและประสบการณ์ของผู้พัฒนาซอฟต์แวร์ และผู้ใช้ สภาพแวดล้อมที่เกี่ยวข้อง ลักษณะของสังคมและวัฒนธรรมในแต่ละประเทศหรือองค์กร ซึ่งปัจจัยเหล่านี้ อาจทำให้เกิดปัญหาขึ้นในการพัฒนาซอฟต์แวร์ทำให้การพัฒนาซอฟต์แวร์ดังกล่าวไม่ประสบผลสำเร็จตามที่ได้คาดหวังเอาไว้ ปัญหาที่มักจะพบบ่อย ๆ ในการพัฒนาซอฟต์แวร์ได้แก่

1. ปัญหาที่เกิดจากการศึกษาและประสบการณ์ของผู้พัฒนาซอฟต์แวร์ และผู้ใช้

1.1 ในแง่ของผู้พัฒนาซอฟต์แวร์

เนื่องจากจากผู้พัฒนาระบบส่วนใหญ่จะไม่ได้เป็นผู้ที่ใช้ระบบที่ตัวเองพัฒนา จึงมีความรู้ในเรื่องระบบที่จะพัฒนาไม่ละเอียดเท่ากับผู้ใช้งานอยู่จริง ๆ การพัฒนาระบบดังกล่าวจึงอาจจะไม่สามารถรองรับความต้องการของผู้ใช้ได้อย่างครบถ้วน และ ปัญหาอีกประการหนึ่งก็คือ ลักษณะมุมมองระบบที่จะพัฒนานั้นผู้พัฒนาระบบมักจะมีมุมมองในเชิงเทคนิค การติดต่อสื่อสารกับผู้ใช้จึงอาจทำให้เกิดความเข้าใจที่ไม่ตรงกัน หรือสื่อสารกันไม่รู้เรื่องทำให้ข้อมูลที่ได้ไม่ถูกต้อง จึงทำให้เกิดปัญหาขึ้นในภายหลัง

1.2 ในแง่ของผู้ใช้ระบบ

ในบางครั้งผู้ใช้ระบบเองก็อาจที่จะยังไม่ทราบความต้องการที่ชัดเจนว่าต้องการได้ อะไรจากระบบบ้างจึงให้ข้อมูลได้ไม่ครบถ้วน หรือบางทีก็คาดหวังเอาไว้ว่าผู้พัฒนาระบบนั้นมีความรู้เรื่องเกี่ยวกับระบบที่จะพัฒนาเป็นอย่างดีจึงเห็นด้วยกับวิธีการที่ผู้พัฒนาเสนอมา ซึ่งบางทีอาจจะไม่ตรงกับความต้องการของตนก็ได้ ปัญหาอีกประการหนึ่งก็คือปัญหาเกี่ยวกับการไม่ยอมรับการเปลี่ยนแปลง เนื่องจากผู้ใช้ซึ่งส่วนใหญ่จะอยู่ในระดับปฏิบัติการนั้นมีความคิดว่าวิธีการเดิมที่เคยทำอยู่นั้นดีอยู่แล้ว ส่วนวิธีการใหม่นั้นอาจทำให้เกิดความยุ่งยากในการเรียนรู้ทำให้ไม่คล่องตัวในการทำงาน จึงไม่ค่อยให้ความร่วมมือกับทีมงานที่มาหาข้อมูลเพื่อนำไปพัฒนาระบบ ซึ่งอาจทำให้การพัฒนาระบบดังกล่าวเกิดความล่าช้า หรือล้มเหลวก็ได้จะเห็นได้ว่าปัญหาต่าง ๆ ที่ได้กล่าวมานี้จะมีมากหรือน้อยนั้นขึ้นอยู่กับความรู้ และประสบการณ์ของผู้พัฒนาซอฟต์แวร์และผู้ใช้ระบบ

2. ปัญหาที่เกิดจากสภาพแวดล้อม

ในบางครั้งเมื่อเรามีการนำซอฟต์แวร์ประยุกต์จากต่างประเทศเข้ามาใช้นั้นปรากฏว่าเรายังไม่สามารถที่จะนำซอฟต์แวร์นั้นมาใช้ได้ทันที จำเป็นต้องมีการปรับปรุงแก้ไขซอฟต์แวร์ให้เข้ากับลักษณะงานขององค์กร หรือต้องมีการจัดการฝึกอบรมการใช้ซอฟต์แวร์นั้นขึ้นจะเห็นได้ว่าปัญหาดังกล่าวเกิดจากลักษณะสภาพแวดล้อมที่แตกต่างกันเนื่องจากว่าซอฟต์แวร์ที่ถูกพัฒนาจากต่างประเทศนั้นจะพัฒนาจากพื้นฐานความต้องการของคนในประเทศของผู้พัฒนาซึ่งอาจไม่ตรงกับความต้องการขององค์กรในประเทศไทย

3. ปัญหาที่เกิดจากลักษณะทางสังคมและวัฒนธรรม

ปัจจุบันจะเห็นได้ว่าคนที่มีความรู้เกี่ยวกับระบบคอมพิวเตอร์นั้นมักจะไม่ค่อยรู้จักลักษณะงานหรือการทำงานของผู้ใช้และในทำนองเดียวกันผู้ใช้ส่วนใหญ่ก็มักจะไม่ค่อยรู้ระบบคอมพิวเตอร์หรือถ้ารู้ก็ไม่รู้ว่าจะนำคอมพิวเตอร์มาช่วยงานได้อย่างไร เนื่องจากว่าการเรียนการสอนในสถาบันต่าง ๆ ไม่ได้เน้นถึงการนำซอฟต์แวร์มาใช้ให้เหมาะสมกับลักษณะแวดล้อมทางสังคม

จากปัญหาที่ได้กล่าวมาจะเห็นได้ว่าการพัฒนาซอฟต์แวร์นั้น ถึงแม้ว่าเราจะมีรูปแบบขั้นตอนที่ใช้ในการพัฒนาแล้วก็ตาม แต่ก็ยังมีปัจจัยอื่น ๆ อีกหลายอย่างที่ต้องคำนึงถึงดังได้กล่าวมาแล้วในข้างต้น ซึ่งปัญหาดังกล่าวอาจทำให้การพัฒนาซอฟต์แวร์นั้นไม่เป็นไปตามที่ได้วางแผนไว้ ดังนั้นหากว่าเรามีรูปแบบหรือขั้นตอนเพื่อใช้ในการตรวจสอบว่า การพัฒนาซอฟต์แวร์นั้นได้เป็นไปตามแผนงานและขั้นตอนต่าง ๆ ที่ได้วางไว้แล้วเชื่อได้ว่าเราจะได้ซอฟต์แวร์ที่มีคุณภาพเป็นที่ยอมรับจากผู้ใช้งาน

วัตถุประสงค์

1. เพื่อออกแบบขั้นตอน และ วิธีการในการติดตาม และ ตรวจสอบการพัฒนาซอฟต์แวร์ประยุกต์
2. เพื่อพัฒนาซอฟต์แวร์ในการติดตาม และ ตรวจสอบการพัฒนาซอฟต์แวร์ประยุกต์

ขอบเขตของวิทยานิพนธ์

1. ศึกษารูปแบบขั้นตอน และ วิธีการตรวจสอบการพัฒนาซอฟต์แวร์ของบริษัทฯ ที่พัฒนาซอฟต์แวร์ในประเทศไทย 3 บริษัทฯ
2. การพัฒนาซอฟต์แวร์เพื่อใช้ในการตรวจสอบนี้ จะทำการออกแบบการตรวจสอบโดยอาศัย แนวทางจากมาตรฐานไอทีเอฟเฟลอี (IEEE Std 1028-1988, IEEE Standard for Software Reviews and Audits.)
3. การพัฒนาซอฟต์แวร์นี้จะออกแบบเพื่อการติดตามและตรวจสอบในเรื่องต่อไปนี้
 - 3.1 การตรวจสอบในขั้นตอนการดำเนินการ (In-Process Audit)
 - 3.2 การตรวจสอบหน้าที่ (The Functional Audit)
 - 3.3 การตรวจสอบคุณภาพของระบบ (The Quality Systems Audit)

4. ระบบนี้จะสามารถใช้ติดตามตรวจสอบแผนการพัฒนาซอฟต์แวร์ประยุกต์ และ องค์ประกอบของการพัฒนาได้อย่างต่อเนื่อง

5. การออกแบบขั้นตอนการตรวจสอบการพัฒนาซอฟต์แวร์นี้ จะไม่รวมถึงการตรวจสอบการพัฒนาซอฟต์แวร์ ที่ใช้เคสทูลต่างๆมาช่วยในการพัฒนา

ขั้นตอนและวิธีดำเนินการวิทยานิพนธ์

1. ศึกษารูปแบบและขั้นตอนต่างๆในการพัฒนาซอฟต์แวร์ประยุกต์
2. รวบรวมสิ่งที่จะต้องตรวจสอบในแต่ละขั้นตอนของการพัฒนาซอฟต์แวร์ประยุกต์
3. รวบรวมข้อมูล และ วิธีการ ที่ใช้ในการตรวจสอบการพัฒนาซอฟต์แวร์จากตำราต่างๆ หรือ เอกสารที่เกี่ยวกับมาตรฐานต่าง ๆ
4. รวบรวม และ ศึกษารูปแบบขั้นตอนในการตรวจสอบการพัฒนาซอฟต์แวร์จากบริษัทที่มีการพัฒนาซอฟต์แวร์ในประเทศไทย
5. วิเคราะห์หาแนวทางต่างๆเกี่ยวกับการตรวจสอบการพัฒนาซอฟต์แวร์ในแต่ละขั้นตอน
6. ออกแบบขั้นตอนในการตรวจสอบการพัฒนาซอฟต์แวร์
7. สรุปผลการวิจัย และ ข้อเสนอแนะ

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้หลักการและวิธีการที่ถูกต้องในการพัฒนาซอฟต์แวร์
2. ได้ขั้นตอนการตรวจสอบเพื่อใช้ในการพัฒนาซอฟต์แวร์
3. เป็นแนวทางเพื่อให้ผู้ตรวจสอบนำไปใช้ในการตรวจสอบการพัฒนาซอฟต์แวร์