บทที่ 7
ทางเลือกการพัฒนาระบบ
(Alternatives Systems Development Methodologies)

เนื้อหาภายในบท

7.1 บทนำ

ปัญหาที่เกิดขึ้นกับการพัฒนาระบบแบบโครงสร้าง

ทางเลือกในการพัฒนาระบบ

7.2 การทำต้นแบบ

ประสิทธิภาพของการทำต้นแบบ

ประเภทของต้นแบบ

ขั้นตอนการพัฒนาต้นแบบ

7.3 การพัฒนาโปรแกรมของผู้ใช้ขั้นปลาย

ประเภทของผู้ใช้ขั้นปลาย

ผู้ใช้ขั้นปลายกับการวิเคราะห์และออกแบบระบบ

ความเสี่ยงและปัญหาที่เกิดจากการให้ผู้ใช้ขั้นปลายพัฒนาระบบ

7.4 เครื่องมือซอฟต์แวร์ของผู้ใช้ขั้นปลาย

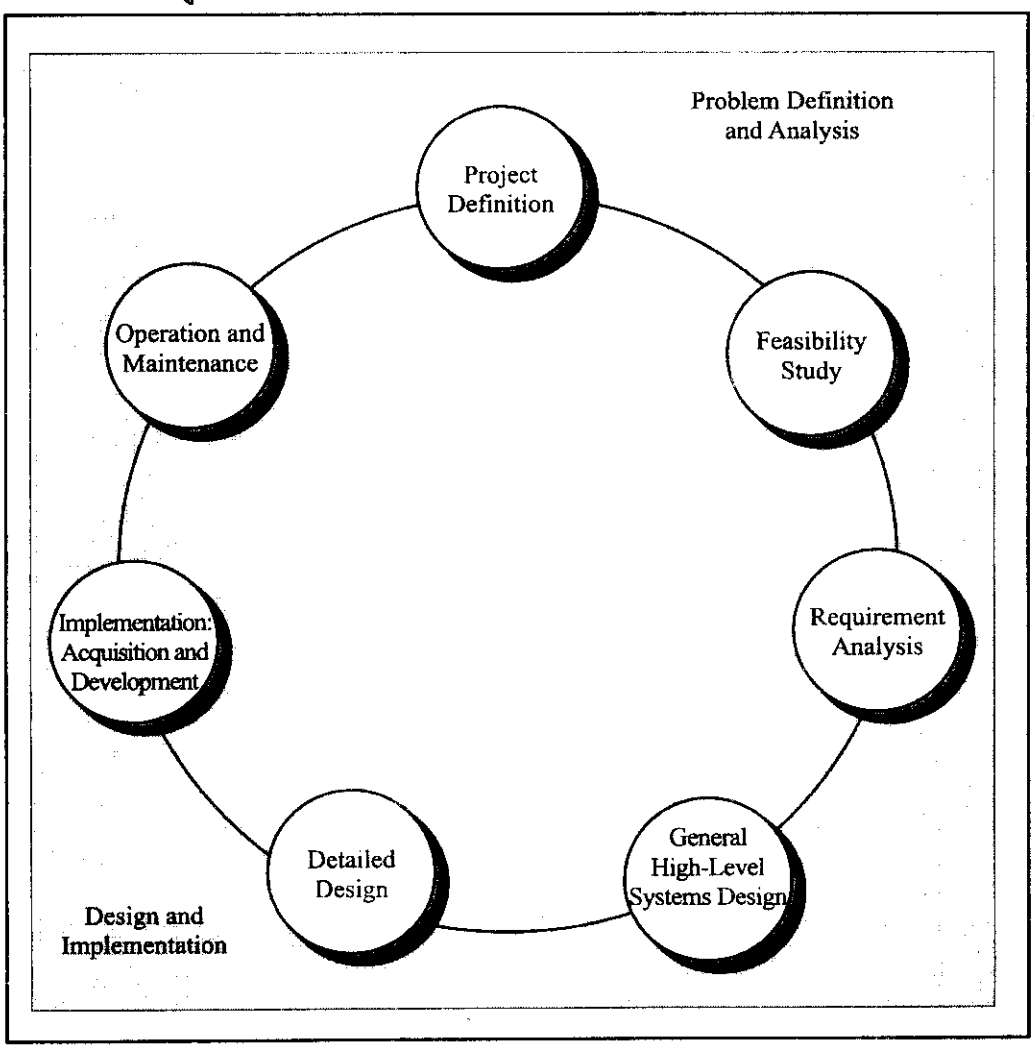
คำถามท้ายบท

วัตถุประสงค์ประจำบท

1. สามารถอธิบายสาเหตุที่ทำให้กระบวนการพัฒนาระบบแบบโครงสร้างเกิดความล่าช้า
2. สามารถอธิบายความแตกต่างระหว่างการทำตัวแบบ (prototype) และการพัฒนางานของผู้ใช้ขั้นปลาย (EUC)
3. สามารถอธิบายคุณลักษณะของตัวแบบและตัวแบบประเภทใดที่เรียกเป็นระบบสาธิต (demo system)
4. สามารถอธิบายปัญหาที่เกิดขึ้นในการทำตัวแบบ
5. สามารถบอกเครื่องมือทำตัวแบบ
6. สามารถอธิบายความหมายของผู้ใช้ขั้นปลายและ EUC
7. สามารถอธิบายประเภทของผู้ใช้ขั้นปลาย
8. สามารถเปรียบเทียบการพัฒนางานของผู้ใช้ขั้นปลายและกระบวนการพัฒนาแบบโครงสร้าง
9. สามารถอธิบายความเสี่ยงและปัญหาที่เกิดขึ้นกับ EUC
10. สามารถอธิบายความสัมพันธ์ระหว่าง IC และ EUC

7.1 บทนำ

ในบทที่ผ่านมา ได้กล่าวถึงวิธีการพัฒนาระบบสารสนเทศโดยด้วยคอมพิวเตอร์ วิธีที่รู้จักกันแพร่หลายก็คือการพัฒนาระบบแบบโครงสร้าง (structured system development) ประกอบด้วยกิจกรรมต่างๆ ได้แก่ การวางแผน การวิเคราะห์ความต้องการ การออกแบบและการทำให้เกิดผลและการบำรุงรักษาระบบ เป็นต้น หากระบบใหม่ต้องการซอฟต์แวร์ก็ต้องออกแบบและพัฒนาโปรแกรม ในทางตรงข้ามหากเป็นการสั่งซื้อซอฟต์แวร์ นักวิเคราะห์ระบบต้องมีความรู้ด้านซอฟต์แวร์ทางธุรกิจที่จะนำมาใช้ หากระบบสารสนเทศระบบใหม่ต้องการฮาร์ดแวร์ การสั่งซื้อฮาร์ดแวร์ก็จะรวมอยู่ในส่วนนี้ด้วย



รูป 7-1 วัฏจักรการพัฒนาระบบ

กระบวนการพัฒนาระบบมีการทำงานแบ่งออกเป็น 5 เฟส แต่ละเฟสขึ้นอยู่กับเฟสก่อนหน้าคือต้องทำตามลำดับ การทำงานทั้ง 5 เฟสนี้เรียกว่าวัฏจักรการพัฒนาระบบหรือวัฏจักรการวิเคราะห์และออกแบบระบบแบบโครงสร้าง (structured systems analysis and design cycle) แสดงดังรูป 7-1 วัฏจักรการพัฒนาระบบเริ่มจากการให้คำนิยามโครงการ (project definition) เมื่อผู้ใช้พบว่าระบบงานที่ใช้อยู่เกิดปัญหาหรือผู้ใช้มีความต้องการใช้ระบบงานใหม่แทนระบบงานเดิม การให้คำนิยามโครงงานจึงกำหนดขึ้นและมีการศึกษาความเป็นไปได้ของการพัฒนาระบบและการปฏิบัติงาน ซึ่งวิเคราะห์จากด้านกายภาพและความเข้าใจทางตรรกะของปัญหา ในส่วนของเฟสต้นๆ มีกิจกรรมด้านการกำหนดความต้องการ การศึกษาความเป็นไปได้ และการวิเคราะห์ความต้องการ นักวิเคราะห์ระบบเป็นผู้ที่ทำกิจกรรมต่างๆ ได้แก่ การเก็บรวบรวมข้อมูล ตัดสินใจว่าปัญหาใดบ้างที่ต้องแก้ไข กำหนดวิธีการแก้ปัญหาที่เป็นไปได้ แจกแจงความต้องการ ศึกษาความต้องการและทำความเข้าใจรายละเอียดของระบบ ในเฟสของการออกแบบทั่วไปนั้น นักวิเคราะห์ระบบทำการศึกษาความต้องการและเริ่มออกแบบระบบทางกายภาพ ในเฟสนี้มีการพัฒนาและประเมินทุกทางเลือกแต่มีเพียงทางเลือกเดียวที่ได้รับคัดเลือก เฟสการออกแบบโดยละเอียดมีการพัฒนาคุณลักษณะต่างๆเพื่อเป็นทางเลือก ซึ่งรวมถึงลักษณะของซอฟต์แวร์ที่ต้องการ (ได้กล่าวไว้ในบทที่ 5) หรือการได้มาของระบบ (ได้กล่าวไว้ในบทที่ 6) เฟสการทำให้เกิดผลเป็นการให้ได้มาซึ่งซอฟต์แวร์หรือการพัฒนาซอฟต์แวร์ตามที่ได้ออกแบบและการติดตั้งซอฟต์แวร์ เฟสการบำรุงรักษาระบบเป็นการเฝ้าติดตามดูประสิทธิภาพของระบบ เฝ้าดูปัญหาที่เกิดขึ้นประจำวันและบันทึกการทำงานของระบบ

ปัญหาที่เกิดขึ้นกับการพัฒนาระบบแบบโครงสร้าง
(Problems with the structured Approach)

การวิเคราะห์และออกบบแบบโครงสร้าง มีวิวัฒนาการมาหลายทศวรรษเพื่อตอบสนองต้นทุนที่เพิ่มขึ้นและระบบมีขนาดใหญ่และซับซ้อนมากขึ้น แนวโน้มการใช้คอมพิวเตอร์ที่นำไปสู่การพัฒนาระบบมี 2 ประการ ประการแรกคือ การใช้เครื่องคอมพิวเตอร์ส่วนบุคคลมีอัตราเพิ่มสูงขึ้นและ มีการใช้เครื่องปลายทางเข้าถึงเครื่องเมนเฟรม ผู้ใช้พบว่ามีหลายหนทางในการนำคอมพิวเตอร์มาใช้ในการทำงานได้อย่างมีประสิทธิภาพ ประการที่สองคือ ในขณะเดียวกันนั้นผู้ใช้พบว่าคอมพิวเตอร์สามารถช่วยงานได้เป็นอย่างมาก มีแผนกระบบสารสนเทศในองค์การซึ่งมีศักยภาพในการให้บริการ ทำหน้าที่ตอบสนองความต้องการของผู้ใช้ แนวโน้มทั้งสองประการนี้เองทำให้ผู้ใช้เกิดความคาดหวังกับแผนกระบบสารสนเทศว่าสามารถตอบสนองด้านการพัฒนาซอฟต์แวร์ตาม

ลักษณะงานประยุกต์ให้กับผู้ใช้ แต่บางครั้งแผนกระบบสารสนเทศไม่สามารถพัฒนางานได้ทันตามกำหนด เกิดความล่าช้าใช้ระยะเวลาในการพัฒนาระบบยาวนานทำให้ไม่ทันต่อการใช้งาน หรือเมื่อพัฒนาเสร็จแล้วระบบที่ได้ไม่ตรงตามความต้องการของผู้ใช้หรือล้าสมัย จากปัญหาการพัฒนาระบบล่าช้าใช้ระยะเวลายาวนานนี้ วิธีการวิเคราะห์และออกแบบระบบแบบโครงสร้างจึงไม่เหมาะสมที่จะนำไปพัฒนาระบบสนับสนุนการตัดสินใจ (DSS) และระบบสารสนเทศสำหรับผู้บริหารระดับสูง (EIS) เนื่องจากวิธีการแบบโครงสร้างต้องกำหนดความต้องการของผู้ใช้และความต้องการนั้นต้องหยุดนิ่งไม่เปลี่ยนแปลง จึงจะไปทำเฟสต่อไปได้ ซึ่งหากนำไปใช้กับระบบ DSS และ EIS ไม่ได้ เนื่องจากผู้ใช้ระบบดังกล่าวคือผู้จัดการหรือผู้บริหารระดับสูง ซึ่งเป็นผู้ที่มีเวลาน้อย ไม่เคยชินกับการที่ต้องระบุความต้องการระบบ ความต้องการมักไม่หยุดนิ่งเปลี่ยนแปลงเร็ว ไม่คุ้นเคยกับการใช้เทคโนโลยีสารสนเทศ และไม่สามารถที่จะรอขั้นตอนต่างๆ กว่าจะได้ใช้ระบบ นักวิเคราะห์ระบบจึงต้องจัดหาระบบ DSS และ EIS ในรูปแบบที่สามารถตอบสนองความต้องการของผู้บริหารได้ทันต่อการใช้ ไม่ต้องรอนาน

ทางเลือกในการพัฒนาระบบ (Alternative Approaches to Systems Development)

จากการที่ผู้ใช้ไม่ได้ใช้ระบบตามระยะเวลาที่คาดหวังอันมีสาเหตุมาจากการพัฒนาระบบเกิดความล่าช้า ผู้ใช้ต้องการนำระบบใหม่มาตอบสนองความต้องการของผู้ใช้โดยที่ไม่ต้องรอคอยเป็นระยะเวลานาน สาเหตุดังกล่าวนำไปสู่วิธีการพัฒนาระบบ 2 รูปแบบด้วยกันคือ การทำต้นแบบ (Prototyping) และผู้ใช้ขั้นปลายพัฒนาโปรแกรมเอง (End User Computing)

7.2 การทำต้นแบบ (Prototyping)

การทำต้นแบบสร้างขึ้นมาจากการนำความต้องการของผู้ใช้มาสร้างเป็นระบบสารสนเทศและให้ผู้ใช้ได้ทดลองทำงานและรับรองว่าต้นแบบนั้นถูกต้องตามความต้องการของผู้ใช้ ขั้นตอนแรกเป็นการติดต่อระหว่างนักพัฒนาระบบกับผู้ใช้ มีการสนทนาให้ข้อมูล อธิบายปัญหาต่างๆที่เกิดขึ้น มีการทดสอบต้นแบบแต่ละเวอร์ชั่นที่พัฒนาขึ้น มีการให้ข้อคิดเห็นเพื่อนำไปปรับปรุงเปลี่ยนต้นแบบที่ทำขึ้น การทำต้นแบบเป็นสิ่งที่ต้อง "ทดลอง" ต้นแบบที่ทำขึ้นก่อนที่จะพัฒนาเป็นระบบจริง แต่ไม่ใช่การกำหนดจุดมุ่งหมายเริ่มต้น จุดมุ่งหมายพื้นฐานคือเพื่อให้เห็นระบบหลังการพัฒนาเสร็จแล้ว (final product) ว่ามีลักษณะเป็นเช่นไร นักพัฒนาระบบจะทำการพัฒนาเป็นเวอร์ชันแรกอย่างรวดเร็ว ส่วนจะกลายเป็นซอฟต์แวร์ที่สามารถใช้งานได้ต้องผ่านกระบวนการทำซ้ำ

ประสิทธิภาพของการทำต้นแบบ (Features of Prototyping)

Efraim Turban กำหนดลักษณะสำคัญของการทำต้นแบบไว้ 5 ประการ ดังนี้

1. ผู้ใช้เรียนรู้สารสนเทศที่ตนเองต้องการ อันเป็นการนำไปสู่กระบวนการออกแบบ
2. การพัฒนาต้นแบบเป็นกระบวนการทำซ้ำ ในแต่ละเวอร์ชันต้องใช้เวลาอันสั้น จึงต้องการผลย้อนกลับจากผู้ใช้อย่างรวดเร็ว เพราะระยะเวลาการให้ผลย้อนกลับนี้มีผลต่อความเร็วในการพัฒนา
3. ผู้ใช้มีส่วนร่วมอย่างใกล้ชิดในกระบวนการพัฒนา
4. การทำต้นแบบเบื้องต้นมีค่าใช้จ่ายต่ำ
5. การทำต้นแบบมีการทำซ้ำและต้องการผลย้อนกลับเพื่อที่จะได้ทราบความต้องการของผู้ใช้อย่างชัดเจนและเพื่อให้ผู้ใช้ได้เห็นว่าระบบใหม่มีลักษณะและมีความสามารถอย่างไรบ้าง

ประเภทของต้นแบบ (Type of Prototypes)

ต้นแบบที่ใช้ในการพัฒนาระบบมีหลายประเภท ได้แก่

1. mock-up prototype

มีลักษณะเหมือน final product แต่ยังไม่สามารถทำงานใดๆได้ เป็นเพียงการสาธิตให้ผู้ใช้ได้มองเห็นรูปแบบการนำข้อมูลเข้าและการแสดงผลการนำออก การใส่ข้อคำถาม และการโต้ตอบกับระบบ ต้นแบบชนิดนี้ช่วยให้ผู้ใช้สามารถกำหนดความต้องการของตนเองได้อย่างรวดเร็ว ซึ่งกระบวนนี้มีบางครั้งเรียกว่า ระบบสาธิต (demo system) อันเป็นการแสดงวิธีการใช้ให้ดูเป็นตัวอย่างเพื่อให้เห็นการทำงาน

2. interactive prototype

มีลักษณะเหมือน mock-up แต่มีส่วนที่ให้ผู้ใช้สามารถทำการโต้ตอบกับจอภาพได้แต่ยังไม่มีการประมวลผล เป็นเพียงการจำลองการโต้ตอบ เพื่อให้ผู้ใช้เห็นรูปแบบของการนำข้อมูลเข้าและการนำข้อมูลออก ช่วยให้ผู้ใช้มองเห็นและรู้สึกว่า final product จะมีลักษณะเป็นอย่างไร

3. working model prototype

ต้นแบบชนิดนี้สามารถทำงานได้บ้าง แต่ยังไม่มีประสิทธิภาพเหมือน final product ช่วยให้ผู้ใช้เข้าใจปัญหา ต้นแบบชนิดนี้รียกว่า disposable system เป็นการจัดทำต้นแบบขึ้นมาอย่างเร่งด่วนเพื่อปฏิบัติงานได้ แต่ไม่มีการจัดทำเอกสาร ไม่มีแผนการจัดทำสำเนา (backup) เป็นการทำงานให้เสร็จในเวลาอันสั้นไม่ได้มุ่งหมายใช้ในระยะยาวหรือเพื่อใช้ต้นแบบนี้ตลอดไป

4. production model prototype

ต้นแบบประเภทนี้มีจุดเริ่มพัฒนาให้เป็นเวอร์ชันที่สมบูรณ์ โดยนำเอาส่วนการทำงานที่สำคัญของระบบมาพัฒนาให้เป็นต้นแบบส่วนย่อยๆ เมื่อแต่ละส่วนย่อยพัฒนาได้ตรงตามความต้องการของผู้ใช้แล้วจะนำแต่ละส่วนย่อยมารวมกันจนเป็น final product ที่เป็นเวอร์ชันสมบูรณ์

ขั้นตอนการพัฒนาต้นแบบ (The Prototype Development Process)

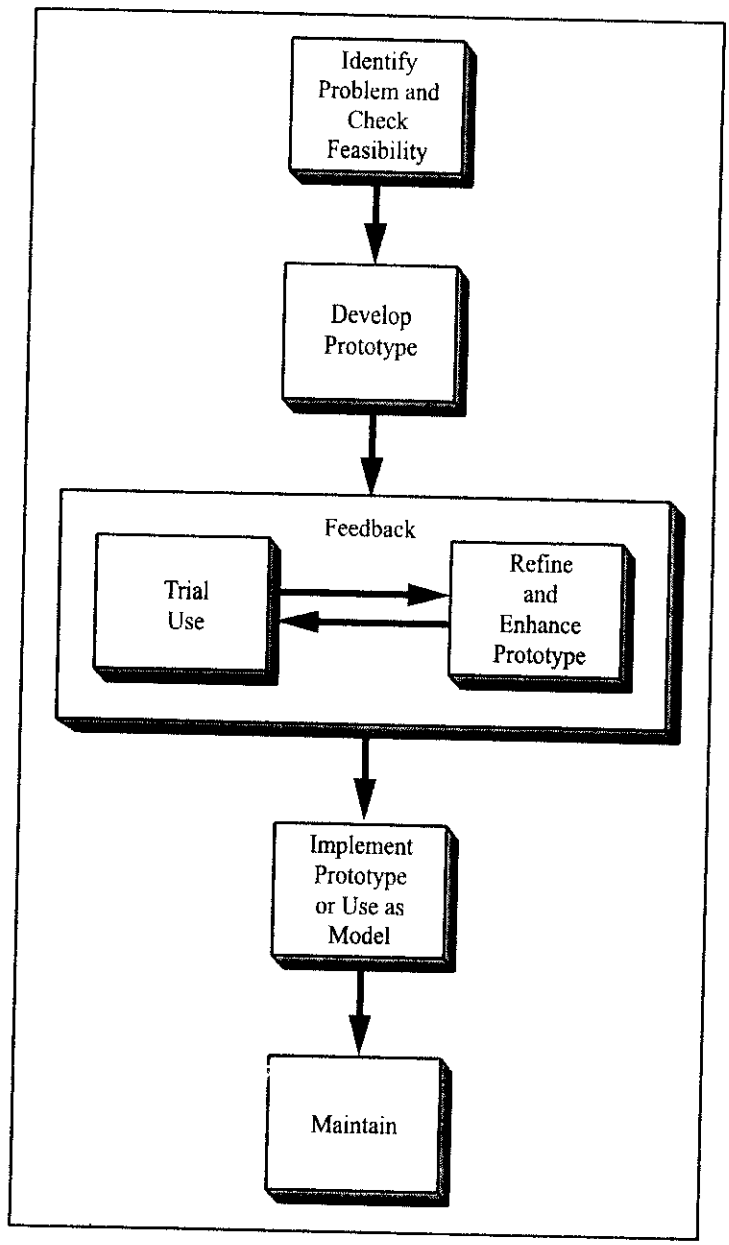
ขั้นตอนการพัฒนาต้นแบบมี 5 ขั้นตอน ดังนี้

1. นักพัฒนาระบบและผู้ใช้ร่วมสนทนา สรุปปัญหาของงานที่เกิดขึ้น กำหนดปัญหาและความต้องการที่เหมาะสม เพื่อนำไปพัฒนา
2. นำปัญหาหลักๆที่ต้องแก้ไขนั้นมาพัฒนาเป็นต้นแบบของระบบใหม่
3. ผู้ใช้เรียนรู้งานของตนเองกับต้นแบบว่าตรงตามความต้องการหรือไม่ และให้ผลสะท้อนกลับกับนักพัฒนาระบบและนักพัฒนาระบบจะนำต้นแบบไปปรับปรุงให้ดีขึ้นจนกว่าจะตรงตามความต้องการของผู้ใช้
4. การบำรุงรักษาระบบสารสนเทศ

ขั้นตอนการออกแบบต้นแบบแสดงดังรูป 7-2 ในขั้นตอนที่สามเป็นการทำซ้ำซึ่งมีความสำคัญมากทำให้นักพัฒนาทราบความต้องการของผู้ใช้ที่ชัดเจน เป็นขั้นตอนที่ได้มาจากการยอมรับของผู้ใช้ การเปรียบเทียบการทำต้นแบบกับกระบวนการออกแบบแบบโครงสร้าง แสดงดังตาราง 7-1 การศึกษาความเป็นไปได้ของกระบวนการแบบโครงสร้างเปรียบเทียบได้กับขั้นตอนแรกของการทำต้นแบบ การวิเคราะห์ การออกแบบทั่วไปและการออกแบบรายละเอียดเปรียบเทียบได้กับขั้นตอนที่ 2 และ 3 ของการทำต้นแบบ วิธีการพัฒนาระบบทั้งสองวิธีมีเฟสที่เหมือนกันคือ การทำให้เกิดผลและการดูแลรักษาระบบ สิ่งที่ต่างกันคือในกระบวนการทำต้นแบบมีการทำซ้ำ แต่ไม่ปรากฏการทำซ้ำในการวิเคราะห์ การออกแบบทั่วไปและการออกแบบรายละเอียด เพราะหากเฟสเหล่านี้เกิดการทำซ้ำแล้วจะไม่สามารถไปสู่การดำเนินการในเฟสถัดไปได้

จากที่กล่าวมาข้างต้นจะเห็นว่าการทำต้นแบบมีข้อดีเหนือวิธีแบบโครงสร้าง การทำต้นแบบทำได้รวดเร็วกว่าเพราะมีความกระชับกว่าเฟสการวิเคราะห์ การออกแบบทั่วไปและการออกแบบรายละเอียด ทำให้โครงการสำเร็จเร็วกว่า การลดเวลานำไปสู่การลดค่าใช้จ่าย การทำต้นแบบทำให้เกิดการติดต่อประสานงานกันระหว่างผู้ใช้และนักพัฒนาระบบซึ่งสถาณการณ์เช่นนี้ช่วยให้ผู้

ใช้ที่ไม่สามารถระบุความต้องการที่ชัดเจนของตนเองได้สามารถเข้าใจระบบได้ง่ายขึ้น และช่วยให้นักพัฒนาระบบที่ไม่แน่ใจความต้องการด้านฮาร์ดแวร์และซอฟต์แวร์ทำได้ง่ายขึ้นเพราะเปรียบเสมือนกับมีฉากจำลองให้สามารถพิจารณาได้ง่ายขึ้น



รูป 7-2 ขั้นตอนการออกแบบต้นแบบ

ปัญหาของการทำต้นแบบเกิดขึ้นเมื่อมีการใช้ต้นแบบกับปัญหาผิดประเภทหรือนำไปใช้อย่างไม่ชัดเจน ปัญหาเหล่านี้รวมถึงการที่นักพัฒนาระบบไม่จัดทำเอกสาร ด้านผู้ใช้เกิดความเบื่อหน่ายหรือไม่เต็มใจทดสอบต้นแบบและการที่ต้องให้ผลย้อนกลับ ปัญหาดังกล่าวจะไม่เกิดขึ้นในตอนต้นแต่จะเกิดขึ้นในเวลาต่อมา เมื่อพบว่าระบบใหม่เข้ากันไม่ได้กับระบบปัจจุบันที่มีอยู่ซึ่งไม่ได้ถูกพิจารณาในการทำต้นแบบ หรือเกิดปัญหาขึ้นเมื่อนักพัฒนาระบบหรือผู้ใช้ย้ายงานหรือสถานที่ทำงานแล้วการที่ไม่จัดทำเอกสารอธิบายระบบไว้จึงไม่มีใครสามารถอธิบายเบื้องหลังของการทำต้นแบบหรือโปรแกรมว่ามีความสำคัญอย่างไร การหลีกเลี่ยงปัญหาเหล่านี้ทำได้โดยกลับไปสู่วิธีการแบบโครงสร้าง โดยใช้เฟสการวิเคราะห์และเฟสการออกแบบ หลังจากนั้นสร้างต้นแบบขึ้นมาและทำให้เกิดผลและทดสอบความถูกต้อง

การทำต้นแบบเป็นวิธีที่เหมาะสมและเป็นประโยชน์สำหรับการพัฒนาระบบสารสนเทศบางประเภทอย่างเช่น DSS และ EIS แต่ไม่เหมาะสมกับระบบสารสนเทศประเภทอื่นๆ ได้แก่ ระบบที่มีขาดใหญ่ การประมวลผลแบบกลุ่มหรือระบบที่มีอัลกอริทึมยุ่งยาก การทำต้นแบบอาจเข้ากันไม่ได้กับระบบสารสนเทศที่มีอยู่ในปัจจุบันหรืออาจเกิดความขัดแย้ง เข้ากันไม่ได้ ไม่สามารถเชื่อมโยงกับฐานข้อมูลขององค์กร

Structured	**Prototyping**
1. Project Identification 2. Feasibility Study	1. Identify Project and check feasibility
3. Analysis 4. General Design 5. Detailed Design	2. Develop Protype 3. Trial use of prototype and refinement or enhancement or prototype
6. Implementation 7. Maintenance	4. Prototype is implemented or used as model for implementation 5. Maintenance

ตาราง 7-1 เปรียบเทียบวิธีการใช้ต้นแบบและวิธีการแบบมีโครงสร้าง

7.3 การพัฒนาโปรแกรมของผู้ใช้ขั้นปลาย (End User Computing)

การพัฒนาระบบสารสนเทศใช้ระยะเวลายาวนานกว่าจะเสร็จสมบูรณ์ ผู้ใช้จึงหาหนทางที่จะทำโครงการให้เสร็จสมบูรณ์ด้วยตนเอง ผู้ใช้บางคนมีความสามารถเรียนรู้ภาษาโปรแกรมเพื่อสร้างโปรมแกรมมาใช้งานเอง หรือค้นหาซอฟต์แวร์สำเร็จรูปมาใช้ในงานประยุกต์ต่างๆของผู้ใช้เอง จึงเรียกผู้ใช้กลุ่มนี้ว่า "power user"

ผู้ใช้ขั้นปลาย (end user) เป็นผู้ใช้ขั้นปลายไม่ใช่ผู้เชี่ยวชาญด้านการออกแบบหรือพัฒนาระบบสารสนเทศ เป็นผู้ใช้คอมพิวเตอร์ที่นำโปรมแกรมคอมพิวเตอร์ที่คนอื่นสร้างหรือเขียนไว้แล้วมาใช้งานเพื่อเพิ่มประสิทธิภาพการทำงานของตนเอง ผู้ใช้ขั้นปลายมีการเรียนรู้ซอฟต์แวร์หรือเทคนิคการทำงานใหม่ๆด้วยตนเองและถ่ายทอดความรู้เหล่านั้นให้กับพนักงานใหม่ที่เข้ามาทำงาน จนกระทั่งช่วงปลายของปี ค.ศ. 1970 ผู้ใช้ขั้นปลายที่มีความสามารถด้านคอมพิวเตอร์หรือ End User Computing หรือ EUC กลายเป็นผู้ที่มีส่วนสำคัญในการใช้คอมพิวเตอร์ในองค์การ มีการประมาณการณ์ว่า EUC เป็นผู้ใช้ทรัพยากรคอมพิวเตอร์ให้เป็นประโยชน์ประมาณ 40-50 เปอร์เซ็นต์ EUC เพิ่มจำนวนขึ้นเป็น 50-90 เปอร์เซ็นต์ต่อปี เหตุผลที่ทำให้ EUC มีความสำคัญเพิ่มมากขึ้นคือ ผู้ใช้ขั้นปลายสามารถลดระยะเวลาการพัฒนาระบบอันยาวนานลงได้ ทำให้ได้รับระบบสารสนเทศที่สามารถตอบสนองความต้องการได้ในระยะเวลาอันสั้น

ถึงแม้ผู้ใช้ขั้นปลายจะสามารถพัฒนางานประยุกต์ขึ้นมาใช้งานเองได้ แต่ว่าปัญหาในงานนั้นยังคงมีอยู่ EUC จะทราบว่างานตนเองเกิดปัญหาใดบ้างจึงทำให้ข้ามขั้นตอนการศึกษาความเป็นไปได้ แต่มุ่งตรงไปที่การเขียนโปรแกรม ในกระบวนการพัฒนาระบบหากไม่รู้ว่าปัญหาที่แท้จริงคืออะไรแล้วการแก้ปัญหาไม่ใช่สิ่งที่ง่ายเลย EUC รวมเอาการวิเคราะห์และออกแบบเป็นขั้นตอนการแก้ปัญหาขั้นตอนเดียวกัน ผู้ใช้จะเลือกใช้ซอฟต์แวร์สำเร็จรูปหรือเขียนโปรแกรมเอง และมักทำการออกแบบโปรแกรมและเขียนโปรแกรมไปพร้อมๆกัน หลังจากพัฒนางานประยุกต์เสร็จแล้วผู้ใช้ก็คิดว่างานสำเร็จแล้วไม่คำนึงถึงขั้นตอนการบำรุงรักษาระบบ และยิ่งไปกว่านั้นงานประยุกต์ที่คาดว่าเสร็จสิ้นแล้วแต่กลับมีปัญหาที่ต้องแก้ไขในภายหลังอยู่เสมอ ซึ่งการกระทำเช่นนี้ทำให้เกิดปัญหากับองค์การ

ประเภทของผู้ใช้ขั้นปลาย (Types of End Users)

ตามคำนิยามแล้วบุคคลที่ใช้คอมพิวเตอร์โดยทางตรงหรือทางอ้อมเรียกว่าผู้ใช้ขั้นปลาย เช่น บุคคลที่ใช้คอมพิวเตอร์ในการจองตั๋วเครื่องบินแบบออนไลน์ด้วยตนเองเป็นผู้ใช้ทางอ้อม ในส่วนนี้จะกล่าวเฉพาะผู้ใช้ทางตรงเท่านั้น ผู้ใช้ขั้นปลายแบ่งออกเป็น 5 กลุ่ม ได้แก่

1. Non-programming end users

เป็นผู้ใช้ซอฟต์แวร์ประยุกต์เพียงอย่างเดียวและไม่มีการศึกษาความรู้เพิ่มเติมเพื่อมาแก้ไขงาน มีความรู้เฉพาะการเลือกใช้เมนูในซอฟต์แวร์ ตัวอย่างเช่น ผู้ใช้ใช้ซอฟต์แวร์ประเภทตารางทำการ Ms-Excel จัดทำรายการงบประมาณแต่ไม่รู้วิธีการใช้กราฟิกหรือฐานข้อมูลเพื่อการวิเคราะห์ข้อมูลในอนาคต

2. Command level end users

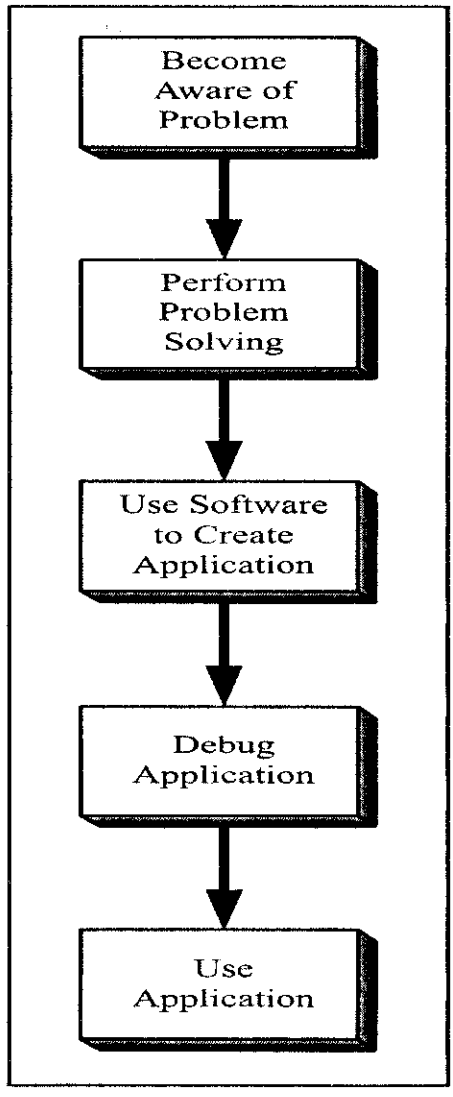
ผู้ใช้ประเภทนี้มีการเรียนรู้ซอฟต์แวร์เพื่อนำมาใช้แก้ปัญหางานของตนเอง ตัวอย่างเช่น ผู้ใช้รู้จักวิธีการใช้กราฟิกหรือฐานข้อมูลใน MS-Excel รู้จักประโยชน์การใช้งานของซอฟต์แวร์มากกว่าผู้ใช้แบบแรก

3. End user programmers

เป็นผู้ใช้ที่สามารถเขียนกระบวนงานในซอฟต์แวร์เพื่อแก้ปัญหางานที่ซับซ้อนได้ เขียนโปรแกรมได้แต่ยังไม่จัดเป็นซอฟต์แวร์ประยุกต์เชิงการค้า เช่น สามารถเขียนโปรแกรมโดยใช้มาโคร (macro) ใน Ms-Excel พัฒนาเป็นแผ่นแบบ (templates) และผู้ใช้ก็ป้อนข้อมูลที่แผ่นแบบที่ได้สร้างขึ้น

4. Functional support end users

เป็นผู้ใช้ที่มีความเชี่ยวชาญในงานของตนเอง มีความรู้ในการใช้โปรแกรมประยุกต์หรือภาษาโปรแกรมคอมพิวเตอร์ และพัฒนาซอฟต์แวร์ประยุกต์ให้กับแผนกตนเองหรือเพื่อนร่วมงาน งานที่พัฒนาขึ้นมาเป็นงานที่ผู้ใช้นั้นเกี่ยวข้องเท่านั้น เพราะสามารถเข้าใจลักษณะงานและปัญหาที่เกิดขึ้นเป็นอย่างดี ส่วนงานของแผนกอื่นหรือลักษณะอื่นนอกเหนือความรับผิดชอบ ผู้ใช้ไม่สามารถพัฒนาซอฟต์แวร์เพื่อแก้ปัญหาได้ ซึ่งต่างจากการทำงานของนักวิเคราะห์ระบบ



รูป 7-3 การพัฒนางานประยุกต์ของผู้ใช้ขั้นปลาย

ผู้ใช้ขั้นปลายกับการวิเคราะห์และออกแบบระบบ

(End Users and Systems Analysis and Design)

ในตอนต้นของบทนี้ได้กล่าวถึงวิธีการวิเคราะห์และออกแบบระบบที่อาจทำให้โครงการเกิดความล่าช้าได้เนื่องมาจากมีการทำงานตามลำดับของเฟสคือต้องทำเฟสที่อยู่ก่อนให้เสร็จจึงทำเฟสถัดไปได้ ซึ่งเป็นสาเหตุหนึ่งที่ผู้ใช้เปลี่ยนแปลงบทบาทตนเองจากผู้ใช้เพียงอย่างเดียวมาพัฒนาระบบให้กับแผนกตนเอง ผู้ใช้ไม่ต้องรอคอยอย่างยาวนาน

ผู้ใช้ขั้นปลาย หรือ EUC มีการพัฒนาระบบที่แตกต่างจาก SA เนื่องจาก EUC เป็นเจ้าของระบบงานจึงรู้ขั้นตอนวิธีในการดำเนินงานของระบบงาน เมื่อทำการแก้ปัญหาด้วยคอมพิวเตอร์จึงทำขั้นตอนต่างๆเองตั้งแต่ต้นจนจบ และไม่ได้ทำตามขั้นตอนในวัฏจักรการพัฒนาระบบ อาจข้ามขั้นตอนการวิเคราะห์และออกแบบระบบไป การทำงานเป็นแบบลองผิดลองถูก มุ่งหวังที่ผลลัพธ์เพียงเดียว หากไม่ได้ผลลัพธ์ตามที่ต้องการอาจสรุปเอาว่าไม่มีวิธีที่สามารถทำได้ นั่นคือเป็นการสรุปการแก้ไขปัญหาที่ผิดประเด็น เพราะข้ามขั้นตอนต่างๆในวัฏจักรการพัฒนาระบบ รูป 7-3 แสดงให้เห็นถึงวิธีการที่ EUC ใช้ในการพัฒนางานประยุกต์ คือเริ่มจากการรับรู้ถึงปัญหาที่เกิดขึ้น การแก้ปัญหา การใช้ซอฟต์แวร์สร้างงานประยุกต์ การแก้จุดบกพร่องและการใช้งานประยุกต์

ความเสี่ยงและปัญหาที่เกิดจากการให้ผู้ใช้ขั้นปลายพัฒนาระบบ

(Risks and Problems with End User Development)

ความเสียงที่เกิดขึ้นกับจากการที่ผู้ใช้พัฒนาระบบเป็นสิ่งที่ต้องนำมาพิจารณา เนื่องจากผู้ใช้ข้ามขั้นตอนการวิเคราะห์และออกแบบระบบ การกระทำเช่นนี้ถือเป็นความเสี่ยงสูงมาก ปัญหาที่อาจเกิดขึ้นเช่น การแก้ปัญหาผิดประเด็น แก้ปัญหาผิดวิธี เลือกใช้เครื่องมือไม่เหมาะสม ละเลยการจัดทำเอกสารสำคัญในการพัฒนาระบบ ทดสอบงานประยุกต์ไม่ละเอียดหรือทดสอบไม่สมบูรณ์ทำให้ผู้ใช้หาข้อสรุปผิดๆ เป็นต้น

เนื่องจากผู้ใช้มักแก้ปัญหาได้เฉพาะงานที่ตนเองรับผิดชอบอยู่เนื่องจากเป็นเจ้าของงาน ดังนั้นจึงไม่มีการจัดทำเอกสารที่เกี่ยวข้องกับงานประยุกต์นั้น หากไม่มีผู้ใช้คนอื่นใช้งานประยุกต์นี้ก็เสมือนกับว่าไม่เคยมีการใช้ซอฟต์แวร์ที่ผู้ใช้สร้างขึ้นแก้ปัญหางานมาก่อนเลยเพราะไม่มีการบันทึกเป็นเอกสารระบบ หรือในทางตรงกันข้าม หากมีการนำเอาซอฟต์แวร์นี้มาใช้แล้วผู้ใช้คนนี้ลาออกจากงานก็ทำให้เกิดปัญหาอีก เนื่องจากไม่มีเอกสารระบบให้ผู้ใช้อื่นศึกษางาน อีกปัญหาหนึ่งคือหากมีการใช้งานประยุกต์นี้ไปแล้วและมีข้อผิดพราดทางตรรกะ (logic errors) เกิดขึ้นซึ่งทำให้ได้ข้อมูลหรือสารสนเทศผิดพราด

ความเสี่ยงต่างๆที่เกิดขึ้นจากการกระทำของผู้ใช้สามารถหลีกเลี่ยงได้ โดยผู้ใช้มีการสนทาสื่อสาร แลกเปลี่ยนความคิดเพื่อให้ทราบปัญหาที่เกิดขึ้นกับเพื่อนร่วมงาน มีการเลือกใช้เครื่องมือพัฒนางานที่เหมาะสม มีการพัฒนาเอกสารของงานประยุกต์ที่พัฒนาขึ้น มีหลายองค์การที่ต้องการให้ผู้ใช้พัฒนางานให้กับองค์การอย่างมีมาตรฐาน เพื่อให้เพื่อนร่วมงานของผู้ใช้ได้เข้าใจว่ามีอะไรเกิดขึ้นกับงานประยุกต์มาก

7.4 เครื่องมือซอฟต์แวร์ของผู้ใช้ขั้นปลาย (End User Software Tools)

เครื่องมือที่ผู้ใช้ใช้เป็นเครื่องมือที่อยู่บนเครื่องพีซี ได้แก่ ภาษาในยุคที่สี่ ภาษาสอบถามข้อมูล มาโครในตารางแผ่นทำการ desktop organizer และ keyboard macro instruction เป็นต้น

- Personal Computer Based Languages เช่นภาษาในยุคที่สามที่เคยใช้สำหรับเครื่องเมนเฟรมแล้วมีเวอร์ชันสำหรับเครื่องพีซี ภาษาในยุคที่สี่ และการเขียน batch command เป็นต้น
- Database Query Language เป็นภาษาสอบถามข้อมูลที่มีประสิทธิภาพใช้ในการสั่งเรียกข้อมูลจากฐานข้อมูลมาดูบนจอภาพหรือสั่งให้พิมพ์ออกมา ภาษาสอบถามเพียงคำสั่งเดียวแต่เทียบเท่ากับการเขียนชุดคำสั่งในภาษายุคที่สามหลายๆชุดคำสั่ง
- Spreadsheet Macros เป็นซอฟต์แวร์ชนิดตารางแผ่นทำการที่ผู้ใช้นิยมกันมาก เป็นชุดของรายการคำสั่ง แต่ละคำสั่งจะถูกกระทำการ (execute) เป็นลำดับ คำสั่งมาโครเขียนเป็นวงวน (loop)ได้หรือสั่งให้คำสั่งถัดไปกระทำการได้ เป็นคำสั่งที่ผู้ใช้เขียนขึ้นเองเพื่อสั่งให้ทำงานพิเศษตามที่ผู้ใช้ต้องการนอกเหนือไปจากการทำงานปกติของโปรแกรมนั้นๆ รูป 7-4 แสดงการใช้ Spreadsheet Macro ที่เขียนการเคลื่อนย้ายตัวชี้ตำแหน่ง (cursor) ที่ผู้ใช้ป้อนข้อมูล
- Desktop Organizers and Keyboard Macros เป็นเครื่องมือของผู้ใช้ขั้นปลายสองประเภท Desktop Organizers เป็นโปรแกรมอรรถประโยชน์ที่มีการซ่อนไว้ในหน่วยความจำและสามารถเข้าถึงได้ในขณะที่ผู้ใช้ทำงานอยู่กับโปรแกรมตัวอื่น เมื่อมีการเข้าถึง Desktop Organizers ผู้ใช้สามารถเลือกกิจกรรมต่างๆ ได้เช่นการเรียกดูปฏิทิน การใช้สมุดพก (notepad) หรือการติดต่อโดยการหมุนโทรศัพท์ถึงตัวแทนจำหน่าย เป็นต้น ส่วน Keyboard Macro เป็นการบันทึกชุดของคำสั่งเหมือนกับเป็นที่พักคำสั่งชั่วคราวไว้แล้วสามารถเรียกใช้คำสั่งที่ต้องการได้อีกในภายหลัง
- Nonprocedural Software Package ซอฟต์แวร์สำเร็จรูปประเภทนี้ผู้ใช้ไม่จำเป็นต้องมีความรู้ด้านการเขียนโปรแกรม ผู้ใช้สามารถทำงานได้โดยใช้ข้อความ กราฟิก การโต้ตอบแบบง่ายและการป้อนข้อมูลเข้า ตัวอย่างเช่น ซอฟต์แวร์สำเร็จรูปงานบัญชี และซอฟต์แวร์สำเร็จรูปทางกราฟิก เป็นต้น

```
{GOTO}A5~
{GETLABEL "Input employee's name",@CELLPOINTER("ADDRESS")}{RIGHT}
{GETLABEL "Input employee's SS Number",@CELLPOINTER("ADDRESS")}{RIGHT}
{GETLABEL "Input employee's pay rate",@CELLPOINTER("ADDRESS")}{RIGHT}
{DOWN}{END}{LEFT}
{GETLABEL "Do you wish to continue-Y or N?",YESNO}
{IF YESNO="Y"}{BRANCH LOOP}
{QUIT}
```

รูป7-4 Spreadsheet Macro

คำถามท้ายบท

1. จงบอกชื่ออักษรย่อของกระบวนการพัฒนาระบบแบบโครงสร้าง

2. จงอธิบายสาเหตุที่ทำให้กระบวนการพัฒนาระบบแบบโครงสร้างเกิดความล่าช้า

3. จงอธิบายความแตกต่างระหว่างการทำตัวแบบ (prototype) และการพัฒนางานของผู้ใช้ขั้นปลาย (EUC)

4. จงอธิบายคุณลักษณะของตัวแบบ และตัวแบบประเภทใดที่มักเรียกเป็นระบบสาธิต (demo system)

5. จงอธิบายเฟสใดในกระบวนการพัฒนาระบบมีการนำตัวแบบมาใช้และใช้อย่างไร

6. จงอธิบายปัญหาที่เกิดขึ้นในการทำตัวแบบ

7.จงอธิบายเครื่องการทำตัวแบบมีอะไรบ้าง และการใช้ภาษาในยุคที่สี่และเคส (CASE) มีความแตกต่างกันอย่างไร

8. จงอธิบายผู้ใช้ขั้นปลาย (end user) คืออะไร และ EUC คืออะไร

9.จงอธิบายผู้ใช้ขั้นปลายแบ่งออกเป็นกี่ประเภทอะไรบ้างและประเภทใดบ้างที่สามารถเขียนโปรแกรมได้

10. จงเปรียบเทียบการพัฒนางานของผู้ใช้ขั้นปลายและกระบวนการพัฒนาแบบโครงสร้าง

11. จงอธิบายความเสี่ยงและปัญหาที่เกิดขึ้นกับ EUC

12. จงอธิบายความสัมพันธ์ระหว่าง IC และ EUC