

Chapter 14

System Implementation



อ.สันติภูธร นรบิน

เรียบเรียงโดย

อ.วไลลักษณ์ วงษ์รัตน์

Content

- 1) ขั้นตอนการพัฒนาและติดตั้งระบบ
- 2) การทดสอบโปรแกรม (Software Testing)
- 3) การติดตั้งระบบ (Installation)
- 4) การจัดทำเอกสาร (Documentation)
- 5) การฝึกอบรมพนักงาน (Training)
- 6) การบริการให้ความช่วยเหลือหลังการติดตั้งระบบ (Support)

1. ขั้นตอนการพัฒนาและติดตั้งระบบ



ขั้นตอนการพัฒนาและติดตั้งระบบ

- ❑ มีวัตถุประสงค์เพื่อการปรับเปลี่ยนจากระบบงานเดิมเข้าสู่ระบบงานใหม่ที่ได้ผ่านการวิเคราะห์และออกแบบมาแล้ว
- ❑ เริ่มจาก
 - เขียนโปรแกรมของระบบงาน
 - ทดสอบโปรแกรมที่เขียนขึ้น เพื่อให้เป็นโปรแกรมที่น่าเชื่อถือสามารถทำงานได้อย่างมีข้อผิดพลาดน้อยที่สุด
 - จัดทำเอกสารคู่มือการใช้ระบบ เพื่อเตรียมจัดฝึกอบรมให้กับพนักงาน

ขั้นตอนการพัฒนาและติดตั้งระบบ (ต่อ)

- ❑ ขั้นตอนดังกล่าวข้างต้นนั้นโดยส่วนมากแล้ว จะไม่อยู่ในความรับผิดชอบของ SA ยกเว้นกรณีที่องค์กรนั้นแต่งตั้งให้ SA รับผิดชอบการเขียนโปรแกรมและติดตั้งระบบรวมอยู่ด้วย
- ❑ อย่างไรก็ตาม เนื่องจาก SA เป็นผู้ประสานงานระหว่างผู้เกี่ยวข้องกับการพัฒนาระบบทั้งหมด ดังนั้นหน้าที่ในส่วนนี้จัดว่ายังเป็นส่วนหนึ่งของความรับผิดชอบ เพื่ออำนวยความสะดวกให้กับทีมงานเขียนโปรแกรมและติดตั้งระบบ
- ❑ นอกจากนี้ยังคอยดูแลการทำงานให้เป็นไปตามสิ่งที่ได้ออกแบบและวิเคราะห์มาเพื่อให้ตรงกับความต้องการของผู้ใช้มากที่สุด

ขั้นตอนการพัฒนาและติดตั้งระบบ (ต่อ)

- ❑ ก่อนการติดตั้งระบบ นักวิเคราะห์ระบบควรมีการวางแผนงานในการติดตั้ง
- ❑ แผนงานจะต้องครอบคลุมกิจกรรมตั้งแต่
 - การทดสอบโปรแกรม
 - การทดสอบระบบ
 - การติดตั้งระบบ
 - วิธีการติดตั้ง
 - การจัดทำเอกสารคู่มือ
 - การจัดฝึกอบรม
 - และการสนับสนุนหลังการติดตั้ง

ขั้นตอนการพัฒนาและติดตั้งระบบ (ต่อ)

1.1 Rapid Application Development : RAD

1.2 Tool-based Development

1.3 Packaged Software

1.1 Rapid Application Development : RAD

- ❑ เป็นวิธีการพัฒนาระบบ (Methodology) วิธีการหนึ่ง ที่รวบรวมเทคนิค (Techniques) เครื่องมือ (Tools) และ เทคโนโลยี เพื่อผสมผสานและประยุกต์ใช้ในการสนับสนุนการพัฒนาระบบ ให้สามารถถูกล่วงโดยใช้เวลาน้อยที่สุด
- ❑ วัตถุประสงค์เพื่อให้การพัฒนาระบบในวงจรการพัฒนา สามารถดำเนินการเสร็จสิ้นได้อย่างรวดเร็ว
- ❑ ใช้เวลาในการดำเนินงานน้อยกว่าวิธีการแบบ Waterfall

วิธีการต่าง ๆ ในกลุ่มของการพัฒนาระบบด้วยวิธี RAD

1.1.1 การพัฒนาด้วยการสร้างตัวต้นแบบ

(Prototype Approach to Development)

1.1.2 การพัฒนาด้วยวิธีการวนรอบ

(Spiral Approach to Development)

1.1.1 การพัฒนาด้วยการสร้างตัวต้นแบบ

- ❑ เป็นกระบวนการเพื่อสร้างการทำงานบางส่วนหรือทั้งหมดของระบบ ที่เหมือนระบบจริงมากที่สุดเท่าที่จะเป็นไปได้
- ❑ แต่ต้นแบบนั้นจะยังไม่สมบูรณ์จนกว่าจะได้รับการเพิ่มเติมการทำงานที่ละส่วนจนครบทุกส่วนของระบบ
- ❑ ขั้นตอนการพัฒนาระบบด้วยตัวต้นแบบ แบ่งออกเป็น 3 ขั้นตอน
 - ขั้นตอนที่ 1 การวางแผน (Planning)
 - ขั้นตอนที่ 2 วิเคราะห์ (Analysis)
 - ขั้นตอนที่ 3 ออกแบบและพัฒนา (Design and Implementation)

1.1.2 การพัฒนาด้วยวิธีการวนรอบ

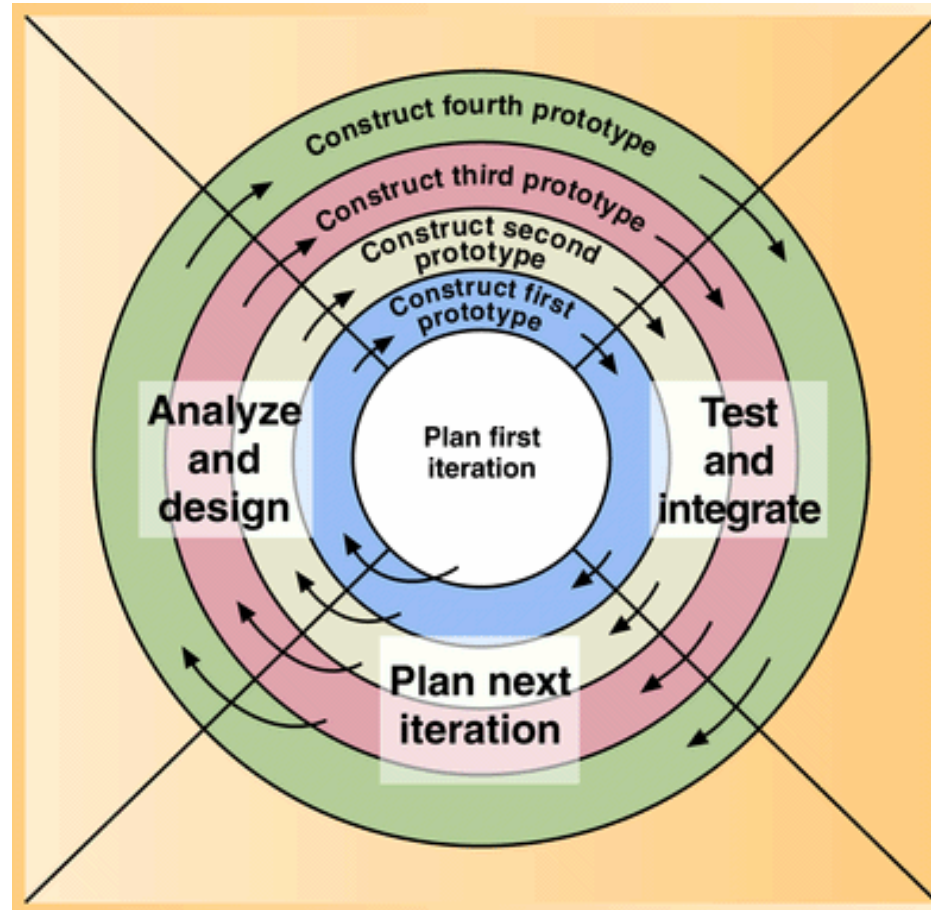
- ❑ เป็นวิธีการพัฒนาระบบแบบทำซ้ำ (Iterative Development)
- ❑ ในการทำซ้ำแต่ละรอบนั้นอาจจะทำซ้ำในขั้นตอนการวางแผน (Planning) วิเคราะห์ (Analysis) ออกแบบ (Design) และการพัฒนา (Development) รวมอยู่ด้วย
- ❑ จะเริ่มต้นที่การวางแผนการทำซ้ำก่อน โดยการทำซ้ำแต่ละรอบ จะทำการสร้างตัวต้นแบบเพิ่มเติมจนครบทั้งระบบ

1.1.2 การพัฒนาด้วยวิธีการวนรอบ (ต่อ)

□ ในการสร้างตัวต้นแบบแต่ละรอบจะแบ่งขั้นตอนในการสร้าง

- 1) วางแผนเพื่อทำซ้ำรอบต่อไป (Plan next Iteration)
- 2) วิเคราะห์และออกแบบ (Analysis and Design)
- 3) ทดสอบและรวมตัวต้นแบบ

Spiral model



1.2 Tool-based Development

- ❑ ใช้วิธีการเลือกเครื่องมือสนับสนุนการพัฒนาระบบที่ตรงกับความต้องการที่สุด และจะไม่พัฒนาระบบที่มีความต้องการที่ยากแก่การพัฒนาด้วยเครื่องมือ
- ❑ องค์กรต่าง ๆ ไม่นิยมเลือกใช้วิธีการพัฒนาระบบวิธีนี้ เนื่องจากองค์กรเองไม่สามารถระบุความต้องการที่ยากต่อการพัฒนาระบบได้ ซึ่งในความเป็นจริงแล้ว ระบบงานที่จะพัฒนาขึ้นมา นั้นเพื่อตอบสนองความต้องการในระหว่างการทำงาน ดังนั้นวิธีการนี้ถึงแม้จะพัฒนาระบบได้ง่ายแต่ไม่สามารถตอบสนองความต้องการสำหรับระบบใหม่ได้อย่างเต็มที่

1.3 Packaged Software

- ❑ กรณีที่องค์กรเลือกที่จะซื้อซอฟต์แวร์สำเร็จรูปมาใช้ในระบบงานใหม่ ก่อนที่จะมีการติดตั้งระบบ โปรแกรมเมอร์จะต้องพิจารณาว่าซอฟต์แวร์ที่ซื้อมานั้น จะต้องได้รับการแก้ไขอย่างไรจึงจะมีลักษณะการทำงานที่ตรงกับความต้องการขององค์กร

การแก้ไขซอฟต์แวร์ที่ซื้อมาจำแนกได้ 3 ลักษณะ

- 1.3.1 การปรับฟังก์ชันการทำงานของซอฟต์แวร์สำเร็จรูป
(Customize Packages)
- 1.3.2 การรวมซอฟต์แวร์สำเร็จรูปให้ทำงานร่วมกัน
(Integration Packages)
- 1.3.3 การยกระดับความสามารถของซอฟต์แวร์สำเร็จรูป
(Upgrading Packages)

1.3.1 Customize Packages

- ❑ เป็นการแก้ไขซอฟต์แวร์ที่ซื้อจากภายนอกองค์กร ด้วยการเพิ่มหรือลดฟังก์ชันการทำงานของซอฟต์แวร์ที่ซื้อมา
- ❑ การเพิ่มหรือลดฟังก์ชันการทำงานมี 3 ชนิดดังนี้
 - 1) Configuration
 - 2) Modification
 - 3) Enhancement

(1) Configuration

- ❑ เป็นการเลือกตัวเลือกลักษณะการทำงานของซอฟต์แวร์ที่มีการเตรียมตัวเลือกนั้นไว้ให้แล้ว
- ❑ หรือเป็นการเปลี่ยนแปลงค่าพารามิเตอร์ต่าง ๆ ของซอฟต์แวร์ เพื่อให้รูปแบบการทำงานเหมาะกับระบบงานขององค์กร เช่น การเลือกรูปแบบของ Interfaces จากตัวเลือกที่ซอฟต์แวร์มีให้ เป็นต้น

(2) Modification

- ❑ เป็นการเปลี่ยนแปลงโค้ดของซอฟต์แวร์ เพื่อให้มีรูปแบบการทำงานที่เหมาะสมกับระบบงานขององค์กร
- ❑ ต้องอาศัยโปรแกรมเมอร์ที่มีความชำนาญเป็นพิเศษเนื่องจากซอฟต์แวร์สำเร็จรูปบางชนิดไม่อนุญาตให้มีการแก้ไขโค้ด หรือหากอนุญาตให้แก้ไข แต่อาจจะมีเงื่อนไขคือจะไม่รับผิดชอบต่อความเสียหายที่อาจจะเกิดขึ้น
- ❑ ข้อเสียของ Modification คือ เมื่อแก้ไขโค้ดไปแล้วจะทำให้เกิดปัญหาในการ Upgrade ซอฟต์แวร์จากทางผู้ขาย

(3) Enhancement

- ❑ เป็นการเพิ่มโค้ดหรือเพิ่มโมดูลการทำงานของซอฟต์แวร์
- ❑ เพื่อเพิ่มฟังก์ชันการทำงานให้กับซอฟต์แวร์สำเร็จรูปที่ซื้อมา
- ❑ การ Customize ซอฟต์แวร์สำเร็จรูปในลักษณะนี้ไม่ใช่การแก้ไขโค้ด แต่เป็นการเพิ่มฟังก์ชันการทำงานเข้าไปในซอฟต์แวร์

1.3.2 Integration Packages

- ❑ เป็นการรวมซอฟต์แวร์สำเร็จรูปในงานเฉพาะด้านของแต่ละหน่วยงาน ให้สามารถทำงานร่วมกันได้ด้วยการใช้ข้อมูลร่วมกัน
- ❑ แนวคิดคือการสร้างและใช้ข้อมูลร่วมกันได้จากทุกแผนกไม่ว่าจะเป็นแผนกบุคคล แผนกบัญชี การเงิน ผลิต จัดซื้อ เป็นต้น
- ❑ ข้อดีของการแก้ไขซอฟต์แวร์สำเร็จรูปประเภทนี้คือ แต่ละแผนกจะได้ข้อมูลที่มีรูปแบบเป็นมาตรฐานเดียวกันในการใช้งานนั่นเอง

1.3.3 Upgrading Packages

□ มีหลายลักษณะ เช่น

- ยกระดับความสามารถของซอฟต์แวร์ให้ทำงานง่ายขึ้น
- การแก้ไขข้อผิดพลาดที่พบจากซอฟต์แวร์เวอร์ชัน (Version) เดิม
- การเพิ่มฟังก์ชันการทำงานของซอฟต์แวร์ เป็นต้น

2. การทดสอบโปรแกรม (Software Testing)



การทดสอบโปรแกรม

- ❑ เป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งหลังจากที่โปรแกรมเมอร์ได้เขียนโปรแกรมเสร็จสิ้นแล้ว จะต้องทดสอบว่าโปรแกรมนั้นให้ผลลัพธ์ที่ถูกต้องหรือไม่
- ❑ เมื่อพบข้อผิดพลาดจะได้ทำการแก้ไขและป้องกันข้อผิดพลาดที่จะเกิดขึ้นต่อไป ทั้งนี้เพื่อเป็นการทดสอบความสมบูรณ์ของโปรแกรม รวมทั้งความน่าเชื่อถือและความถูกต้องของผลลัพธ์จากโปรแกรมที่พัฒนาขึ้น

2.1 เทคนิคในการทดสอบโปรแกรม

2.1.1 การทดสอบโดยไม่ใช้เครื่องคอมพิวเตอร์ (Manual Testing)

- การทดสอบแบบตรวจการณ (Inspection)
- การทดสอบตามลำดับคำสั่งในโปรแกรม (Desk Checking)

2.2.2 การทดสอบด้วยเครื่องคอมพิวเตอร์ (Automated Testing)

- การทดสอบด้วยการตรวจสอบไวยากรณ์ (Syntax Checking)
- การทดสอบทีละโมดูล (Unit Testing)
- การทดสอบแบบเพิ่มโมดูล (Integration Testing)
- การทดสอบด้วยโมดูลตัวแทน (Stub Testing)
- การทดสอบรวม (System Testing)

การทดสอบแบบตรวจการณ์ (Inspection)

- ❑ ทดสอบโดยโปรแกรมเมอร์เอง ด้วยการเปรียบเทียบโค้ดของโปรแกรมที่เขียนขึ้นกับรายการ Error ที่โปรแกรมเมอร์ทราบแล้วว่า จะต้องเกิดขึ้นจากโปรแกรมภาษาที่ใช้ในการพัฒนาโปรแกรม โดยการตรวจสอบว่าโค้ดที่เขียนนั้นมี Error เกิดขึ้นตามรายการหรือไม่
- ❑ เทคนิคชนิดนี้ใช้ในการป้องกันการเกิดข้อผิดพลาดในรูปแบบเดิม ไม่ให้เกิดขึ้นซ้ำอีกครั้ง สามารถทดสอบโปรแกรมแบบนี้ในขณะที่เขียนโปรแกรมได้
- ❑ การทดสอบชนิดนี้ไม่ได้ทำให้โปรแกรมเมอร์ทราบว่าได้ว่ามีผลลัพธ์ที่ถูกต้องหรือไม่ เนื่องจากไม่ได้ทดสอบการทำงานของโปรแกรม เป็นเพียงการทดสอบความผิดพลาดของโค้ดเท่านั้น

การทดสอบตามลำดับคำสั่งในโปรแกรม (Desk Checking)

- ❑ เทคนิคการทดสอบโปรแกรมชนิดนี้ กระทำโดยผู้ที่ได้รับการแต่งตั้งให้เป็นผู้ทดสอบโปรแกรมซึ่งอาจจะเป็นโปรแกรมเมอร์หรือไม่ก็ได้
- ❑ ผู้ทดสอบโปรแกรมต้องมีความเข้าใจในการทำงานทางตรรกะของโปรแกรม ด้วยการตรวจสอบโค้ดของโปรแกรมตามลำดับคำสั่งในโปรแกรม ว่ามีตรรกะที่ผิดพลาดหรือไม่
- ❑ วิธีการนี้จะทำให้เสียเวลาในการทดสอบโปรแกรมค่อนข้างมาก ถ้าระบบงานที่มีความซับซ้อนสูง

การทดสอบด้วยการตรวจสอบไวยากรณ์ (Syntax Checking)

- ❑ เป็นการทดสอบโปรแกรม ด้วยการตรวจสอบไวยากรณ์ (Syntax) ที่เขียนขึ้น โดยปกติแล้วจะได้รับการตรวจสอบด้วย Compiler ซึ่งจะใช้เวลาไม่นานสามารถทราบผลได้
- ❑ วิธีการนี้ไม่ทำให้ทราบว่าผลลัพธ์จากการทำงานของโปรแกรมนั้นจะถูกต้องหรือไม่ เนื่องจากเป็นทดสอบเพียงไวยากรณ์ของโปรแกรมเท่านั้น

การทดสอบทีละโมดูล (Unit Testing)

- ❑ บางครั้งเรียกอีกอย่างว่า Module Testing
- ❑ เป็นการทดสอบโปรแกรมทีละโมดูล เพื่อหาข้อผิดพลาดที่จะเกิดขึ้นภายในการทำงานของแต่ละโมดูล

การทดสอบแบบเพิ่มโมดูล (Integration Testing)

- ❑ เป็นการทดสอบโปรแกรมโดยการเพิ่มจำนวนโมดูลเพื่อการทดสอบ
- ❑ วิธีการนี้จะอาศัย Structure Chart ที่มีอยู่แล้ว ช่วยในการทดสอบโปรแกรม
- ❑ ซึ่งวิธีการในการทดสอบแบบ Integration นี้แบ่งออกเป็น 2 ลักษณะได้แก่
 - Top-down Approach
 - Bottom-up Approach

การทดสอบด้วยโมดูลตัวแทน (Stub Testing)

- ❑ โดยทั่วไปแล้ว โมดูลที่อยู่ในระดับบนจะเรียกใช้ข้อมูลจากโมดูลระดับล่าง แต่การทดสอบโปรแกรมแบบเพิ่มโมดูล (Integration) นั้นจะทดสอบโมดูลทีละระดับ
- ❑ ฉะนั้น Stub Testing คือ กลุ่มคำสั่งสั้น ๆ ที่เขียนขึ้นมาเพื่อเป็นโมดูลตัวแทนในการทดสอบโปรแกรม

การทดสอบรวม (System Testing)

- ❑ มีวิธีการคล้ายกับการทดสอบแบบ Integration แตกต่างกันตรงที่ Integration Testing จะทดสอบโดยใช้โมดูลเพิ่มไปเรื่อยๆ จนกระทั่งครบทุกโมดูลของโปรแกรม แต่ System Testing จะทดสอบจากโปรแกรมเพิ่มไปเรื่อยๆ จนกระทั่งครบทุกโปรแกรมของระบบงาน ว่าโปรแกรมทุกโปรแกรมเมื่อทำงานร่วมกันแล้วจะให้ผลลัพธ์ที่ถูกต้องหรือไม่
- ❑ นอกจากนี้แล้ว System Testing ยังเป็นการทดสอบระบบงานว่าสามารถทำงานให้ผลลัพธ์ที่มีประสิทธิภาพเป็นที่ยอมรับหรือไม่ และเพื่อให้มั่นใจได้ว่าระบบงานนั้นสามารถตอบสนองความต้องการของผู้ใช้ได้อย่างตรงจุดมากที่สุด

2.2 กลยุทธ์ในการทดสอบระบบ

2.2.1 การทดสอบการทำงานสูงสุด (Peak Load Testing)

2.2.2 การทดสอบประสิทธิภาพของเวลา
(Performance Testing)

2.2.3 การทดสอบการกู้ระบบ (Recovery Testing)

2.2.4 การทดสอบการเก็บข้อมูล (Storage Testing)

2.2.5 การทดสอบกระบวนการ (Procedure Testing)

2.2.6 การทดสอบผู้ใช้ (User Testing)

2.2.1 การทดสอบการทำงานสูงสุด (Peak Load Testing)

- ❑ เป็นการทดสอบประสิทธิภาพในการประมวลผลของระบบ เมื่อมีการทำรายการมากที่สุด ณ ช่วงเวลาใดเวลาหนึ่ง
- ❑ เพื่อทดสอบว่าระบบจะสามารถรองรับการทำรายการคำสั่งมากที่สุดได้เพียงใด และนานเท่าใดเมื่อต้องประมวลผลจำนวนรายการคำสั่งที่มากที่สุดดังกล่าวในช่วงเวลาหนึ่ง

2.2.2 การทดสอบประสิทธิภาพของเวลา (Performance Testing)

- ❑ เป็นการทดสอบระบบ เพื่อพิจารณาถึงช่วงเวลาที่ใช้ในการประมวลผลรายการ ว่าใช้ระยะเวลาสั้นเพียงใดในการทำรายการ
- ❑ ทั้งการประมวลผลแบบกลุ่ม (Batch Processing) หรือการประมวลผลแบบออนไลน์ (On-line Processing)
- ❑ รวมทั้งทดสอบช่วงเวลาที่ใช้ในการเข้าถึงข้อมูลแบบลำดับ (Sequential Access) และแบบสุ่ม (Random Access) ด้วย

2.2.3 การทดสอบการกู้ระบบ (Recovery Testing)

- ❑ เป็นการทดสอบความสามารถในการกู้ระบบกรณีที่ระบบล่ม
- ❑ ความสามารถในการกู้ระบบนี้รวมทั้งการกู้ข้อมูลด้วย

2.2.4 การทดสอบการเก็บข้อมูล (Storage Testing)

- ❑ เป็นการทดสอบความสามารถของระบบในการเก็บข้อมูล ว่าสามารถเก็บข้อมูลได้สูงสุดเป็นจำนวนเท่าใด
- ❑ เพื่อเตรียมการรองรับจำนวนข้อมูลที่อาจจะเพิ่มมากขึ้นในอนาคต

2.2.5 การทดสอบกระบวนการ (Procedure Testing)

- เป็นการทดสอบการจัดทำเอกสารคู่มือการดำเนินงานของระบบ และคู่มือการใช้งานสำหรับผู้ใช้ ว่าสามารถสร้างความเข้าใจให้กับผู้ใช้ได้มากน้อยเพียงใด และเมื่อเกิดปัญหาในเบื้องต้น ผู้ใช้สามารถอ่านคู่มือเพื่อแก้ไขปัญหานั้นได้หรือไม่

2.2.6 การทดสอบผู้ใช้ (User Testing)

- หรือ Human Factors Testing
- เป็นการทดสอบการใช้งานจริงของระบบเพื่อต้องการทราบว่าผู้ใช้จะทำอย่างไรเมื่อพบปัญหาที่เกิดขึ้น

2.3 การทดสอบการยอมรับของระบบโดยผู้ใช

- ❑ ซึ่งเป็นการทดสอบที่สำคัญเทียบเท่ากับการทดสอบโปรแกรม เนื่องจากการพัฒนาระบบนั้นเพื่อตอบสนองความต้องการในการดำเนินงานของผู้ใช้ระบบ
- ❑ ระบบงานใหม่ จะสามารถติดตั้งได้ต้องผ่านการยอมรับจากผู้ใช้
- ❑ แบ่งออกเป็น 2 ประเภทคือ

2.3.1 Alpha Testing

2.3.2 Beta Testing

2.3.1 Alpha Testing

- ❑ การทดสอบความสมบูรณ์ของระบบโดยผู้ใช้ และใช้ข้อมูลสมมติในการทดสอบ ในการทดสอบประเภทนี้ จะสมมติให้ระบบอยู่ในสถานการณ์ที่อาจจะเกิดขึ้นได้จาก Alpha Testing นี้จะทำให้ทราบได้ว่าระบบมีข้อผิดพลาดอะไรเกิดขึ้นบ้าง โดยการทดสอบจะมี 4 ประเภทคือ
 - การทดสอบการกู้ระบบ (Recovery Testing)
 - การทดสอบความปลอดภัยของระบบ (Security Testing)
 - การทดสอบประสิทธิภาพการทำงานของระบบภายใต้ความกดดัน (Stress Testing)
 - การทดสอบประสิทธิภาพการทำงานของระบบภายใต้สภาพแวดล้อมของคอมพิวเตอร์ (Performance Testing)

2.3.2 Beta Testing

- ❑ คือ การทดสอบความสมบูรณ์ของระบบโดยผู้ใช้ และใช้ข้อมูลจริงในการทดสอบและภายใต้สถานการณ์ที่เกิดขึ้นจริง
- ❑ การทดสอบประเภทนี้ถือว่าการซ้อมติดตั้งระบบเพื่อใช้งานจริง เนื่องจากเป็นการทดสอบระบบอย่างสมจริงไม่ว่าจะเป็นสถานการณ์ ข้อมูล ขั้นตอนการดำเนินงาน เอกสารคู่มือ การฝึกอบรม การสนับสนุนการทำงาน
- ❑ รวมทั้งยังเป็นการแก้ปัญหาที่พบจากการทดสอบแบบ Alpha Testing ด้วย

3. การติดตั้งระบบ (Installation)



Installation

- 3.1 การติดตั้งระบบทันทีทันใด (Direct Installation)
- 3.2 การติดตั้งแบบขนาน (Parallel Installation)
- 3.3 การติดตั้งแบบนำร่อง (Single Location Installation/Pilot Installation)
- 3.4 การติดตั้งแบบทยอยติดตั้งเป็นระยะ (Phased Installation)

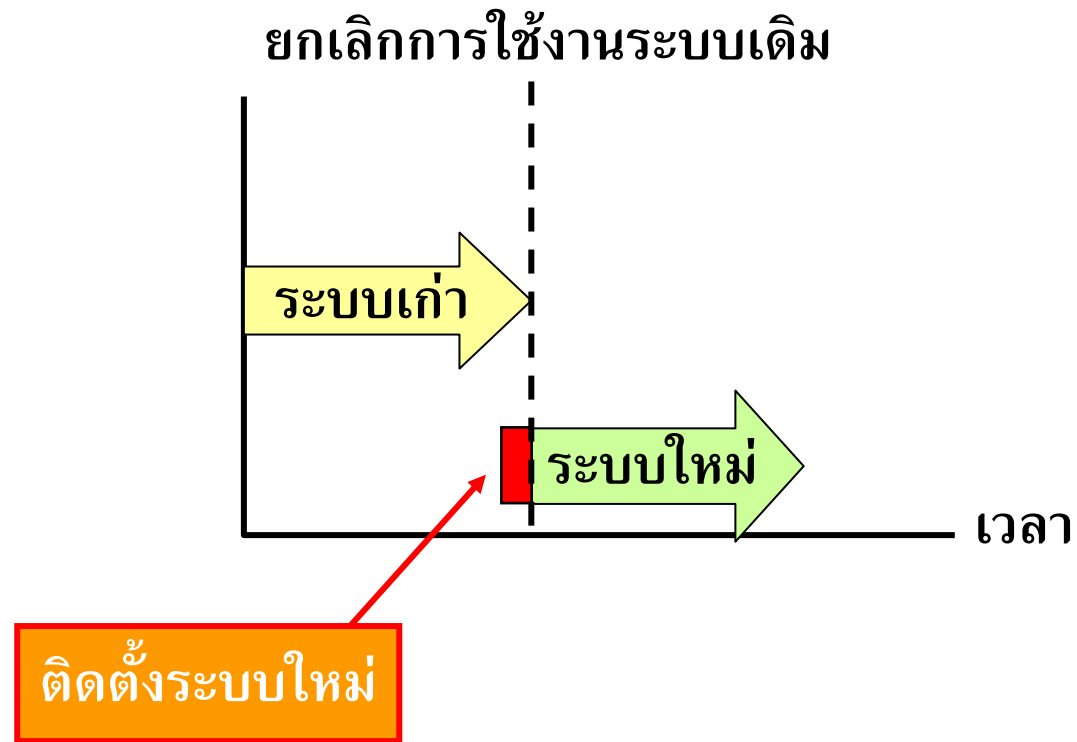
3.1 Direct Installation

- ❑ เป็นวิธีการติดตั้งที่มีการใช้ระบบงานใหม่ทันที และระบบงานเก่ายกเลิกการใช้งานทันทีเช่นเดียวกัน
- ❑ วิธีการนี้องค์กรเสียค่าใช้จ่ายน้อยแต่มีความเสี่ยงสูง เนื่องจากในการใช้งานจริงนั้นข้อมูลที่นำเข้าสู่ระบบมีความเป็นไปได้ในการเกิดข้อผิดพลาดขึ้นได้มากกว่าข้อมูลที่สมมติขึ้นเพื่อการทดสอบ
- ❑ หากเกิดปัญหาในระบบจะส่งผลกระทบต่อผู้ใช้ระบบ จนทำให้ระบบหยุดชะงักได้ วิธีการแก้ปัญหายังหนึ่งคือนำระบบเก่าเข้ามาใช้งานอีกครั้ง แต่จะเสียเวลามาก เนื่องจากต้องปรับปรุงข้อมูลในระบบเก่าทั้งหมด

3.1 Direct Installation (ต่อ)

- ❑ ข้อดีคือ สามารถเลื่อนกำหนดการใช้งานระบบใหม่ออกไปได้ หากองค์กรหรือผู้ใช้งานยังไม่มีความพร้อมสำหรับระบบใหม่
- ❑ ระบบที่พร้อมจะติดตั้งเพื่อใช้งาน จะต้องผ่านการทดสอบอย่างหนักในทุกสถานการณ์ และผ่านการแก้ไขข้อผิดพลาดที่เกิดขึ้นจากการทดสอบมาแล้วอย่างครบถ้วน

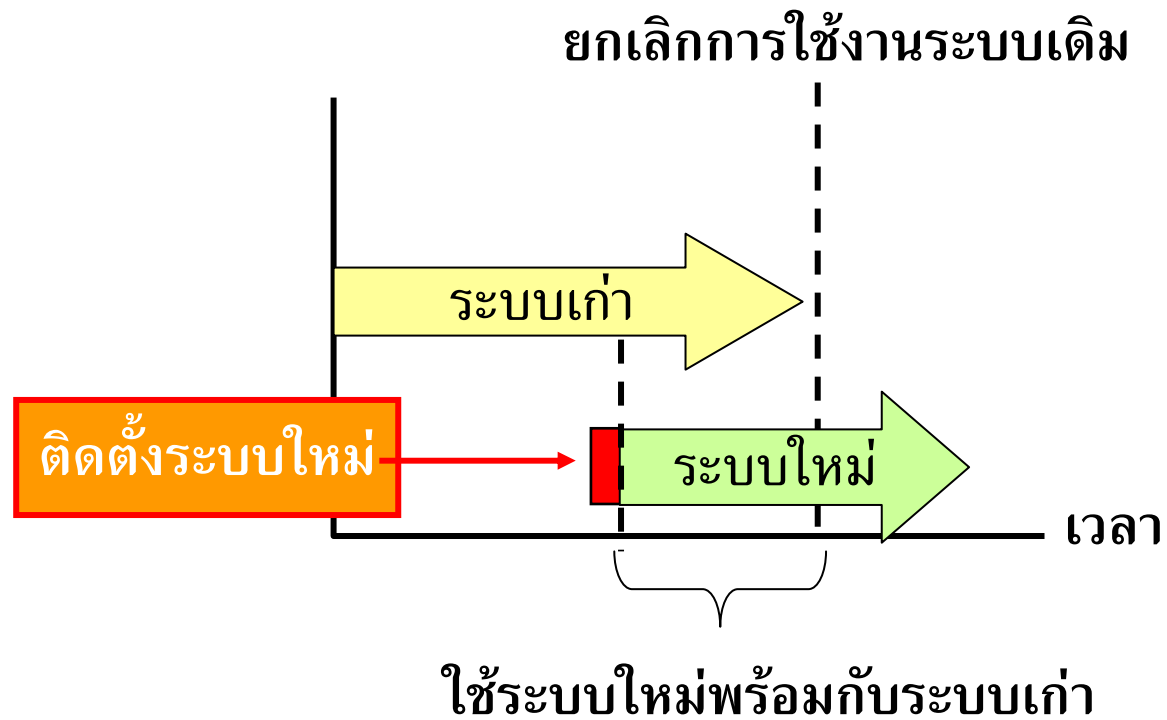
Direct Installation



3.2 Parallel Installation

- ❑ เป็นวิธีการที่มีการใช้ระบบงานใหม่ไปพร้อม ๆ กับการใช้ระบบงานเก่า จนกว่าผู้ใช้และผู้บริหารจะพอใจระบบใหม่และตัดสินใจที่จะหยุดใช้ระบบเก่า
- ❑ ต้องใช้ค่าใช้จ่ายค่อนข้างสูง เนื่องจากการดำเนินงานสองระบบในเวลาเดียวกัน ซึ่งหมายถึงการเพิ่มพนักงานบางตำแหน่งเพื่อดูแลระบบใหม่ ทั้งยังต้องคอยดูแล บำรุงรักษาระบบให้มีความสมบูรณ์และมีประสิทธิภาพมากที่สุด ในบางครั้งอาจทำให้ผู้ใช้ระบบเกิดความสับสนได้
- ❑ แต่วิธีการนี้จะทำให้สามารถเปรียบเทียบผลของการดำเนินงานระหว่างระบบใหม่กับระบบเก่าได้

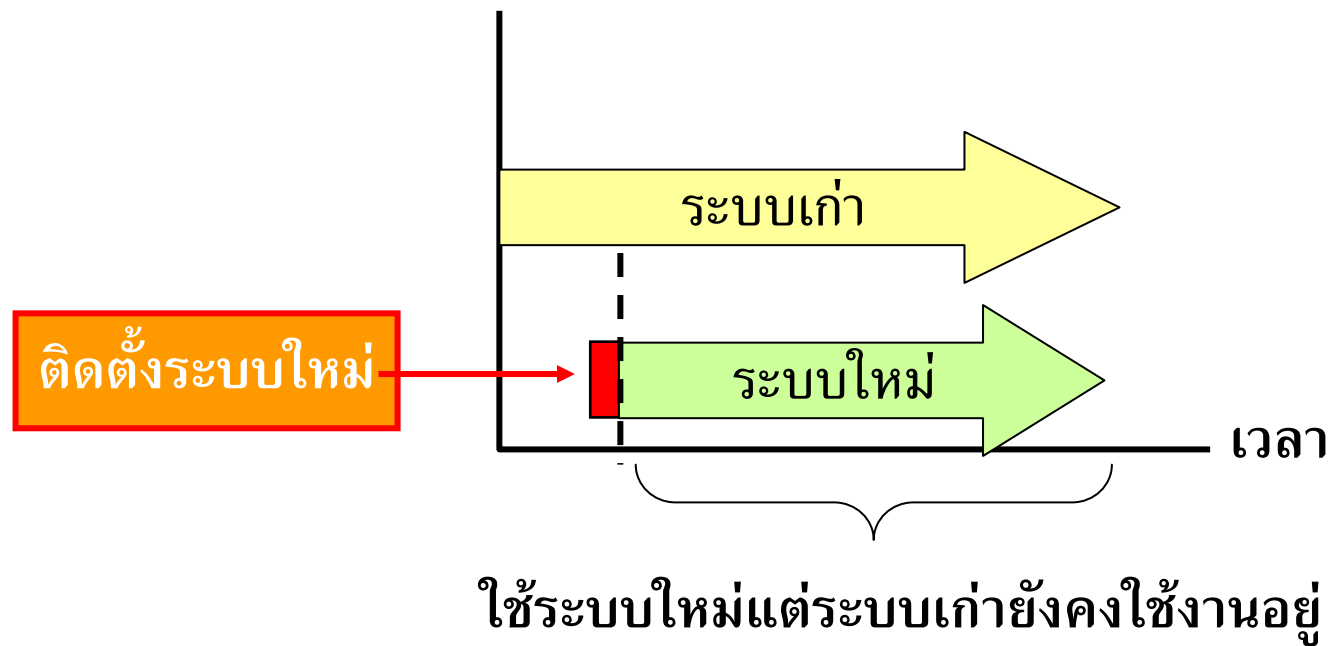
Parallel Installation



3.3 Single Location Installation/Pilot Installation

- ❑ เป็นวิธีการที่มีการใช้ระบบงานใหม่เพียงหน่วยเดียวขององค์กร ก่อนเพื่อเป็นการนำร่อง แล้วจึงค่อยปรับเปลี่ยนทั้งหมดเมื่อ เห็นว่าระบบใหม่นั้นลงตัวแล้ว
- ❑ ข้อดีคือ เสียค่าใช้จ่ายน้อยกว่าสองวิธีแรก
- ❑ เมื่อเกิดข้อผิดพลาดหรือความเสียหายเนื่องจากระบบงานใหม่ ความเสียหายนั้นจะจำกัดอยู่เพียงแค่สถานที่ขององค์กรที่จัดให้มีการติดตั้งระบบใหม่แบบนำร่องเท่านั้น หรือจำกัดอยู่เพียง สาขาเดียว ไม่ทำให้การดำเนินธุรกิจของทั้งองค์กรเสียหายไปด้วย

Single Location Installation/Pilot Installation



3.4 Phased Installation

- ❑ หรือ “Staged Installation” เป็นวิธีการที่มีการใช้ระบบงานใหม่เพียงบางส่วนก่อนระยะหนึ่งควบคู่ไปกับระบบงานเก่า แล้วจึงค่อย ๆ ทดลองใช้ระบบงานใหม่เพิ่มขึ้นทีละส่วนจนกระทั่งครบทุกส่วนของระบบงานใหม่อย่างเต็มรูปแบบในที่สุด
- ❑ มีลักษณะคล้ายกับแบบนำร่อง คือเริ่มจากจุดเดียวก่อน แตกต่างกันตรงที่วิธีแบบทยอยติดตั้งเป็นระยะ ไม่คำนึงถึงสถานที่ แต่คำนึงถึงระบบงานย่อยโดยการติดตั้งทีละระบบ ซึ่งอาจจะกระจายไปตามสาขาต่าง ๆ ที่มีการใช้ระบบงานย่อยนั้นเมื่อระบบย่อยที่ใช้เริ่มแรกสมบูรณ์แล้ว จึงเริ่มนำระบบย่อยต่อไปมาใช้งาน เป็นเช่นนี้ไปเรื่อย ๆ จนครบทั้งระบบในที่สุด

3.4 Phased Installation (ต่อ)

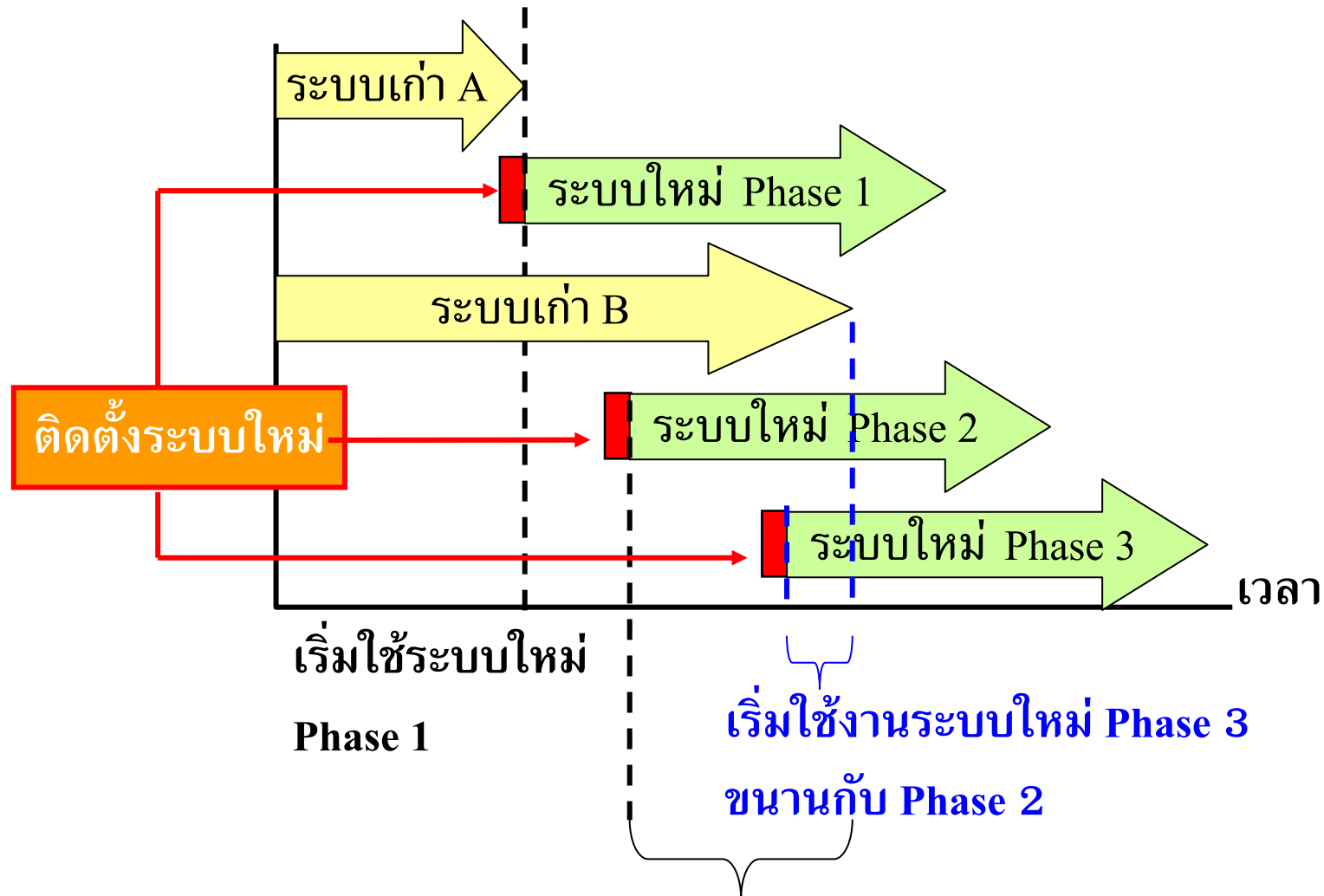
□ ข้อดี

- สามารถจำกัดความเสี่ยงที่เกิดขึ้นได้ เนื่องจากการปรับเปลี่ยนเป็นช่วงระยะเวลาและบางสถานที่ความเสี่ยงหรือข้อผิดพลาดจึงเกิดขึ้นในช่วงที่มีการเปลี่ยนแปลงและสถานที่ที่ใช้ระบบงานใหม่เท่านั้น

□ ข้อเสีย

- ความไม่สอดคล้องของการดำเนินงานในระบบใหม่และระบบเก่า เนื่องจากการนำระบบงานย่อยของระบบใหม่ เข้ามาใช้ร่วมกับระบบเก่า เช่น หากมีการนำระบบการสั่งซื้อซึ่งเป็นระบบใหม่ เข้ามาใช้งานก่อนอาจจะเกิดปัญหาในการติดต่อระหว่างฐานข้อมูลของระบบใหม่กับฐานข้อมูลของระบบเก่าได้

Phased Installation



เริ่มใช้งาน Phase 2 ขนานกับระบบเก่า และระบบใหม่ Phase 3

4. การจัดทำเอกสาร (Documentation)



Documentation

- ❑ ในระหว่างการพัฒนากระบวนการใหม่ จะเกิดเอกสารแสดงรายละเอียดของกระบวนการมากมาย ไม่ว่าจะเป็นเอกสารที่เกิดจากขั้นตอนการวิเคราะห์ระบบ หรือออกแบบระบบก็ตาม
- ❑ เมื่อถึงขั้นตอนการติดตั้งระบบแล้ว ชุดเอกสารเหล่านั้นจะต้องทำการปรับปรุงให้เป็นเอกสารที่แสดงรายละเอียดของระบบใหม่ที่ถูกต้องมากที่สุด
- ❑ นอกจากนี้ในขั้นตอนการติดตั้ง SA ยังจะต้องจัดทำเอกสารคู่มือสำหรับผู้ใช้งาน เพื่อแสดงรายละเอียดขั้นตอนการทำงาน ภาพรวมของระบบใหม่ทั้งหมด เพื่อใช้อ้างอิงในระหว่างการใช้งานระบบใหม่

Documentation (ต่อ)

4.1 การจัดทำเอกสารของระบบ (System Documentation)

4.2 การจัดทำเอกสารของผู้ใช้ (User Documentation)

4.1 การจัดทำเอกสารของระบบ (System Documentation)

- ❑ เอกสารของระบบ คือ เอกสารที่แสดงขั้นตอนการทำงานภายในของระบบ และรายละเอียดข้อมูลเฉพาะของการออกแบบระบบ เช่น แบบจำลองชนิดต่างๆ ที่ใช้เป็นเครื่องมือในระหว่างการวิเคราะห์ระบบ (Diagrams) พจนานุกรมข้อมูล (Data Dictionary) แบบร่างสำหรับการออกแบบแบบฟอร์ม รายงานหน้าจอ และต้นแบบ เป็นต้น
- ❑ จำแนกเอกสารของระบบนี้ออกเป็น 2 กลุ่ม ได้แก่
 - 4.1.1 เอกสารภายใน (Internal Documentation)
 - 4.1.2 เอกสารภายนอก (External Documentation)

4.1.1 เอกสารภายใน (Internal Documentation)

- ❑ คือเอกสารของระบบที่แสดงรายละเอียดส่วนของโค้ดโปรแกรมที่เขียนขึ้น พร้อมทั้งคำอธิบายโปรแกรม ผลการรันโปรแกรม และอื่น ๆ ที่เกี่ยวข้องกับโปรแกรมระบบ

4.1.2 เอกสารภายนอก (External Documentation)

- ❑ คือเอกสารที่เป็นแผนภาพ และแบบจำลองชนิดต่าง ๆ ของระบบ เช่นแผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) แผนภาพแสดงความสัมพันธ์ของข้อมูล (Entity Relationship Diagram : ERD) แผนผังโครงสร้าง (Structure Chart) เป็นต้น

4.2 การจัดทำเอกสารของผู้ใช้ (User Documentation)

- ❑ เอกสารของผู้ใช้ คือ เอกสารที่จัดทำขึ้นเพื่อแสดงขั้นตอนการใช้งานระบบ และวิธีการใช้งานโปรแกรม เอกสารสำหรับผู้ใช้อักรวมถึงคู่มือการใช้งาน (User's Guide) คู่มือสำหรับบริหารระบบ (System Administration's Guide)
- ❑ คู่มือเหล่านี้อาจจะอยู่ในรูปของเอกสารเป็นเล่ม และแสดงไว้ในส่วนช่วยเหลือ (Help) ภายในโปรแกรมก็ได้

4.2 การจัดทำเอกสารของผู้ใช้ (ต่อ)

- ❑ การจัดรูปแบบของการอธิบายรายละเอียดการใช้งานโปรแกรมจะต้องมีรูปแบบที่น่าสนใจ อ่านเข้าใจง่าย ค้นหาได้ง่าย แสดงหัวข้อชัดเจน รวมทั้งแสดงถึงปัญหาที่อาจจะเกิดขึ้นเนื่องจากการใช้งานและวิธีแก้ปัญหา
- ❑ โดยการจัดรูปแบบจะมีลักษณะที่คล้ายกับการออกแบบส่วนช่วยเหลือของโปรแกรมในขั้นตอนการออกแบบระบบนั่นเอง

5. การฝึกอบรมพนักงาน (Training)



Training

- ❑ กรณีที่ระบบงานนั้นเป็นระบบใหม่ทั้งหมด การอาศัยความเข้าใจจากคู่มือการใช้งานโปรแกรมเพียงอย่างเดียว อาจจะไม่เพียงพอ แต่จะต้องสร้างทักษะเริ่มต้นในการดำเนินงานให้แก่ผู้ใช้งานอีกด้วย
- ❑ องค์กรและ SA จึงต้องร่วมมือกัน จัดหลักสูตรฝึกอบรมการใช้งานโปรแกรม ซึ่งมีวิธีการและเทคนิคแตกต่างกันไปแต่ละองค์กร ขึ้นอยู่กับสภาพแวดล้อมและสถานการณ์ในขณะนั้นองค์กร
- ❑ ปัจจัยสำคัญที่ส่งผลกระทบต่อการจัดหลักสูตรฝึกอบรมให้แก่พนักงานนั้นก็คือ “ค่าใช้จ่าย” ดังนั้นรูปแบบในการฝึกอบรมพนักงานจึงมีแตกต่างกันไป

การจัดฝึกอบรมในลักษณะต่าง ๆ

1. จัดเป็นหลักสูตรฝึกอบรมเป็นกลุ่ม
2. ฝึกอบรมด้วยคอมพิวเตอร์ช่วยสอนเช่น CAI (Computer-aided Instruction), CBI (Computer-base Training) หรือ WBT (Web-Base Training)
3. ฝึกอบรมด้วย CAI พร้อมกับมีผู้เชี่ยวชาญคอยแนะนำ
4. ผู้ใช้ฝึกอบรมด้วยตัวเองจากส่วนช่วยเหลือของโปรแกรม
5. ฝึกอบรมโดยผู้จำหน่ายโปรแกรมของระบบ หรือกลุ่มผู้รับพัฒนาระบบภายนอกองค์กร
6. ฝึกอบรมผ่านทางอินเทอร์เน็ตกรณีที่องค์กรมีสาขาอยู่ต่างจังหวัด และต้องการลดค่าใช้จ่ายในการเดินทาง

6. การบริการให้ความช่วยเหลือ หลังการติดตั้งระบบ (Support)



Support

- ❑ คือ การจัดเตรียมความช่วยเหลือเพื่อการแก้ปัญหาที่เกิดขึ้นกับผู้ใช้หลังจากติดตั้งและใช้งานระบบใหม่
- ❑ สำหรับองค์กรอาจจะมีการแต่งตั้งทีมงานเพื่อบริการความช่วยเหลือ ซึ่งอาจจะมีการกำหนดหน้าที่ในการช่วยเหลือได้หลายลักษณะ เช่น
 - ช่วยเหลือเมื่อมีการติดตั้งโปรแกรมใหม่หรือฮาร์ดแวร์ใหม่
 - บริการให้คำปรึกษาและความช่วยเหลือแก่ผู้ใช้งานเมื่อเกิดปัญหาในเบื้องต้น
 - แบ่งแยกข้อมูลจากฐานข้อมูลไปสู่เครื่องคอมพิวเตอร์ของผู้ใช้งาน

ประเภทของการบริการให้ความช่วยเหลือหลังการติดตั้งระบบ

6.1 การให้ความช่วยเหลือแบบอัตโนมัติ (Automating Support)

6.2 Help Desk

6.1 การให้ความช่วยเหลือแบบอัตโนมัติ (Automating Support)

- ❑ สำหรับองค์กรที่มีขนาดเล็ก การจัดตั้งทีมงานเพื่อคอยให้ความช่วยเหลือเมื่อเกิดปัญหาขึ้นนั้น จะทำให้เสียค่าใช้จ่ายในส่วนของการตอบแทนแรงงานส่วนนั้น การลดค่าใช้จ่ายในส่วนนั้น องค์กรอาจจะใช้วิธีการให้การช่วยเหลือแบบอัตโนมัติ ด้วยเทคโนโลยีการติดต่อสื่อสารที่ทันสมัยในปัจจุบัน เช่น
 - On-line Support Forum
 - Bulletin Board
 - Voice-response System

On-line Support Forum

- ❑ ผู้ใช้สามารถตั้งกระทู้ไว้เพื่อแนะนำเทคนิคการใช้งานให้กับผู้ใช้คนอื่นได้ทราบ หรือสามารถเปิดประเด็นตั้งคำถามได้โดยผ่านเครือข่ายอินเทอร์เน็ต
- ❑ หรือกรณีที่เป็นองค์กรขนาดเล็กจะสามารถทำได้โดยผ่านเครือข่ายอินเทอร์เน็ต

Bulletin Board

- ❑ กระดานฝากข้อความหรือข่าวสาร กรณีเช่นนี้ทางองค์กรเองสามารถให้ความรู้หรือข่าวสารแก่ผู้ใช้งานได้โดยฝากเป็นข้อความไว้ที่กระดานฝากข้อความนี้ได้อัตโนมัติ

Voice-response System

- ❑ ระบบเสียงตอบรับอัตโนมัติ ผู้ใช้ระบบสามารถเรียกไปยังหมายเลขที่ทางองค์กรกำหนดไว้ โดยระบบเสียงตอบรับอัตโนมัติจะมีเมนูให้เลือกโดยการกดปุ่มโทรศัพท์ตามหัวข้อที่ผู้ใช้ต้องการ
- ❑ นอกจากนี้องค์กรยังสามารถให้การสนับสนุนหลังการติดตั้งระบบโดยผ่านทาง E-mail ไปยังผู้ใช้ระบบที่เป็นลูกค้าขององค์กรได้อีกด้วย

6.2 Help Desk

- ❑ คือ จุดให้ความช่วยเหลือแก่ผู้ใช้ระบบ ไม่ว่าจะเป็นปัญหาด้านใดก็ตามที่เกิดขึ้นจากการดำเนินงานของระบบ
- ❑ ในปัจจุบันจะสามารถพบกับ Help Desk ได้มากมายซึ่งจะมีชื่อเรียกแตกต่างกันไป และองค์กรเหล่านี้ส่วนใหญ่จะเป็นองค์กรที่มีผู้ใช้ระบบเป็นลูกค้า ซึ่งจะเรียก Help Desk ขององค์กรแตกต่างกันไป เช่น
 - Call Center
 - Customer Service
 - Web call center
 - Customer Care

6.2 Help Desk (ต่อ)

- ❑ องค์กรที่จัดให้มี Help Desk จะต้องจัดฝึกอบรมให้กับทีมงานที่จะมาทำหน้าที่ในส่วนนี้เป็นอย่างดี ทีมงานนั้นจะต้องมีความรู้และเข้าใจเกี่ยวกับการดำเนินงานของระบบ และมีทักษะในการตอบคำถามที่ผู้ใช้ระบบซักถาม พร้อมทั้งสามารถยอมรับฟังปัญหาต่าง ๆ ที่เกิดขึ้นได้อย่างเต็มใจ

สรุปค่าใช้จ่ายในการจัดทำระบบสารสนเทศในแต่ละเฟส

