**สัปดาห์ที่ 1**

- **เรื่องทั่วไปเกี่ยวกับการวิเคราะห์และออกแบบระบบ**
- **หลักการวิเคราะห์และออกแบบระบบ**

.....

**ระบบคืออะไร**

ระบบคือกลุ่มขององค์การต่างๆ ที่ทำงานร่วมกันเพื่อจุดประสงค์อันเดียวกัน ระบบอาจจะประกอบด้วยบุคคลากร เครื่องมือ เครื่องใช้ พัสดุ วิธีการ ซึ่งทั้งหมดนี้จะต้องมีระบบจัดการอันหนึ่งเพื่อให้บรรลุจุดประสงค์อันเดียวกัน ตัวอย่างเช่น ในร่างกายคนเราจะมีระบบในตัวคือมีความสัมพันธ์ติดต่อกันระหว่างสอง เส้นประสาท เซลล์รับรู้ความรู้สึก เพื่อบรรลุเป้าหมายในการรับรู้ความรู้สึกร้อนหนาว เป็นต้น ในการใช้ภาษาก็ถือทำอย่างเป็นระบบนั่นระบบนั่นคือ ความสัมพันธ์ติดต่อกันระหว่าง การใช้คำสัญลักษณ์ต่างๆ เพื่อบรรลุเป้าหมายในการตีความให้เข้าใจภาษานั้นๆ ในธุรกิจก็เป็นระบบอย่างหนึ่ง ซึ่งมีส่วนประกอบคือ การตลาด โรงงาน การขาย การค้นคว้า การขนส่ง การเงิน บุคคล การทำงาน โดยที่ทั้งหมดมีการติดต่อสัมพันธ์ระหว่างกัน เพื่อให้บรรลุเป้าหมายให้เกิดกำไร

**องค์ประกอบของระบบการเรียนการสอน**

1. บุคคลากรได้แก่ ครูและนักเรียน
2. เครื่องมือได้แก่ เครื่องฉายแผ่นใส ชอล์ก กระดานดำ
3. พัสดุได้แก่ โต๊ะ เก้าอี้
4. วิธีการได้แก่ เขียนบนกระดานดำ ใช้แผ่นใส หรืออื่นๆ
5. การจัดการได้แก่ โรงเรียนจัดตารางเรียน เก็บเงินค่าเล่าเรียน จ่ายค่าสอนให้แก่ครู

เมื่อเราศึกษาระบบใดระบบหนึ่ง เราควรจะต้องเข้าใจการทำงานของระบบนั้นให้ดีโดย การถามตัวเองตลอดเวลาด้วยคำถามเหล่านี้

1. ระบบทำอะไร ( What )
2. ทำโดยใคร ( Who )
3. ทำเมื่อไร ( When )
4. ทำอย่างไร ( How )

**การวิเคราะห์ระบบและการออกแบบ ( System Analysis and Design)**

การวิเคราะห์และออกแบบระบบคือ วิธีการที่ใช้ในการสร้างระบบสารสนเทศขึ้นมาใหม่ในธุรกิจใดธุรกิจหนึ่ง หรือระบบย่อยของธุรกิจ นอกจากการสร้างระบบสารสนเทศใหม่แล้ว การวิเคราะห์ระบบช่วยในการแก้ไขระบบสารสนเทศเดิมที่มีอยู่แล้วให้ดีขึ้นด้วยก็ได้ การวิเคราะห์ระบบคือ การหาความต้องการ ( Requirements) ของระบบสารสนเทศว่าคืออะไร หรือต้องการเพิ่มเติมอะไรเข้ามาในระบบและการออกแบบก็คือ การนำเอาความต้องการของระบบมาเป็นแบบแผนหรือเรียกว่าพิมพ์เขียว ในการสร้างระบบสารสนเทศนั้นให้ใช้ในงานได้จริง ผู้ที่ทำหน้านี้ก็คือ นักวิเคราะห์และออกแบบระบบ ( System Analysis : SA )

**นักวิเคราะห์ระบบคือใคร** ?คอมพิวเตอร์เป็นเพียงเครื่องมือที่ใช้สำหรับเก็บรวบรวมและประมวลผลให้กับผู้ใช้โดยให้ประโยชน์ต่อผู้ใช้คือ ความรวดเร็วและความถูกต้องของข้อมูล ซึ่งเป็นหัวใจสำคัญต่อการบริหารของธุรกิจในปัจุบันที่มีการแข่งขันสูง

**ผู้ใช้** ( Users ) จึงเป็นผู้กำหนดปัญหาและแนวทางของระบบงานที่นำมา แก้ไขซึ่งปัญหาแต่ผู้ใช้เองไม่ทราบวิธีจะนำเอาคอมพิวเตอร์มาใช้แก้ปัญหา หรือช่วยเหลือในการบริหาร ในทางตรงกันข้ามโปรแกรมเมอร์ ( programmers)และช่างเทคนิค ( technicians)เป็นผู้ที่สามารถจะใช้เทคโนโลยีของคอมพิวเตอร์และป้อนคำสั่งให้คอมพิวเตอร์ทำงานได้ต้องการ แต่โปรแกรมเมอร์หรือช่างเทคนิคมักจะไม่เข้าใจถึงระบบธุรกิจมากนัก ดังนั้น ช่องว่างระหว่างนักธุรกิจหรือระบบงานในหน่วยงานต่างๆ กับโปรแกรมเมอร์หรือกับช่างเทคนิคจึงอาจเกิดขึ้นได้ นักวิเคราะห์ระบบจึงทำหน้าที่เป็นผู้สมานช่องว่างนี้ นักวิเคราะห์ระบบเป็นผู้ที่เกี่ยวข้องโดยตรงที่จะนำเอาความเข้าใจและเทคโนโลยีของคอมพิวเตอร์มาใช้ ในการพัฒนาระบบงานข้อมูลเพื่อช่วยแก้ปัญหาให้กับงานในหน่วยงานต่างๆ

**บทบาทของนักวิเคราะห์ระบบ**

นักวิเคราะห์ระบบจะเป็นผู้ที่ศึกษาถึงปัญหาและความต้องการของนักธุรกิจ โดยนำเอาปัจจัย 3 ประการ คือ คน ( people ) วิธีการ ( method ) และคอมพิวเตอร์เทคโนโลยี ( computer technology ) ใช้ในการปรับปรุงหรือแก้ปัญหาให้กับนักธุรกิจ เมื่อได้มีการนำเอาพัฒนาการทางเทคโนโลยีของคอมพิวเตอร์มาใช้ นักวิเคราะห์ระบบจะต้องรับผิดชอบถึงการกำหนดลักษณะของข้อมูล ( data ) ที่จะจัดเก็บเข้าสู่ระบบงานคอมพิวเตอร์ การหมุนเวียน การเปลี่ยนแปลงของข้อมูลและระยะเวลาเพื่อให้เกิดประโยชน์สูงสุดต่อผู้ใช้หรือธุรกิจ ( business users )นักวิเคราะห์ระบบไม่ได้เพียงวิเคราะห์หรือดีไซน์ระบบงานเท่านั้น หากแต่ยังขายบริการทางด้านระบบงานข้อมูล โดยนำเอาประโยชน์จากเทคโนโลยีล่าสุดมาใช้ควบคู่กันไปด้วย จากบทบาทของนักวิเคราะห์ระบบที่กล่าวมาแล้วข้างต้น ทำให้นักวิเคราะห์ระบบจะต้องมีความรู้ทั้งทางภาคธุรกิจหรือการดำเนินงาน ในหน่วยงานต่างๆ และคอมพิวเตอร์ควบคู่กัน นักวิเคราะห์ระบบโดยส่วนใหญ่สามารถที่จะดีไซน์ระบบงานและเขียนโปรแกรมขึ้นได้ด้วยตัวเอง ส่วนนี้เองกลับทำให้บุคคลภายนอกเกิดความสับสนระหว่างโปรแกรมเมอร์กับนักวิเคราะห์ระบบ

**ความแตกต่างระหว่างโปรแกรมเมอร์และนักวิเคราะห์ระบบ**

โปรแกรมเมอร์ ( programmer ) หมายถึงบุคคลที่รับผิดชอบในด้านการเขียนโปรแกรม สิ่งที่เขาจะเชื่อมโยงนั้นได้แก่ อุปกรณ์คอมพิวเตอร์ ระบบปฏิบัติ ( Operating System :OS ) หรือแม้กระทั่งภาษาที่ใช้ในการเขียน เช่น COBOL, BASIC และ C++ งานของโปรแกรมเมอร์จะเป็นไปในลักษณะที่มีขอบเขต ที่แน่นอนคือโปรแกรมที่เขาเขียนขึ้นนั้นถูกต้องตามจุดประสงค์หรือไม่ กิจกรรมงานของโปรแกรมจะเกี่ยวข้องกับคนจำนวนน้อย เช่น กับโปรแกรมเมอร์ด้วยกันเอง หรือกับนักวิเคราะห์ระบบที่เป็นผู้วางแนวทางของระบบให้แก่เขา นักวิเคราะห์ระบบ หรือที่เรียกกันย่อๆ ว่า SA (SYSTEM ANALYSIS) นั้น นอกจากจะต้องรับผิดชอบต่อการโปรแกรมคอมพิวเตอร์แล้ว ยังจะต้องรับผิดชอบงานในส่วนที่เกี่ยวกับการจัดหาอุปกรณ์ต่างๆ เกี่ยวกับคอมพิวเตอร์ ผู้ที่จะใช้ระบบแฟ้มหรือฐานข้อมูลต่างๆ รวมทั้งข้อมูลดิบที่จะป้อนเข้าระบบงานของนักวิเคราะห์ระบบไม่ได้อยู่ในลักษณะที่แน่นอนแบบโปรแกรมเมอร์ ไม่มีคำตอบที่แน่นอนจากระบบที่เขาวางไม่ว่าผิดหรือถูก งานของเขาเกิดจากการประนีประนอมและผสมผสานของปัจจัยต่างๆ ที่เกี่ยวข้องกับระบบงาน คือ ผู้ใช้ วิธีการ เทคโนโลยี และอุปกรณ์จนได้ผลลัพธ์ที่เหมาะสมออกมาเป็นระบบงาน (APPLICATION SYSTEM) งานของนักวิเคราะห์ระบบจึงมักจะต้องเกี่ยวข้องกับคนหลายระดับ ตั้งแต่ลูกค้าหรือผู้ใช้ นักธุรกิจ โปรแกรมเมอร์ ผู้ตรวจสอบบัญชี หรือแม้กระทั่งเซลล์แมนที่ขายระบบงานข้อมูล แม้ว่างานของนักวิเคราะห์ระบบจะดูเป็นงานที่ยากและซับซ้อน แต่งานใน

ลักษณะนี้ก็เป็นงานที่ท้าทายให้กับบุคคลที่มีความคิดสร้างสรรค์และกว้างไกลเข้ามาอยู่เสมอ ความรู้สึกภาคภูมิใจที่ได้วางระบบงานออกมาเป็นรูปร่างและสามารถ ใช้ปฏิบัติได้จริง จะฝังอยู่ในสำนึกของเขาตลอดเวลา ความรู้สึกอันนี้คงจะถ่ายทอดออกมาเป็นตัวหนังสือไม่ได้ แต่จะทราบกันเองในหมู่ของนักวิเคราะห์ระบบด้วยกัน เพื่อให้เข้าใจความหมายของนักวิเคราะห์ระบบมากยิ่งขึ้น ตารางที่ 1 จะแสดงให้เห็นรายละเอียดของหน้าที่ของนักวิเคราะห์ระบบ

| Job Description |
| --- |
| **ตำแหน่งงาน** : **นักวิเคราะห์ระบบ** |
| **รายละเอียดของงาน** :<br>1. รวบรวมและวิเคราะห์ข้อมูลเพื่อพัฒนาระบบงานข้อมูล (INFORMATION SYSTEM) รับผิดชอบในการศึกษาถึงปัญหาและความต้องการของธุรกิจ เพื่อที่จะหาทางนำเอาอุปกรณ์คอมพิวเตอร์ วิธีการทางธุรกิจ รวมถึงบุคลากรต่างๆ มาใช้ในการพัฒนาเพื่อหาทางแก้ไขซึ่งปัญหา และบรรลุถึงความต้องการของธุรกิจนั้นๆ<br>2. ดีไซต์และจัดวางระบบฐานข้อมูล รวมทั้งการติดตั้งด้วย<br>3. ให้คำแนะนำและอบรมทั้งทางด้านเอกสารและการพบปะพูดจา หรือสัมมนาในหัวข้อของระบบงาน |
| **ความรับผิดชอบ** :<br>1. วิเคราะห์และประเมินผล เพื่อหาความเป็นไปได้ (Feasibility Study) ของระบบ<br>2. วิเคราะห์ซึ่งปัญหาที่เกิดขึ้นจากระบบธุรกิจที่เป็นอยู่<br>3. แจกแจงถึงสิ่งต่าง ๆ ที่จำเป็นจะร้องพัฒนาขึ้นเพื่อใช้หรือทดแทนระบบเดิม<br>4. กำหนดทางเลือกต่างๆ ที่เป็นไปได้ (Altermative solution) ในการแก้ปัญหา<br>5. เลือกอุปกรณ์คอมพิวเตอร์และซอฟต์แวร์ตามความเหมาะสม<br>6. ดีไซน์และวางระบบงานให้คล้องจองกัน เพื่อแก้ปัญหาของธุรกิจ<br>7. ให้คำแนะนำต่างๆ เมื่อระบบงานถูกนำมาใช้จริง |
| **หน้าที่** :<br>1. จัดทำงบประมาณค่าใช้จ่ายต่างๆ รวมทั้งด้านกำลังคน<br>2. กำหนดแผนงานและระยะเวลาที่ใช้ในการพัฒนาระบบงาน<br>3. ดำเนินการสัมภาษณ์และรวบรวมข้อมูลที่สำคัญต่อการพัฒนาระบบงาน<br>4. จัดทำเอกสารและวิเคราะห์ระบบงานธุรกิจในปัจจุบัน<br>5. พัฒนาระบบงานโดยใช้เทคโนโลยีปัจจุบันเพื่อปัญหาให้แก่ธุรกิจ<br>6. วิเคราะห์ถึงความเหมาะสมต่างๆ ของเทคโนโลยี การปฏิบัติการ และฐานทางเศรษฐกิจ<br>7. ทบทวนและยื่นข้อเสนอของระบบงานเพื่อพิจารณาอนุมัติ<br>8. ดีไซน์และตรวจสอบความถูกต้องของระบบงาน<br>9. ดีไซน์แฟ้มหรือฐานข้อมูลและโครงสร้างต่างๆ ที่ใช้ในระบบ<br>10. ดีไซน์ลักษณะการติดต่อระหว่างผู้ใช้ระบบกับระบบงานคอมพิวเตอร์ ( USERS) กับโปรแกรมเมอร์ (PROGRAMMERS) อย่างไรก็ตามธุรกิจหรือหน่วยงานต่างๆ จึงมักจะมีความคิดที่ว่า<br>11. ดีไซน์วิธีการเก็บข้อมูลและเทคนิค |

12. ดีไซน์ระบบรักษาความปลอดภัย และการควบคุมระบบ
13. ให้คำแนะนำทางด้านการเขียนโปรแกรมเพื่อให้ระบบดำเนินไปได้ตามเป้าหมาย
14. วางแผนงานต่างๆ เพื่อให้ระบบได้พัฒนาขึ้นใหม่ถูกนำมาใช้แทนระบบเดิมโดยให้มีความยุ่งยากน้อยที่สุด (CONVERSION PLANS)

**การเตรียมตัวเป็นนักวิเคราะห์บทบาทของนักวิเคราะห์ระบบ**

หลังจากที่เราได้วิเคราะห์ว่า นักวิเคราะห์ระบบจะทำหน้าที่เป็นแกนกลางระหว่างนักธุรกิจ (BUSINESS PEOPLE) หรือผู้ใช้ระบบ (USERS) กับโปรแกรมเมอร์(PROGRAMMERS) อย่างไรก็ตามธุรกิจหรือหน่วยงานต่างๆ จึงมักจะมีความคิดที่ว่านักวิเคราะห์ระบบจะต้องมีพื้นฐานการเขียนโปรแกรมเป็นอันดับแรกแนวความคิดนี้ แท้จริงแล้วเป็นเพียงส่วนหนึ่งเท่านั้นในคุณสมบัติอันควรมีของนักวิเคราะห์ระบบ นักวิเคราะห์ระบบจะต้องมีความสามารถที่จะพัฒนาระบบเพื่อแก้ปัญหาให้กับผู้ใช้หรือธุรกิจอย่างมีเทคนิคและแบบแผน โปรแกรมเมอร์ที่เก่งมิได้หมายความว่าเขาจะเป็นนักวิเคราะห์ระบบที่ดีได้ในทางตรงกันข้าม โปรแกรมเมอร์ที่ไม่เก่งมิได้หมายความว่าเขาจะเป็นนักวิเคราะห์ระบบที่ดีไม่ได้ หากเราจะพิจารณาถึงคุณสมบัติพื้นฐานที่นักวิเคราะห์ระบบควรมี โดยยึดตามแนวทางของงานที่นักวิเคราะห์ระบบต้องใช้ปฏิบัติ ก็จะเป็นดังต่อไปนี้

1. ความรู้ทางด้านเทคนิคของระบบงานข้อมูลและเทคโนโลยี

นักวิเคราะห์ระบบมักจะถูกมองว่าเป็น "ตัวแทนแห่งความเปลี่ยนแปลง" เหตุที่ว่างงานของนักวิเคราะห์ระบบเป็นงานที่ก่อให้เกิดความเปลี่ยนแปลงของระบบงานโดยตรง นักวิเคราะห์ระบบจะต้องนำเอาความรู้ทางด้านเทคนิค และเทคโนโลยีมาช่วยให้เกิดประโยชน์ต่อระบบงานของผู้ใช้หรือธุรกิจนักวิเคราะห์ระบบจะต้องทำตัวเองให้ทันสมัยอยู่เสมอ และทราบถึงแนวโน้มที่จะเป็นไปในอนาคตด้วยการอ่านแม็กกาซีนรายเดือนหรือรายสัปดาห์ ในส่วนของระบบงานข้อมูลหรือคอมพิวเตอร์แม็กกาซีนจึง เป็นสิ่งที่นักวิเคราะห์ระบบสมควรต้องติดตามอย่างสม่ำเสมอ นอกจากนี้การสัมมนาทางด้านเทคโนโลยีใหม่ๆ ก็เป็นอีกสิ่งหนึ่งที่ควรติดตามเช่นกัน

2.ประสบการณ์ทางด้านการเขียนโปรแกรม

เป็นที่แน่นอนว่านักวิเคราะห์ระบบจะต้องทำงานใกล้ชิดกับโปรแกรมเมอร์ ความรู้ทางโปรแกรมจึงเป็นสิ่งสำคัญยิ่งที่จะช่วยให้การสื่อสาร ระหว่างนักวิเคราะห์ระบบกับโปรแกรมเมอร์ดำเนินไปได้โดยสะดวกและเข้าใจกัน ถึงแม้ว่านักวิเคราะห์ระบบอาจจะไม่ได้ไปนั่งเขียนโปรแกรมด้วยตนเองก็ตาม นักวิเคราะห์ระบบควรจะมีความรู้ทางด้าน HIGH-LEVEL PROGRAMMING LANGUABE อย่างน้อย 1 ภาษา เช่น FORTRAN, BASIC, COBLO,ADA หรือ ที่เราเรียกกันอย่างย่อๆ ว่า 4GL อย่างน้อย 1 ภาษาเช่น DBASE, FOXPRO,ORACLE,MAGIC, หรือ RBASE เป็นต้น ภาษาที่เลือกเพื่อนำไปใช้ในการพัฒนาระบบควรจะพิจารณาตามความเหมาะสม เช่น หากเป็นบริษัททีพัฒนาซอฟต์แวร์เพื่อขายโดยตรงก็อาจเลือกใช้ภาษา C เป็นต้น หรือหากระบบงานเกี่ยวข้องกับงานทางด้าน วิศวกรรมหรืองานค้นคว้าทางด้านวิทยาศาสตร์ ก็อาจเลือก FORDTRAN เป็นภาษาหลัก อย่างไรก็ตามการเลือกใช้ภาษามิได้จำกัดอยู่เฉพาะว่าภาษานั้นเหมาะสมกับงาน ชนิดใด หากแต่ควรคำนึงถึงว่าภาษาที่จะใช้นั้น เป็นภาษาที่สามารถนำไปพัฒนาระบบได้อย่างต่อเนื่อง และมีความสามารถที่จะรองรับงานนั้นได้จึงเป็นจุดสำคัญ

3. ความรู้ทั่วไปทางด้านธุรกิจ

คงเป็นไปไม่ได้ว่านักวิเคราะห์ระบบจะเป็นผู้เชี่ยวชาญไปซะทุกเรื่องทั้งทางด้านคอมพิวเตอร์และ ธุรกิจ แต่อย่างไรก็ดี เนื่องจากนักวิเคราะห์ระบบเป็นตัวกลางที่จะ เชื่อมต่อระหว่างการใช้ความสามารถของคอมพิวเตอร์ มาเป็นประโยชน์ต่อธุรกิจหรือผู้ใช้ ดังนั้น นักวิเคราะห์ระบบจึงควรจะมีความรู้ในเชิงธุรกิจบ้าง เช่น การตลาด การบัญชี ระบบงานสินค้าคงคลัง หรือธุรการงานบุคคลจากความรู้พื้นฐานดังกล่าว อาจได้มาเองในระหว่างการพัฒนาระบบ จากการสัมภาษณ์ หรือสอบถามจากผู้รู้หรือจากหนังสือ

4. ความสามารถในการแก้ปัญหาและวิธีแก้ปัญหา

นักวิเคราะห์ระบบจะต้องมีความสามารถที่จะตีปัญหาใหญ่ที่เกิดขึ้นแก่ธุรกิจเป็นส่วนๆ และวิเคราะห์ซึ่งปัญหาเหล่านั้น เพื่อที่จะหาวิธีการแก้ปัญหา นักวิเคราะห์ระบบจะต้องรู้จักวิเคราะห์ปัญหา ในแง่ของการหาเหตุและผล อย่างมีขั้นตอน และรู้จักที่จะใช้ความสามารถของตนเพื่อหาวิธีการต่างๆ เพื่อแก้ปัญหา (ALTERNATIVE SOLUTIONS) แม้ว่าความสามารถอันนี้จะเป็นพรสวรรค์ที่มีมา ในแต่ละคนซึ่งไม่เหมือนกัน อย่างไรก็ตาม ความสามารถที่จะแก้ปัญหาต่างๆ ก็สามารถที่จะพัฒนาและเรียนรู้ได้หัวใจสำคัญของการหาวิธีการแก้ปัญหานั่นก็คือ พยายามมองภาพของปัญหาให้กว้างอย่าคิดว่า วิธีการแก้ปัญหาวิธีแรกที่ตนคิดเป็นวิธีที่ดีที่สุดและเป็นวิธีเดียวเท่านั้น อย่าคิดว่าวิธีการแก้ปัญหาที่คนอื่นคิดเพื่อแก้ปัญหาที่คล้ายๆ กันกับของตนจะเป็นวิธีมาตรฐาน และใช้ได้กับกรณีของเรา เราควรจะพิจารณาจุดแข็งและจุดอ่อน (STRONG AND WEAK POINTS) ของแต่ละวิธีโดยละเอียดก่อนตัดสินในที่จะนำวิธีการนั้นมาพัฒนาเป็นระบบใช้จริง

5. มนุษย์สัมพันธ์และความสามารถในการติดต่อสื่อสาร

เนื่องจากนักวิเคราะห์ระบบจะต้องพบปะกับบุคคลหลายประเภท หลายอาชีพ และหลายระดับ เช่น ผู้อำนวยการ ผู้จัดการ ช่างเทคนิค โปรแกรมเมอร์ เลขานุการ และนักบัญชี การสื่อสารจึงเป็นสิ่งสำคัญอย่างยิ่งที่จะทำให้บุคคลต่างๆ ที่นักวิเคราะห์ระบบติดต่อผู้นั้นได้เข้าใจสิ่งที่นักวิเคราะห์ต้องการ มนุษย์สัมพันธ์และความสามารถในการสื่อสาร ในที่นี้หมายรวมถึงความสามารถที่จะสัมภาษณ์ (INTERVIEWING) ความสามารถที่จะอธิบายหรือชี้แจงในที่ประชุม (PRESENTATION) รวมทั้งความสามารถในการรับฟัง (LISTENING) ด้วย นอกจากนี้ ความสามารถที่จะทำงานเป็นกลุ่ม (GROUP WORK OR TEAM) ก็เป็นสิ่งหนึ่งที่นักวิเคราะห์จะขาดเสียไม่ได้ เนื่องจากงานของนักวิเคราะห์ส่วนใหญ่จะต้องกระจายให้กับโปรแกรมเมอร์ หรือถัดลงไปการทำงานเป็นกลุ่มหรือทีมจึงย่อมส่งผลต่อความสำเร็จและความเชื่อถือต่อนักวิเคราะห์ระบบเองโดยตรง นักวิเคราะห์ระบบควรเล็งเห็นความสำคัญของการทำงานเป็นกลุ่ม ไม่ใช่เฉพาะแต่กับฝ่ายของตนเอง หรือกับโปรแกรมเมอร์เท่านั้น หากแต่ต้องแฝงตัวเองเป็นสมาชิกในกลุ่มของผู้ใช้ระบบหรือธุรกิจที่ตนวางระบบให้อีกด้วย การทำเช่นนั้น หากแต่ต้องแฝงตัวเองเป็นสมาชิกในกลุ่มของผู้ใช้ระบบหรือธุรกิจที่ตนวางระบบให้อีกด้วย การทำเช่นนั้นจะทำให้ผู้ใช้ระบบรู้สึกเป็นนักวิเคราะห์และจะทำให้การติดตั้งระบบงานคอมพิวเตอร์เป็นไปโดยสะดวกขึ้น พร้อมกับลดแรงกดดันหรือต่อต้านจากผู้ใช้ระบบที่มีแนวความคิดว่า ตนโดนยัดเยียดระบบงานใหม่ให้แทนระบบงานแบบดั้งเดิม นักวิเคราะห์ระบบจะต้องทำตัวเองให้ทันสมัยอยู่เสมอ และทราบถึงแนวโน้มที่จะเป็นไปในอนาคตด้วยการอ่านแม็กกา-ซีนรายเดือนหรือรายสัปดาห์ ในส่วนของระบบงานข้อมูลหรือคอมพิวเตอร์แม็กกาซีน จึงเป็นสิ่งที่นักวิเคราะห์ระบบสมควรต้องติดตามอย่างสม่ำเสมอ นอกจากนี้การสัมมนาทางด้านเทคโนโลยีใหม่ๆ ก็เป็นอีกสิ่งหนึ่งที่ควรติดตามเช่นกัน

**บทนำการวิเคราะห์และออกแบบระบบ**

Introduction to Systems Analysis and Design

อธิบายถึงบทบาทของเทคโนโลยีสารสนเทศในแวดวงธุรกิจยุคปัจจุบัน การเรียนรู้เกี่ยวกับพัฒนาการของระบบข้อมูลสารสนเทศ หลักการวิเคราะห์และออกแบบระบบ วงจรการพัฒนาระบบ กระบวนการ เครื่องมือและเทคนิควิธีในการพัฒนาระบบ พร้อมทั้งบทบาทของแผนกเทคโนโลยีสารสนเทศและหน้าที่ของบุคลากรในแผนก

**เทคโนโลยีสารสนเทศ**

เทคโนโลยีสารสนเทศ (Information Technology : IT) เป็นการผสมผสานระหว่างฮาร์ดแวร์ (Hardware) ซอฟต์แวร์ (Software) และระบบสื่อสารโทรคมนาคม (Telecommunication Systems) ที่ช่วยในการปฏิบัติงานเพื่อเพิ่มประสิทธิภาพให้กับงานหรือผลผลิต ตลอดจนช่วยให้ข้อมูลเพื่อประกอบการตัดสินใจแก่ผู้บริหาร

นักวิเคราะห์ระบบ (Systems Analyst : SA) เป็นบุคคลที่มีความสำคัญมากในแผนกเทคโนโลยีสารสนเทศ ที่ต้องทำหน้าที่ในการวางแผน วิเคราะห์และติดตั้งระบบ

ความสำเร็จของธุรกิจส่วนหนึ่งจะขึ้นอยู่กับเทคโนโลยีสารสนเทศ ทั้งนี้จากการสำรวจของบริษัทไอบีเอ็ม จากองค์กรต่างๆ กว่า 3,000 แห่ง พบว่าองค์กรส่วนใหญ่ที่ใช้เทคโนโลยี ช่วยให้งานด้านการบริการลูกค้าดีขึ้น สามารถรองรับการเปลี่ยนแปลงได้อย่างรวดเร็ว ตลอดจนเพิ่มประสิทธิภาพในการทำงานของบุคลากรในองค์กร และได้แสดงให้เห็นว่าระบบข้อมูลสารสนเทศช่วยให้รู้ทิศทางความคาดหวังของลูกค้าและความต้องการของธุรกิจ

**แบบจำลองกระบวนการทางธุรกิจ** (Business Process Modeling) แสดงให้เห็นรายละเอียดของการดำเนินการธุรกิจและข้อมูลสารสนเทศที่เป็นประโยชน์ ซึ่งแบ่งออกได้ดังนี้

- **ลักษณะเค้าโครงธุรกิจ** (Business Profile) เป็นข้อมูลโดยภาพรวมขององค์กร กล่าวถึงการดำเนินธุรกิจทั้งหมด
- **แบบจำลองทางธุรกิจ** (Business Model) เป็นการแสดงภาพรายละเอียดของกระบวนการในการดำเนินธุรกิจ
- **กระบวนการทางธุรกิจ** (Business Process) เป็นการอธิบายลักษณะงานแต่ละงานและผลของงานนั้นๆ

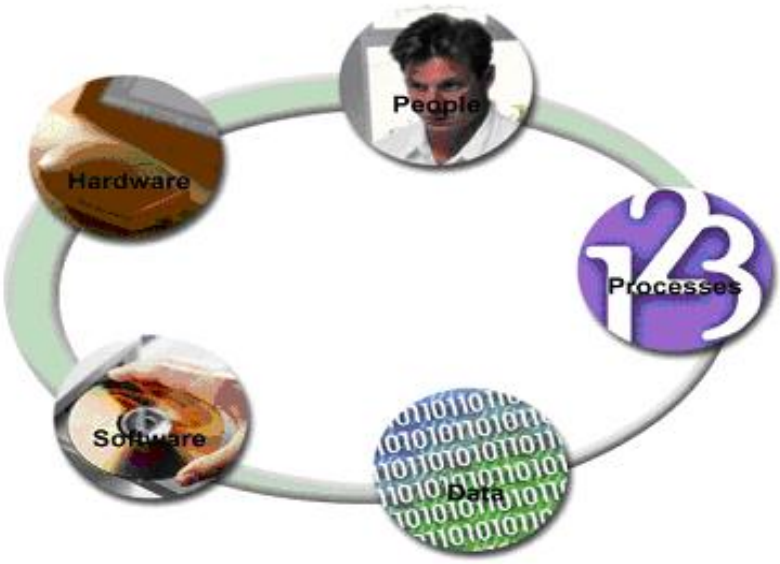
รูปที่ 1 กระบวนการทางธุรกิจที่แสดงถึงเหตุการณ์ที่ประกอบด้วย 3 กระบวนการย่อยและผลลัพธ์ที่ได้

**ส่วนประกอบของระบบสารสนเทศ**

ระบบ (System) หมายถึง ชุดขององค์ประกอบที่สัมพันธ์กันและก่อให้เกิดผลลัพธ์โดยเฉพาะออกมา ปัจจุบันระบบสารสนเทศ (Information System) จึงมีบทบาทอย่างมากต่อการดำเนินการของธุรกิจหรือที่เรียกว่า ระบบความจำเป็นยิ่งยวด (Mission-critical System)

**สารสนเทศ** เป็นข้อมูลที่ถูกปรับเปลี่ยนเป็นข้อมูลที่เกิดประโยชน์ ซึ่งขั้นตอนของการปรับเปลี่ยนข้อมูล (Data) ให้มาเป็นข้อมูลที่เกิดประโยชน์หรือสารสนเทศ (Information) เรียกว่า กระบวนการ (Processing)

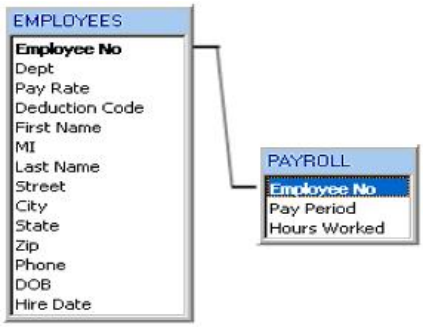
ระบบสารสนเทศประกอบด้วย 5 ปัจจัยสำคัญ คือ ฮาร์ดแวร์ ซอฟต์แวร์ ข้อมูล การประมวลผล และบุคลากร ดังแสดงในรูปที่ 1-5




รูปที่ 2 องค์ประกอบของระบบสารสนเทศ

1. **ฮาร์ดแวร์** สิ่งที่จับต้องได้ในระบบสารสนเทศ หมายรวมถึง คอมพิวเตอร์ เครือข่าย สแกนเนอร์ อุปกรณ์ดิจิตอลในการจับภาพ หรือสิ่งประดิษฐ์ด้านเทคโนโลยีอื่นๆ
2. **ซอฟต์แวร์** รายละเอียดของชุดคำสั่งที่ควบคุมให้คอมพิวเตอร์ทำงานตามคำสั่ง แบ่งออกเป็น 2 ประเภท
   - ซอฟต์แวร์ระบบ (System Software) เป็นรายละเอียดของชุดคำสั่งที่ควบคุมคอมพิวเตอร์และระบบการปฏิบัติงานของคอมพิวเตอร์ การเชื่อมต่อกับฮาร์ดแวร์เพื่อควบคุมภาระงาน เช่น การโอนถ่ายข้อมูลที่อยู่ในรูปแบบที่แตกต่างกัน การป้องกันการโจมตีของไวรัส การจัดการสร้างฐานข้อมูลสำรอง เป็นต้น สำหรับในระบบเครือข่ายจะมีซอฟต์แวร์ระบบต่างหากที่เรียกว่า ระบบปฏิบัติงานเครือข่าย (Network Operation System : NOS) เพื่อช่วยควบคุมการส่งผ่านของข้อมูล ช่วยป้องกันความปลอดภัยและบริหารจัดการผู้ใช้งาน ซึ่งซอฟต์แวร์ระบบดังกล่าว มักจะถูกจัดเตรียมมาให้จากโรงงานผู้ผลิตหรือจากบริษัทผู้จัดจำหน่าย หรืออนุญาตให้ผู้ใช้สามารถเรียกจากระบบเครือข่ายขององค์กรหรือจากอินเทอร์เน็ต
   - ซอฟต์แวร์ประยุกต์ (Application Software) ประกอบด้วยรายละเอียดของชุดคำสั่งที่ช่วยสนับสนุนผู้ใช้และองค์กรให้สามารถดำเนินงานได้ตามความต้องการ โดยช่วยให้ผู้ใช้สามารถเพิ่มประสิทธิภาพของงานได้มากขึ้น ตัวอย่างของซอฟต์แวร์ประยุกต์ ได้แก่ แผ่นทำการหรือสเพร็ดชีด (Spreadsheet) โปรแกรมประมวลผลคำหรือเวิร์ดโพรเซสเซอร์ (Word Processors) หรือระบบจัดการฐานข้อมูล สำหรับบริษัทใหญ่อาจมีหลายๆ ระบบงาน เช่น ระบบเงินเดือน ระบบการสั่งสินค้า ระบบบัญชีมาช่วยในการทำงาน ถ้าแผนกไอทีพัฒนาขึ้นเอง จะเรียก โปรแกรมประยุกต์ภายในหรืออินเฮ้าส์แอพพลิเคชัน (In-house Application) หากซื้อซอฟต์แวร์สำเร็จ (Software Package) เช่น ระบบควบคุมสินค้า ระบบเงินเดือน ที่สามารถประยุกต์ใช้งานร่วมกันได้ จัดเป็นระบบงานแนวราบหรือฮอร์ริซอนเทลซิสเต็ม (Horizontal System) แต่หากว่าจ้างพัฒนาออกแบบเฉพาะให้ตรงตามความต้องการของธุรกิจ จัดเป็นระบบงานแนวดิ่งหรือเวอร์ทิเคลซิสเต็ม (Vertical System) เช่น ระบบบริการเช่าวิดีโอหรือระบบขายสินค้าทางอินเทอร์เน็ต

     บริษัทส่วนมาก มักนิยมที่จะผสมผสานระหว่างซอฟต์แวร์ที่พัฒนาขึ้นเองกับซอฟต์แวร์สำเร็จที่มีจำหน่ายอยู่ทั่วไป ดังนั้น ในการวางแผนจัดทำระบบสารสนเทศ จะต้องคำนึงถึงการต่อเชื่อมระหว่างระบบงานใหม่กับระบบงานเก่า (Legacy Systems) เช่น การจัดทำระบบงานบุคลากรขึ้นใหม่ ต้องคำนึงถึงการโอนถ่ายข้อมูลกับระบบเงินเดือนเดิมที่ยังใช้งานอยู่ ว่าทั้งสองระบบสามารถส่งผ่านข้อมูลกันได้อย่างถูกต้องเที่ยงตรง
3. **ข้อมูล** (Data) ระบบสารสนเทศเป็นการนำข้อมูลดิบมาทำให้เกิดประโยชน์ ดังเช่นรูปที่ 1-6 จากข้อมูลของพนักงานแต่ละราย ซึ่งมีอัตราค่าจ้างกับรายการเงินหักจัดเก็บอยู่ในแฟ้มบุคลากร และจำนวนชั่วโมงปฏิบัติงานกับงวดของการจ่ายเงินจัดเก็บอยู่ในแฟ้มข้อมูลการจ่ายเงิน ระบบสารสนเทศจะใช้ข้อมูลจากทั้งสองแฟ้มคำนวณหายอดค่าจ้างรวม เพื่อจัดทำเช็คจ่ายด้วยยอดเงินสุทธิที่หักภาษีและรายการเงินหักของพนักงานแล้ว




รูปที่ 3 แสดงระบบสารสนเทศที่ใช้ 2 แฟ้มข้อมูลในการจัดทำเช็คจ่ายเงิน

4. **กระบวนการหรือการประมวลผล** (Processes หรือ Procedures) อธิบายถึงวิธีการดำเนินงานตามแบบจำลองทางธุรกิจ ซึ่งอาจเขียนอธิบายอยู่ในรูปของเอกสารคู่มือหรือเอกสารอ้างอิงในลักษณะออนไลน์ก็ได้
5. **บุคลากร** (People) จุดมุ่งหมายหลักของการจัดทำระบบสารสนเทศ เพื่อการนำข้อมูลสารสนเทศมาก่อประโยชน์แก่ผู้บริหารและผู้ใช้งานทั้งภายในและภายนอกองค์กร ผู้ใช้งาน (Users หรือ End Users) หมายรวมถึง พนักงาน ลูกค้า บริษัทผู้จัดจำหน่ายและผู้ที่มีส่วนเกี่ยวข้องกับระบบสารสนเทศ โดยแบ่งออกเป็นผู้ใช้งานภายใน (Internal Users) ได้แก่ ผู้จัดการ ช่างผู้เชี่ยวชาญ พนักงานขายและเจ้าหน้าที่ ส่วนผู้ใช้งานภายนอก (External Users) เช่น บรรดาลูกค้าที่สั่งสินค้าเข้ามาทางเว็บไซต์ และบริษัทผู้จัดส่งวัตถุดิบที่ใช้ระบบงานบริหารลูกค้าในการวางแผนจัดสายงานผลิต ทั้งนี้ ระบบจะประสบความสำเร็จหรือความล้มเหลวขึ้นอยู่กับความพึงพอใจของผู้ใช้งานในข้อมูลที่ได้รับจากระบบงานเป็นสำคัญ

**การจำแนกประเภทของธุรกิจ**

แต่เดิมมีการจำแนกประเภทของบริษัทออกเป็น ประเภทขายผลิตภัณฑ์ (Production-oriented) และประเภทขายบริการ (Service-oriented) ปัจจุบันมีธุรกิจอีกประเภทหนึ่ง คือ กิจการที่ต้องพึ่งพาอินเทอร์เน็ต (Internet-dependent Firm) หรือที่เรียกว่า ธุรกิจดอทคอม (.com) ซึ่งแตกต่างจากบริษัทประเภทดั้งเดิมที่เรียกว่า ก่ออิฐถือปูน (Brick-and-mortar) เพราะไม่ต้องมีการลงทุนจัดตั้งห้างร้านแสดงสินค้า และในช่วงปีที่ผ่านมาบริษัทบนอินเทอร์เน็ตมีอัตราการเติบโตและมียอดราคาหุ้นที่พุ่งสูงขึ้นเป็นอย่างมาก เช่น amazon.com หรือ e-trade.com และหนึ่งในการเติบโตที่เคียงคู่ไปกับธุรกิจประเภทใหม่นี้ คือ พาณิชย์อิเล็กทรอนิกส์ (Electronic Commerce : e-commerce หรือ Internet Commerce : I-Commerce) ทั้งในลักษณะของผู้ขายกับผู้ซื้อ (Business-to-Consumer : B2C) และผู้ขายกับผู้ขาย (Business-to-Business : B2B) เมื่อเกิดความต้องการระบบพาณิชย์อิเล็กทรอนิกส์มากขึ้น ย่อมมีความต้องการผู้พัฒนาระบบหรือนักวิเคราะห์ระบบมากขึ้นเป็นลำดับ

**สิ่งที่ควรคำนึงถึงในการวิเคราะห์ระบบ**

ในการวิเคราะห์ระบบนั้น นอกจากนักวิเคราะห์ระบบต้องศึกษาระบบสารสนเทศขององค์กร โดยศึกษาการดำเนินธุรกิจและสิ่งที่ช่วยในการดำเนินธุรกิจอย่างละเอียดแล้ว ควรพยายามให้ครอบคลุมคำถามเหล่านี้ด้วย ได้แก่

1. ระบบงานนี้ มีผลกระทบกับระบบอื่นหรือไม่
2. ระบบงานนี้ มีขอบเขตของระบบขนาดไหน
3. ระบบงานนี้ มีความต้องการเฉพาะพิเศษอื่นอย่างไร
4. ขนาดและความเติบโตของธุรกิจในอนาคตเป็นอย่างไร

**ชนิดของระบบงานสารสนเทศทางธุรกิจ**

จากการที่แต่ละบริษัทมีความต้องการระบบข้อมูลสารสนเทศมากมายที่แตกต่างกัน ซึ่งในอดีตได้แบ่งชนิดของระบบงานสารสนเทศตามความต้องการของผู้ใช้เป็นหลัก เช่น ระบบสำนักงาน (Office Systems) ใช้งานโดยเจ้าหน้าที่ธุรการ ระบบปฏิบัติการ (Operating Systems) ใช้งานโดยเจ้าหน้าที่ป้อนข้อมูล ระบบข้อมูลสารสนเทศเพื่อการจัดการ (Management Information Systems) ใช้งานโดยผู้บริหารระดับกลางและระดับล่าง ระบบข้อมูลเพื่อผู้บริหาร (Executive Information Systems) ใช้งานโดยผู้บริหารระดับสูง บางทฤษฎีแบ่งเป็น ระบบช่วยการตัดสินใจ (Decision Support

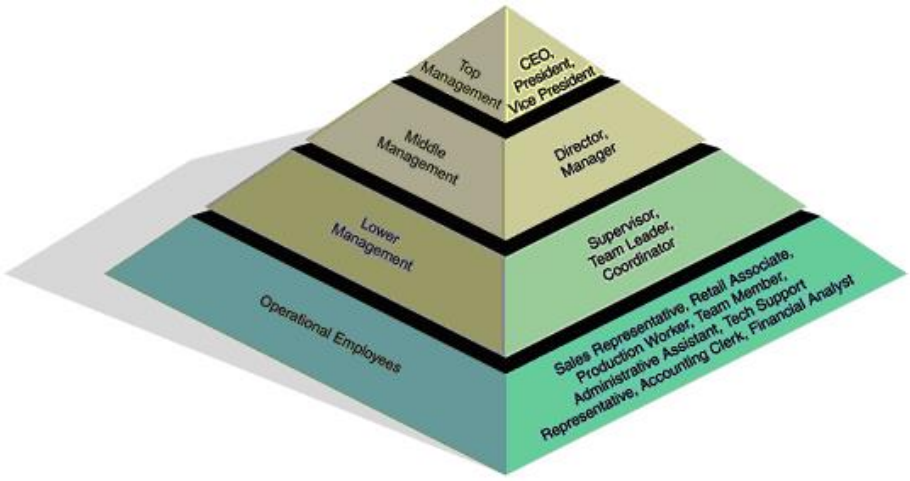
Systems) ใช้งานโดยผู้วางแผนตัดสินใจเบื้องต้น และระบบผู้เชี่ยวชาญ (Expert Systems) ใช้งานโดยผู้ที่ทำหน้าที่ควบคุมหรือผู้ทำหน้าที่วินิจฉัยแก้ไขปัญหา

ซึ่งมีความไม่สอดคล้องกับลักษณะการใช้งานในปัจจุบัน ดังนั้น จึงมีการแบ่งชนิดระบบงานขึ้นใหม่ ตามลักษณะและหน้าที่ของงานเป็นหลัก จำแนกออกได้ดังนี้

- ระบบงานคำนวณขนาดใหญ่ (Enterprise Computing Systems) ที่ต้องการระบบสารสนเทศเพื่อนำมาใช้ในการบริหารข้อมูลขนาดใหญ่ เช่น ระบบการสำรองที่นั่งจากทั่วทุกมุมโลกของสายการบิน ระบบบัตรเครดิตที่รองรับจำนวนผู้ใช้บัตรจำนวนมหาศาล
- ระบบข้อมูลเพื่อการประมวลผล (Transaction Processing Systems : TP) หรือระบบประมวลผลแบบทันที (Online Transaction Processing : OLTP) จัดเป็นระบบงานปฏิบัติการ เนื่องจากเป็นงานที่ต้องทำประจำในแต่ละวัน เช่น ระบบการเรียกเก็บเงิน
- ระบบข้อมูลเพื่อสนับสนุนธุรกิจ (Business Support Systems : BSS) เป็นระบบที่ช่วยสร้างความสัมพันธ์เพื่อช่วยในการปฏิบัติงานประจำของทุกหน่วยงานในองค์กร โดยช่วยวิเคราะห์ข้อมูล สร้างสารสนเทศที่จำเป็น ควบคุมและจัดการกระบวนการปฏิบัติงานและสร้างข้อมูลช่วยประกอบการตัดสินใจ
- ระบบองค์ความรู้เพื่อการจัดการ (Knowledge Management Systems) บางตำราเรียกระบบผู้เชี่ยวชาญ เพราะเป็นการจำลองความเป็นเหตุเป็นผลของความนึกคิดของมนุษย์เข้ากับฐานองค์ความรู้กับกฎเกณฑ์เงื่อนไข โดยจัดเก็บองค์ความรู้ไว้ในฐานข้อมูลขนาดใหญ่
- ระบบเพิ่มประสิทธิภาพผู้ใช้งาน (User Productivity Systems) เป็นเครื่องมือสำคัญที่ช่วยเพิ่มหรือปรับปรุงคุณภาพในการปฏิบัติงานให้กับพนักงานภายในองค์กรทุกระดับ เช่น ระบบเครือข่ายภายในองค์กรและในระยะไกล จดหมายอิเล็กทรอนิกส์ โทรสาร การประชุมทางไกล การนำเสนอด้วยภาพ อินทราเน็ตและอินเทอร์เน็ต
- การรวมระบบสารสนเทศ บริษัทขนาดใหญ่ส่วนมากต้องการระบบที่มีลักษณะของระบบงานคำนวณขนาดใหญ่ ระบบข้อมูลเพื่อการประมวลผล ระบบข้อมูลเพื่อสนับสนุนธุรกิจ ระบบองค์ความรู้เพื่อการจัดการ และระบบเพิ่มประสิทธิภาพผู้ใช้งาน รวมอยู่ด้วยกัน

**โครงสร้างผังองค์กร**

ในช่วงปี 2-3 ปีที่ผ่านมาบริษัทต่างๆ มีการลดขนาดและปรับกระบวนการในการดำเนินธุรกิจเป็นจำนวนมาก อย่างไรก็ตาม โครงสร้างผังองค์กรโดยทั่วไป ยังคงเป็นตามรูปที่ 1-9




รูปที่ 4 รูปแบบขององค์กรโดยทั่วๆ ไป

- ผู้บริหารระดับสูง ทำหน้าที่กำหนดแผนระยะยาวที่เรียก แผนกลยุทธ์ เพื่อให้บริษัทดำเนินพันธกิจให้ไปถึงเป้าหมายที่กำหนด รวมการกำหนดแผนพัฒนาเทคโนโลยีในระยะยาว

- ผู้บริหารระดับกลาง ทำหน้าที่ในการดำเนินการบริหารงานตามนโยบายผู้บริหารระดับสูง ให้บรรลุวัตถุประสงค์ในช่วงสั้นๆ โดยทั่วไปจะอยู่ในระยะ 1 เดือนถึง 1 ปี ที่เรียก แผนยุทธวิธี ผู้บริหารระดับกลางจะควบคุมดูแลผู้ที่รับผิดชอบรองลงไป เพื่อช่วยกำหนดทิศทาง ให้การสนับสนุนสิ่งที่จำเป็นในการปฏิบัติงานและติดตามให้กิจกรรมเสร็จสมบูรณ์
- ผู้บริหารระดับล่าง ได้แก่ หัวหน้างาน หรือผู้ควบคุม ทำหน้าที่ตรวจตราดูแลพนักงานระดับปฏิบัติงานให้สามารถปฏิบัติงานในแต่ละวันได้อย่างสมบูรณ์
- ระดับผู้ปฏิบัติงาน หน้าที่หลักเป็นการใช้ระบบการประมวลผลรายการในการป้อนและรับข้อมูลเพื่อการปฏิบัติงานประจำ

**เทคนิควิธีและเครื่องมือในการพัฒนาระบบ**

นักวิเคราะห์ระบบ นอกจากต้องทำความเข้าใจการดำเนินงานทางธุรกิจแล้ว ต้องรู้จักการใช้เทคนิควิธีหลายแบบ ได้แก่ การสร้างแบบจำลอง การสร้างต้นแบบและการใช้เครื่องมือช่วยในการออกแบบ เพื่อการวางแผนออกแบบและติดตั้งระบบงานสารสนเทศ อีกทั้งต้องทำงานได้ในสภาพแวดล้อมเป็นทีม เพื่อรับข้อมูลทั้งจากผู้ใช้งาน ผู้จัดการ และผู้ปฏิบัติงานด้านไอที นำมาออกแบบให้สอดคล้องกับความต้องการของทุกคน และแสดงเป็นภาพของงานออกแบบให้เห็นได้อย่างชัดเจน

**แบบจำลอง**

การสร้างแบบจำลอง (Modeling) เป็นการนำเสนอแนวความคิดหรือกระบวนการในรูปแบบของภาพตามที่ได้ทำการวิเคราะห์และออกแบบไว้ เพื่อให้ง่ายต่อการทดสอบ การแก้ไข และสามารถให้คำอธิบายระบบสารสนเทศได้ชัดเจนขึ้น

แบบจำลองทางธุรกิจ (Business Model) หรือแบบจำลองความต้องการ (Requirements Model) จะแสดงถึงการทำงานของธุรกิจที่ระบบสารสนเทศต้องให้การสนับสนุน แบบจำลองข้อมูล (Data Model) จะอธิบายถึงโครงสร้างข้อมูลและการออกแบบ แบบจำลองแบบเชิงวัตถุ (Object Model) จะอธิบายถึงวัตถุซึ่งเป็นที่รวมของข้อมูลและการประมวลผล แบบจำลองเครือข่าย (Network Model) จะแสดงภาพของการออกแบบและโปรโตคอลที่เชื่อมโยงในการสื่อสารทางไกล แบบจำลองการประมวล (Process Model) จะอธิบายถึงตรรกะและการประมวลผลของระบบสำหรับนักเขียนโปรแกรม (Programmers) เพื่อนำไปใช้ในการพัฒนาโปรแกรมต่อไป

**ต้นแบบ**

การสร้างต้นแบบ (Prototype) เกี่ยวข้องกับการสร้างงานในเบื้องต้นของระบบสารสนเทศและองค์ประกอบที่เกี่ยวข้อง หากเปรียบกับโรงงานผลิตเครื่องบิน ต้องทำการทดสอบชิ้นงานที่ออกแบบมาใหม่โดยใช้ลมในช่องอุโมงค์ก่อน ต้นแบบจะช่วยทดสอบแนวคิดของระบบงาน และเป็นการตรวจสอบ อินพุท (Input) เอ้าท์พุท (Output) และส่วนต่อประสานผู้ใช้ (User Interfaces) ก่อนที่จะตัดสินใจผลิตในขั้นสุดท้าย โดยต้นแบบสามารถนำไปพัฒนาเป็นระบบงานได้จริง ซึ่งจะเห็นว่าช่วยให้การพัฒนาระบบงานเป็นไปได้อย่างรวดเร็ว

**วิศวกรรมซอฟต์แวร์ช่วยการพัฒนาระบบ**

วิศวกรรมซอฟต์แวร์ช่วยการพัฒนาระบบ (Computer-Aided Systems Engineering- CASE) เป็นเทคนิควิธีที่ใช้โปรแกรมที่มีความสามารถสูงเป็นเครื่องมือ เรียกย่อๆ ว่า เคสทูล (CASE Tools) เพื่อช่วยนักวิเคราะห์ระบบพัฒนาและบำรุงรักษาระบบสารสนเทศ โดยมองเห็นกรอบของการพัฒนาระบบงานทั้งหมด

**ความร่วมมือในการพัฒนาระบบ**

ในอดีต แผนกไอทีจะเป็นหน่วยงานที่คิดพัฒนาระบบสารสนเทศ โดยติดต่อประสานงานกับผู้ใช้งาน ซึ่งวิธีการดังกล่าวทำให้เกิดช่องว่างระหว่างผู้พัฒนาระบบงานและผู้ใช้งานอยู่เสมอ ต่อมาหลาย ๆ บริษัทได้ค้นพบว่า การสร้างทีมงานพัฒนาระบบงานอันประกอบด้วยบุคลากรด้านไอที ผู้ใช้งานและผู้จัดการ สามารถจะทำให้งานเสร็จสมบูรณ์ได้เร็วกว่าและมีผลงานที่ดีกว่า โดยมี 2 ระเบียบวิธีที่ได้รับความนิยม คือ Joint Application Development (JAD) และ Rapid Application Development (RAD)

**เครื่องมืออื่น ๆ ที่ใช้พัฒนาระบบ**

นอกจากเคสทูลแล้ว นักวิเคราะห์ระบบสามารถใช้เครื่องมือที่มีประสิทธิภาพอื่นในการจัดการและจัดทำโครงสร้างของงานพัฒนาระบบสารสนเทศ เช่น เวิร์ดโพรเซสเซอร์ สเพร็ดชีด และที่ได้รับความนิยมมาก คือ วิซิโอ (VISIO) ของ

ไมโครซอฟท์ สามารถใช้ช่วยในการวาดแผนภูมิต่างๆ

## ภาพรวมของระเบียบวิธีในการพัฒนาระบบ

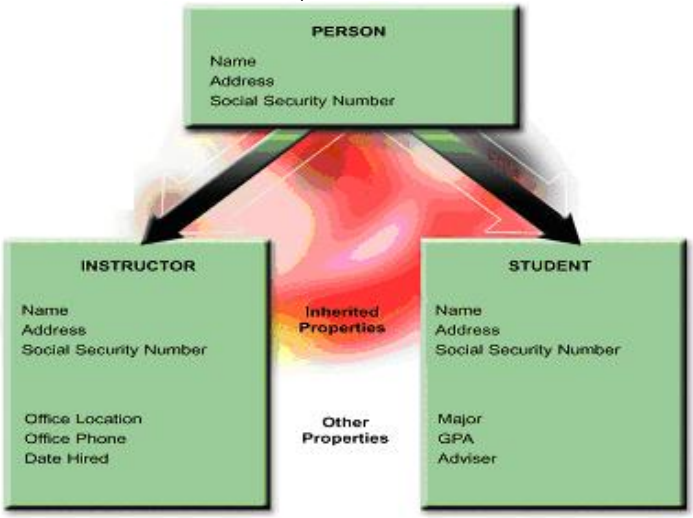
ระเบียบวิธีแบบเดิม คือ การวิเคราะห์ระบบเชิงโครงสร้าง แต่ระเบียบวิธีใหม่ที่นิยมกันอย่างกว้างขวาง คือ การวิเคราะห์ระบบเชิงวัตถุ

### การวิเคราะห์เชิงโครงสร้าง

การวิเคราะห์เชิงโครงสร้าง (Structured Analysis) เป็นเทคนิควิธีที่ใช้ในการพัฒนาระบบที่ง่ายและมีการใช้งานกันมาตั้งแต่เมื่อปี ค.ศ. 1960 ซึ่งรวมการจัดการข้อมูล โครงสร้างข้อมูล การออกแบบฐานข้อมูล และการออกแบบส่วนต่อประสานผู้ใช้ โดยแบ่งออกเป็นหลายระยะที่เรียกว่า วงจรการพัฒนาระบบ

### การวิเคราะห์เชิงวัตถุ

การวิเคราะห์เชิงวัตถุ (Object-Oriented Analysis : O-O) เป็นการรวมข้อมูลและกระบวนการเข้าด้วยกัน โดยเรียกเป็น วัตถุหรือออบเจ็กต์ (Object) เพื่อจำลองสภาพที่แท้จริงของกระบวนการและการปฏิบัติงานของธุรกิจ ผลที่ได้คือซอฟต์แวร์เชิงวัตถุชุดหนึ่งที่เป็นตัวแทนของคน สิ่งของ รายการหรือเหตุการณ์ ซึ่งนักเขียนโปรแกรมสามารถใช้ภาษาโปรแกรมเชิงวัตถุ (O-O Programming Language) แปลงออบเจ็กต์ที่ได้เป็นรหัสที่นำกลับมาใช้ได้อีก ส่วนประกอบของออบเจ็กต์จะจัดเป็นสมาชิกของคลาส (Class) ที่รวบรวมวัตถุที่เหมือนๆ กัน โดยมีคุณสมบัติที่เรียกว่า ลักษณะประจำหรือแอททริบิวท์ (Attributes) ที่ได้รับการสืบทอดมาจากคลาสหรือเป็นคุณสมบัติส่วนตัวของตนเอง ดังได้แสดงในรูปที่ 1-13 มีคลาสชื่อ PERSON ซึ่งมี INSTRUCTOR และ STUDENT เป็นสมาชิก ด้วยเหตุที่ PERSON มีแอททริบิวท์ชื่อ Address วัตถุที่ชื่อ STUDENT จะได้รับการสืบทอดแอททริบิวท์ Address มากจาก PERSON นอกจากนั้น STUDENT ยังมีแอททริบิวท์ชื่อ Major ที่ไม่ได้ใช้ร่วมกับสมาชิกอื่นๆ ในคลาสของ PERSON




รูปที่ 5 คลาสชื่อ PERSON ประกอบด้วย INSTRUCTOR และ STUDENT ออบเจ็กต์ทั้งสองจะมีคุณสมบัติสืบทอดและคุณสมบัติอื่นๆ

ในการสร้างโปรแกรม O-O จะเรียกการประมวลผลว่า เมทธอด (Methods) ซึ่งสามารถเปลี่ยนแอททริบิวท์ของออบเจ็กต์ได้ ตัวอย่างเช่น เรือใบเป็นออบเจ็กต์ มีความเร็วที่แล่นเป็นแอททริบิวท์ ซึ่งจะเปลี่ยนไปโดยเมทธอดให้ยกใบเรือขึ้นหรือลดใบเรือลง ออบเจ็กต์หนึ่งสามารถส่งเมซเสจ (Message) ให้อีกวัตถุหนึ่งได้สามารถขอการกระทำที่จำเพาะหรือข่าวสารจากผู้รับได้ เช่น เมื่อไม่มีแรงลม เจ้าของเรือใบอาจส่งสารข้อความว่า “ให้ติดเครื่องยนต์” ส่งไปยังตัวเรือใบ เป็นต้น

## วงจรการพัฒนาระบบ

เทคนิคของการวิเคราะห์ระบบเชิงโครงสร้าง เรียกว่า วงจรการพัฒนาระบบ (Systems Development Life Cycle : SDLC) ทั้งนี้เพื่อเตรียมการวางแผนและจัดกระบวนการในการพัฒนาระบบอย่างมีขั้นตอน โดยแบ่งออกเป็น 5 ระยะ ดังนี้

1. การวางแผนระบบ
2. การวิเคราะห์ระบบ

3. การออกแบบระบบ
4. การทำให้เกิดระบบ
5. การปฏิบัติงานและสนับสนุนระบบ

**การวางแผนระบบ**

การวางแผนระบบ (Systems Planning) มักถูกกำหนดความต้องการมาจากแผนกไอที ที่เรียกว่า ความต้องการระบบ (System Request) เป็นส่วนใหญ่ ซึ่งจะอธิบายถึงปัญหาหรือความต้องการในการเปลี่ยนแปลงระบบสารสนเทศหรือวิธีการประมวลผลทางธุรกิจ จุดมุ่งหมายคือ การกำหนดคุณสมบัติและขอบเขตของโอกาสทางธุรกิจหรือปัญหาอย่างชัดเจน โดยการสำรวจเบื้องต้น หรืออาจเรียกว่า การศึกษาความเป็นไปได้ ซึ่งเป็นขั้นตอนที่สำคัญ เพราะจะมีผลกระทบต่อเนื่องกับกระบวนการพัฒนาระบบต่อไปทั้งหมด

**การวิเคราะห์ระบบ**

การวิเคราะห์ระบบ (Systems Analysis) มีจุดมุ่งหมาย คือ ความเข้าใจความต้องการธุรกิจและการสร้างแบบจำลองเชิงตรรกะของระบบใหม่ ขั้นแรกคือ การกำหนดรูปแบบความต้องการ ให้คำจำกัดความและบรรยายถึงการประมวลผลธุรกิจ การกำหนดรูปแบบความต้องการจะเกี่ยวเนื่องกับการสังเกตการณ์ในระยะของการวางแผนระบบและเกี่ยวข้องกับเทคนิคในการค้นหาความจริง ภารกิจถัดไป คือ การสร้างแบบจำลองข้อมูล แบบจำลองการประมวลผล และแบบจำลองวัตถุ เพื่อพัฒนาจัดทำแบบจำลองทางตรรกะของกระบวนการทางธุรกิจ

**การออกแบบระบบ**

จุดมุ่งหมายของการออกแบบระบบ (System Design) คือ การสร้างแบบพิมพ์เขียวของระบบใหม่ตามความต้องการในเอกสารความต้องการระบบ กำหนดสิ่งที่จำเป็น เช่น อินพุท เอ้าท์พุท ส่วนต่อประสานผู้ใช้ และการประมวลผล เพื่อประกันความน่าเชื่อถือ ความถูกต้องแม่นยำ การบำรุงรักษาได้ และความปลอดภัยของระบบ

**การทำให้เกิดผล**

ในระยะของการทำให้เกิดผล (System Implement) ระบบงานใหม่จะถูกสร้างขึ้น ไม่ว่าผู้พัฒนาจะใช้การวิเคราะห์เชิงโครงสร้างหรือเชิงวัตถุก็ตาม ขั้นตอนจะเหมือนคือ การเขียนโปรแกรม การทำการทดสอบ การจัดทำเอกสารและการนำระบบลงติดตั้งเพื่อใช้งานจริง หากซื้อโปรแกรมสำเร็จรูป นักวิเคราะห์ระบบจะต้องเตรียมการเพื่อดัดแปลงในสิ่งที่จำเป็นและพิจารณาโครงแบบ (Configuration) ที่ต่างกัน วัตถุประสงค์ คือ การส่งมอบระบบงานสารสนเทศที่สามารถปฏิบัติงานได้อย่างสมบูรณ์พร้อมเอกสารระบบงาน รวมการโอนถ่ายข้อมูลเข้าแฟ้มข้อมูลของระบบใหม่ การจัดการฝึกอบรมผู้ใช้ และการปฏิบัติการในช่วงต่อของการเปลี่ยนแปลงระบบเก่ากับระบบใหม่ รวมถึงขั้นการประเมินผล ที่เรียกว่า การประเมินผลระบบ (System Evaluation) เพื่อตัดสินระบบอย่างเหมาะสมและเพื่อคาดการณ์เกี่ยวกับค่าใช้จ่ายและผลประโยชน์ที่จะได้รับ

**การปฏิบัติงานและสนับสนุนระบบ**

ในช่วงการปฏิบัติงานและสนับสนุนระบบ (Systems Operation and Support) บุคลากรด้านไอทีต้องทำหน้าที่ดูแลรักษาและเสริมสร้างระบบ โดยการดูแลรักษา คือ การแก้ไขข้อผิดพลาดและการปรับเปลี่ยนแปลงตามสิ่งแวดล้อม การเสริมสร้างคือ การเพิ่มลักษณะเฉพาะใหม่ๆ และสิ่งที่จะเป็นประโยชน์กับระบบ

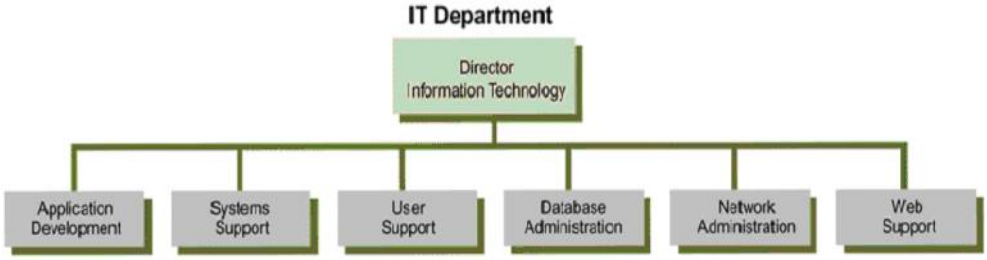
**ข้อแนะนำในการพัฒนาระบบ**

ข้อแนะนำพื้นฐานในการสร้างระบบสารสนเทศ ได้แก่

1. ดำเนินตามแผนของการพัฒนา
2. ต้องมั่นใจว่าผู้ใช้ได้มีส่วนร่วมในกระบวนการพัฒนาระบบ
3. กำหนดหลักไมล์เป็นระยะในการทบทวนโครงการและการประเมินค่า
4. กำหนดจุดตรวจเป็นช่วงๆ ระหว่างหลักไมล์ที่สำคัญ
5. กรอบแผนงานต้องยืดหยุ่นได้
6. ให้ข้อมูลเกี่ยวกับค่าใช้จ่ายและผลประโยชน์ที่จะได้รับอย่างเที่ยงตรงและเชื่อถือได้

## แผนกเทคโนโลยีสารสนเทศ

แผนกเทคโนโลยีสารสนเทศ เป็นหน่วยงานที่ทำหน้าที่พัฒนาและดูแลระบบข้อมูลสารสนเทศขององค์กร ซึ่งในแต่ละองค์กรอาจมีโครงสร้างหรือชื่อเรียกที่แตกต่างกันออกไป จากรูปที่ 1-17 แสดงให้เห็นแผนผังทั่วๆ ไป ของแผนกเทคโนโลยีสารสนเทศ ที่มี Networked PCs, Enterprise-wide Databases, Centralized Processing และ Web Site ประกอบด้วย 6 งานหลัก ได้แก่ งานพัฒนา งานดูแลระบบ งานบริการ งานดูแลฐานข้อมูล งานเครือข่าย และงานอินเทอร์เน็ต ซึ่งบางองค์กรอาจมีงานคาบที่เกี่ยวกันก็จะมีชื่อเรียกที่แตกต่างกันออกไป




รูปที่ 6 ผังองค์กรที่แสดงภาระงานหลักๆ ของเจ้าหน้าที่แผนกเทคโนโลยีสารสนเทศ

- **งานพัฒนาระบบ** มีหน้าที่ออกแบบ พัฒนาและติดตั้งระบบ แต่ปัจจุบันหลายองค์กรได้ใช้วิธีจัดตั้งคณะกรรมการพัฒนาระบบขึ้นมา ประกอบด้วย ผู้ใช้ระบบ ผู้จัดการ และบุคลากรด้านไอที เพื่อพิจารณาปัญหาร่วมกัน โดยรูปแบบที่นิยมกันมากได้แก่ JAD หรือ RAD
- **งานสนับสนุนระบบ** มีหน้าที่จัดเตรียมให้บริการด้านฮาร์ดแวร์และซอฟต์แวร์ เพื่อใช้กับระบบการคำนวณขนาดใหญ่ ระบบเครือข่าย ระบบข้อมูลเพื่อการประมวลผล และประสานเรื่องของเทคโนโลยีในอนาคต
- **งานสนับสนุนผู้ใช้** มีหน้าที่ดูแลช่วยเหลือผู้ใช้ระบบ ในด้านการให้ข้อมูลด้านเทคนิค การฝึกอบรม การแนะนำวิธีการใช้ต่างๆ อาจเรียกว่า หน่วยช่วยเหลือ (Help Desk) หรือ ศูนย์ข้อมูล (Information Center : IC)
- **งานดูแลฐานข้อมูล** มีหน้าที่ออกแบบฐานข้อมูล การจัดการฐานข้อมูล การดูแลความปลอดภัยและการสำรองข้อมูล ตลอดจนจัดทำระดับการเข้าถึงฐานข้อมูลของผู้ใช้
- **งานดูแลเครือข่าย** มีหน้าที่ดูแลบำรุงรักษาฮาร์ดแวร์และซอฟต์แวร์ บำรุงรักษาและดูแลความปลอดภัยของระบบ อีกทั้งควบคุมการเข้าถึงระบบของผู้ใช้ การติดตั้งระบบเครือข่าย การจัดการติดตามและดูแลซอฟต์แวร์ระบบเครือข่าย
- **งานสนับสนุนเว็บ** มีหน้าที่ของผู้ดูแลงานอินเทอร์เน็ต ส่วนใหญ่มักจะเรียก เว็บมาสเตอร์ (Web Masters) ทำหน้าที่ดูแลอินทราเน็ตภายในองค์กรและอินเทอร์เน็ต รวมถึงการออกแบบ จัดทำเว็บเพจ การตรวจตราความถี่ของสัญญาณการใช้งาน

## ตำแหน่งนักวิเคราะห์ระบบ

นักวิเคราะห์ระบบ ต้องตรวจสอบหาข้อมูล นำข้อมูลมาวิเคราะห์ ออกแบบ พัฒนา ติดตั้ง ติดตามประเมินผลของการใช้งานและการดูแลรักษาระบบสารสนเทศขององค์กร ต้องติดต่อกับผู้คนทั้งภายในและภายนอกองค์กร

### ภาระรับผิดชอบ

งานของนักวิเคราะห์ระบบจะคาบเกี่ยวกันระหว่างงานธุรกิจกับงานเทคนิค การแปลงความต้องการของธุรกิจเป็นโครงการไอทีที่ตรงกับความต้องการของบริษัท นักพัฒนาระบบจะประกอบด้วยหน้าที่หลายอย่าง เช่น สร้างลักษณะเค้าโครงธุรกิจ ทบทวนขั้นตอนทางธุรกิจ เลือกสรรฮาร์ดแวร์และซอฟต์แวร์ ออกแบบระบบงานสารสนเทศ ให้การอบรมผู้ใช้งาน รวมถึงการวางแผนในการทำพาณิชย์อิเล็กทรอนิกส์บนเว็บไซต์

### ทักษะและภูมิหลังที่ต้องการ

นักวิเคราะห์ระบบต้องมีความรู้ทางด้านเทคนิคอย่างดี มีทักษะในการสื่อสารด้านการพูดและการเขียนเป็นอย่างดี มีความสามารถสูงในการวิเคราะห์และเข้าใจถึงการทำงานและกระบวนการของธุรกิจ

โดยปกตินักวิเคราะห์ระบบต้องเป็นผู้นำทีมในการพัฒนาระบบ ต้องเป็นผู้วางแผนงาน ประมาณการณ์ และบริหารโครงการ พร้อมกับใช้ภาวะผู้นำและทักษะการสร้างทีมงานในการให้คำแนะนำและกระตุ้นการทำงานของลูกทีม

ความรู้ที่ก้าวหน้าทันสมัย เป็นสิ่งสำคัญยิ่งยวดในการเปลี่ยนแปลงสภาพแวดล้อมของธุรกิจและเทคนิควิธีปัจจุบันอินเทอร์เน็ตเปิดโอกาสอย่างมากให้สามารถหาความรู้และทักษะเพิ่มเติม

**สัปดาห์ที่** 2

**- หลักการวิเคราะห์ระบบงานขององค์กร**

**- การกำหนดเป้าหมายและวัตถุประสงค์ของการวิเคราะห์และออกแบบระบบ**

- Flow Chart

.....

**การสำรวจเบื้องต้น**

Preliminary Investigation

การสำรวจเบื้องต้นอยู่ในส่วนของการวางแผนระบบ ซึ่งเป็นระยะแรกในห้าระยะของวงจรการพัฒนาระบบ เป็นการเรียนรู้ความสำคัญในการทำความเข้าใจระบบการดำเนินธุรกิจ การวางแผนด้านเทคโนโลยี ที่สามารถจะสนองตามแผนกลยุทธ์ขององค์กร และวิธีการในการเริ่มพัฒนาจัดทำระบบ

**ความสำคัญของการวางแผนกลยุทธ์**

การวางแผนกลยุทธ์เป็นการวางแผนเพื่อกำหนดแนวทางขององค์กรในอนาคต แสดงการดำเนินการที่มีจุดหมายหรือหลักชัยในระยะยาว โดยมองข้ามงานประจำวัน มุ่งไปที่ระยะเวลา 3-20 ปีข้างหน้า

หลักที่ใช้ในการวางแผนกลยุทธ์ หลายองค์กรใช้ SWOT Analysis ศึกษาถึง จุดแข็ง (Strengths) จุดอ่อน (Weakness) โอกาส (Opportunities) และภัยคุมคาม (Threats) เช่น

- อะไรคือข้อได้เปรียบที่องค์กรมีมากกว่าคู่แข่ง และจะทำอย่างไรให้เกิดประโยชน์
- อะไรเป็นจุดเสียขององค์กร และจะหาทางป้องกันแก้ไขอย่างไร
- อะไรคือช่องทางอันเป็นโอกาสขององค์กร และจะทำอย่างไรที่จะใช้ประโยชน์นั้นได้อย่างสูงสุด
- อะไรคือปัญหาอุปสรรคที่องค์กรกำลังเผชิญอยู่ และจะหาแนวทางแก้ไขได้อย่างไร

เมื่อผ่านกระบวนการวิเคราะห์ จะทำให้สามารถคาดการณ์ล่วงหน้าได้ ดังนั้น จะต้องจัดทำเป็นแผนกลยุทธ์ เพื่อวางแผนด้านเทคนิค ด้านบุคลากร ด้านบริการ และด้านการเงิน เป็นต้น

ในกระบวนการจัดทำกลยุทธ์ จะต้องพิจารณาถึงจุดมุ่งหมาย บวกกับวิสัยทัศน์และค่านิยม เพื่อนำมากำหนดเป็นพันธกิจหลัก ที่จะนำไปสู่เป้าหมายตามวัตถุประสงค์ โดยอาศัยเทคโนโลยี วิธีการดำเนินการและระบบการสนับสนุนอื่นๆ เพื่อให้ได้ผลของการดำเนินธุรกิจที่เกิดประโยชน์กับผู้ที่มีส่วนได้ส่วนเสียกับองค์กร ในที่นี้หมายถึง ผู้ที่เกี่ยวข้องกับบริษัท อาจได้แก่ พนักงาน ลูกค้า บริษัทผู้จัดส่งสินค้า ผู้ถือหุ้น และชุมชน

**การจัดทำโครงการระบบสารสนเทศ**

การจัดทำโครงการระบบสารสนเทศ จะเริ่มจากความต้องการระบบ (System Request) เพื่อ

- เพิ่มความสามารถในด้านการบริการ
- ระบบที่มีคุณภาพมากขึ้น

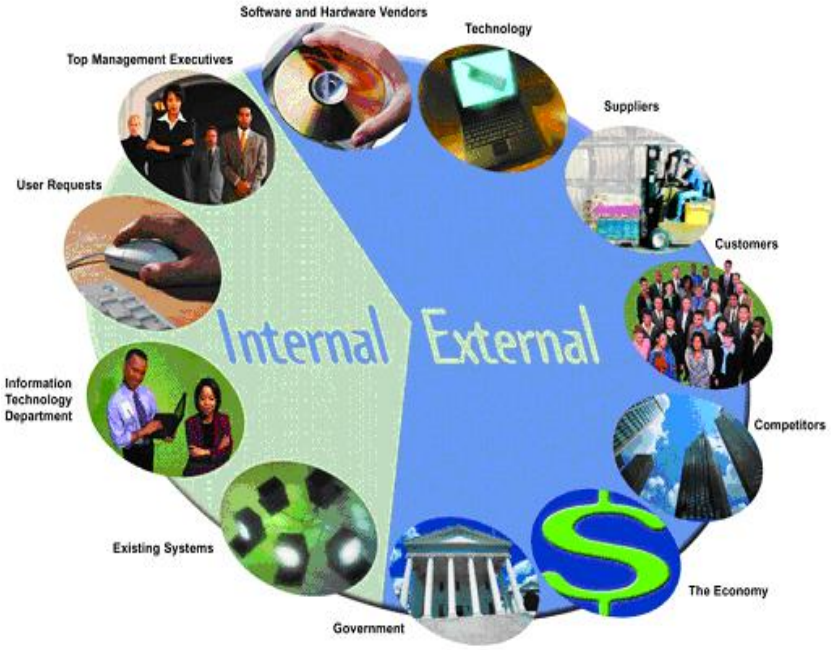
- ข้อมูลสารสนเทศที่เพียงพอ
- มีระบบการควบคุมที่ดี
- ลดค่าใช้จ่าย

รูปที่ 1 ห้าเหตุผลหลักของการความต้องการระบบงาน

**ปัจจัยที่มีผลกระทบกับการจัดทำระบบใหม่**

ปัจจัยที่มีผลกระทบกับการจัดทำระบบใหม่ มีทั้งปัจจัยภายในและภายนอก ประกอบด้วย

- ความต้องการของผู้ใช้
- ทิศทางจากผู้บริหารระดับสูง
- ปัญหาของระบบงานเดิมที่ใช้งานอยู่
- ฝ่ายเทคโนโลยีสารสนเทศ
- ภาวะเศรษฐกิจ
- เทคโนโลยี การพิจารณาถึงการเปลี่ยนแปลงที่รวดเร็วของเทคโนโลยี
- รัฐบาล นโยบายของรัฐบาลที่มีการปรับเปลี่ยน เช่น การเพิ่มหรือลดภาษีมูลค่าเพิ่ม
- ผู้ผลิตหรือผู้จัดจำหน่าย เช่น การปรับเปลี่ยนเวอร์ชันของซอฟต์แวร์
- ภาวะของการแข่งขัน
- ลูกค้า
- ผู้จัดจำหน่าย




รูปที่ 2 ปัจจัยภายในและปัจจัยภายนอกที่มีผลกระทบกับการจัดทำโครงการระบบงานสารสนเทศ

**แบบคำขอระบบงาน**

เพื่อให้ได้ข้อมูลของความต้องการครบถ้วน อาจจัดทำแบบฟอร์มคำขอใช้ระบบงานสำหรับกรอกความต้องการ โดยออกแบบให้ครอบคลุม

SWL **REQUEST FOR INFORMATION SYSTEMS SERVICES**

Date: _____
Submitted by: _____ Title: _____
Department: _____ Location: _____
Phone: _____ e-mail: _____

**REQUEST FOR:**

[ ] Correction of system error
[ ] System enhancement
[ ] New system

**URGENCY:**

[ ] Immediate attention needed
[ ] Handle in normal priority sequence
[ ] Defer until new system is developed

**DESCRIPTION OF REQUESTED SYSTEMS SERVICES:**
*(ATTACH ADDITIONAL DOCUMENTS AS NECESSARY)*

*(To be completed by the Information Technology Department)*

[ ] Approved

Assigned to IT contact person: _____
User: _____
Urgency code (1 low to 5 high): _____

[ ] Modified (see attached notes)
[ ] Rejected (see attached statement)

Date _____ Action: _____

รูปที่ 3 ตัวอย่างของแบบคำขอระบบงาน

**การประเมินความต้องการของระบบงาน**

ในหลายองค์กร มักจะกระทำโดยคณะกรรมการตรวจสอบระบบงาน (Systems Review Committee หรือ Computer Resources Committee) ซึ่งประกอบด้วยกลุ่มของผู้จัดการที่มีความเชี่ยวชาญและกลุ่มผู้ใช้งาน

**การลำดับความสำคัญของโครงการ**

คณะกรรมการจะต้องทำการพิจารณาแบบคำขอต่างๆ ที่ส่งเข้ามา ร่วมกันพิจารณากำหนดลำดับความสำคัญ โดยพิจารณาจากหลายๆ ด้าน เช่น ความจำเป็นเร่งด่วน ความเป็นไปได้

**การประเมินความเป็นไปได้ของโครงการ**

จากแบบคำขอระบบ จะถูกนำมาผ่านกระบวนการหาความคุ้มค่าในการจัดทำและการนำใช้งานต่อไป ที่เรียกว่า การศึกษาความเป็นไปได้ ซึ่งประกอบด้วย 3 ด้าน คือ

1. ความเป็นไปได้ด้านการปฏิบัติงานหรือความเป็นไปได้ในการนำไปใช้งาน (Operational Feasibility) สามารถทำงานตามที่ต้องการได้ เหมาะสมกับสภาพความเป็นจริง จำนวนบุคลากรที่เพียงพอ ทั้งเจ้าหน้าที่ไอทีและผู้ใช้งาน ทั้งนี้เกี่ยวข้องกับ
   - การสนับสนุนของผู้บริหารและผู้ใช้
   - ผู้ใช้มีส่วนร่วมในการวางแผน
   - ผลที่ได้ก่อให้เกิดประโยชน์กับผู้ใช้ ลูกค้า หรือภาพพจน์ของบริษัท
   - ระยะเวลาที่เหมาะสมในการทำโครงการ
2. ความเป็นไปได้ด้านเทคนิค (Technical Feasibility) เทคโนโลยีที่มีอยู่เพียงพอต่อการสร้างระบบหรือไม่ อุปกรณ์ที่มีอยู่สามารถรองรับการขยายหรือเพิ่มเติมตามที่ขอได้หรือไม่ จะต้องเพิ่มฮาร์ดแวร์หรือเครือข่ายอย่างไร ทั้งนี้เกี่ยวข้องกับ
   - ความพร้อมของผู้เชี่ยวชาญ
   - ความพร้อมของอุปกรณ์ที่สามารถนำมาใช้ได้อย่างเหมาะสมกับงาน
   - ความสามารถที่จะใช้งานได้ต่อไปในอนาคต
3. ความเป็นไปได้ด้านเศรษฐกิจ (Economic Feasibility) สามารถอยู่ในงบประมาณที่กำหนดหรือไม่ ผลประโยชน์ที่คาดหวังจะคุ้มกับการลงทุนหรือไม่ ทั้งนี้เกี่ยวข้องกับ
   - ค่าใช้จ่าย ทั้งประเภทจ่ายครั้งเดียว และประเภทต้องจ่ายต่อเนื่อง
   - ประโยชน์ที่ได้รับทั้งเชิงรูปธรรม และเชิงนามธรรม
   - ผลเสียหาย ที่อาจจะเกิดขึ้น หากไม่จัดทำโครงการนี้

   ซึ่งปัจจัยเหล่านี้ มักจะมีความเกี่ยวข้องกัน แบ่งเป็น

   - ผลประโยชน์เชิงรูปธรรม (Tangible Benefits) สามารถมองเห็นในรูปของตัวเงินได้อย่างชัดเจน เช่น จำนวนสินค้าที่ขายได้เพิ่มขึ้น กำไร สิ่งที่ทำให้เกิดการลดค่าใช้จ่ายหรือการเพิ่มรายได้ ทำให้ลดการทำงานล่วงเวลา ช่วยควบคุมสินค้าคงคลัง เพื่อช่วยให้ไม่ต้องลงทุนเก็บสินค้าไว้มากเกินความจำเป็น

- ผลประโยชน์เชิงนามธรรม (Intangible Benefits) ไม่สามารถมองเห็นเป็นตัวเงินได้อย่างชัดเจน เช่น ความพึงพอใจของลูกค้า ชื่อเสียงความน่าเชื่อถือของบริษัท ระบบที่ใช้งานได้ง่าย (User-friendly) ซึ่งช่วยเพิ่มประสิทธิภาพในการทำงานของพนักงาน ให้สามารถมีข้อมูลและวิธีการทำงานที่สะดวกมากขึ้น เว็บไซต์ใหม่ๆ ที่เสริมภาพลักษณ์ให้กับองค์กร

รูปที่ 4 การศึกษาความเป็นไปได้และทดสอบใน 3 ด้าน คือ ด้านการปฏิบัติงาน ด้านเทคนิค และด้านเศรษฐกิจ

**การกำหนดความเป็นไปได้**

การกำหนดความเป็นไปได้หรือการกำจัดสิ่งที่ไม่สามารถกระทำได้ เป็นขั้นตอนแรกของการประเมินความต้องการของระบบ เพื่อการวางแนวทางการศึกษาความเป็นไปได้ของระบบ สำรวจความเป็นไปได้ ตัดสิ่งที่ไม่สามารถกระทำได้ออก

**เกณฑ์ในการพิจารณาการประเมินแบบคำขอ**

หลังจากที่พิจารณากำจัดคำขอที่ไม่สามารถกระทำได้หรือยังไม่เหมาะสมออกไปแล้ว คณะกรรมการตรวจสอบระบบงาน จะต้องสร้างเกณฑ์ในการพิจารณาแบบคำขอที่เหลือ โดยคำนึงถึง

- ประโยชน์สูงสุดที่พึงจะได้รับ เช่น ทำให้การบริการลูกค้ารวดเร็วขึ้น
- งบลงทุนเหมาะสมหรือที่ไม่สูงนัก เช่น สามารถลดค่าใช้จ่ายในการเดินสายสัญญาณ
- ใช้ระยะเวลาในการดำเนินการที่พอควร เช่น บางแบบคำขอที่ต้องใช้เวลาในการพัฒนาหลายปี อาจไม่เหมาะสมหรือไม่ทันต่อความต้องการ

**ภาพรวมของการสำรวจเบื้องต้น**

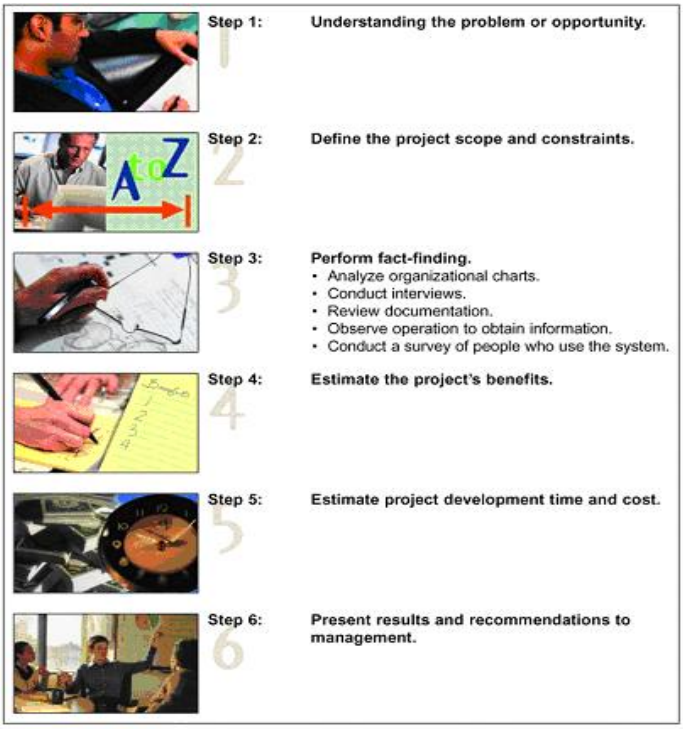
หลังจากที่คณะกรรมการพิจารณาแบบคำขอที่เป็นไปตามเกณฑ์ สามารถจะดำเนินการได้และจัดลำดับความสำคัญก่อนหลังของแบบคำขอแล้ว นักวิเคราะห์ระบบจะต้องทำการตรวจสอบหาข้อมูลเบื้องต้น กับผู้จัดการและผู้ใช้ เพื่อรวบรวมข้อเท็จจริงของปัญหาที่เกิดหรือข้อได้เปรียบที่มี ตลอดจน

ขอบเขต จุดอ่อน จุดแข็ง ประโยชน์ที่จะได้รับ ประมาณการณ์ค่าใช้จ่ายในการพัฒนา และระยะเวลาในการดำเนินการ เพื่อนำเสนอผู้บริหารพิจารณาต่อไป

**ขั้นตอนในการสำรวจความต้องการเบื้องต้น**

นักวิเคราะห์ระบบ มักจะดำเนินตามขั้นตอนการสำรวจความต้องการตามมาตรฐานทั่วๆ ไป ซึ่งแบ่งออกเป็น 6 ขั้นตอน ทั้งนี้อาจมีความจะแตกต่างกันบ้าง โดยขึ้นอยู่กับ

- ปัญหาที่แท้จริงหรือธรรมชาติของระบบงานนั้นๆ
- ขนาดของโครงการ
- ระดับของความเร่งด่วน




รูปที่ 5 ขั้นตอนในการสำรวจเบื้องต้น

**ขั้นตอนที่** 1. การทำความเข้าใจกับปัญหาหรือความจำเป็นของปัญหา

**ขั้นตอนที่** 2 . การกำหนดขอบเขตของระบบงานและข้อจำกัด

- การกำหนดขอบเขตของโครงการ
- การแยกแยะเงื่อนไขความต้องการกับข้อจำกัด

**ขั้นตอนที่** 3. การเก็บรวบรวมข้อเท็จจริง ซึ่งประกอบด้วย

- การวิเคราะห์แผนผังองค์กร
- การดำเนินการสัมภาษณ์
- การตรวจสอบเอกสาร จำเป็นที่จะต้องขอดูเอกสารที่ใช้ในปัจจุบัน
- การเฝ้าดูสังเกตการปฏิบัติงานจริง การสังเกตการณ์การทำงาน ทั้งการนำข้อมูลเข้าและผลลัพธ์ที่ได้ เพื่อเปรียบเทียบกับสิ่งที่สัมภาษณ์หรือตรวจสอบกับเอกสารที่ได้รับ

- การสำรวจเป็นกลุ่ม การสัมภาษณ์ส่วนใหญ่มักจะกินเวลา บางครั้งอาจใช้วิธีหาข้อมูลเป็นกลุ่มใหญ่ โดยให้กรอกข้อมูลในแบบสอบถาม ซึ่งอาจจะไม่ครอบคลุมเหมือนการสัมภาษณ์ แต่มีจุดเด่นที่สามารถให้ข้อมูลได้มากๆ ได้จากทุกพื้นที่

**ขั้นตอนที่ 4.** กำหนดประโยชน์ที่จะได้จากโครงการทั้งในเชิงการปฏิบัติงาน เชิงเทคนิค และเชิงเศรษฐศาสตร์ โดยแสดงให้เห็นถึงประโยชน์และข้อดีที่จะได้รับทั้งปัจจุบันและในอนาคต

**ขั้นตอนที่ 5.** ประมาณการณ์ค่าใช้จ่ายและระยะเวลาในการดำเนินการ

**ขั้นตอนที่ 6.** นำเสนอผลของการสำรวจความต้องการและโครงการให้กับผู้บริหาร

ขั้นตอนการนำเสนอ อาจเป็นในรูปแบบของเอกสารหรือรูปแบบภาพเพื่อการนำเสนอ หรืออาจใช้ทั้งสองวิธีคู่กันไป

รูปแบบของรายงานการสำรวจเบื้องต้น มักประกอบด้วย 7 ส่วน ดังนี้

1. บทนำ แสดงภาพรวมของรายงาน ประกอบคำอธิบายระบบงานอย่างย่อ และรายชื่อของผู้ที่ดำเนินการสำรวจเบื้องต้น
2. สรุปความต้องการของผู้ใช้
3. ผลการดำเนินการ แสดงผลของการสำรวจความต้องการเบื้องต้น รวมทั้ง ขอบเขตของโครงการ จุดแข็งและความเป็นไปได้
4. ข้อเสนอแนะ พร้อมเหตุผลประกอบ
5. ระยะเวลาและค่าใช้จ่ายโดยประมาณ อธิบายค่าใช้จ่ายที่จะต้องใช้ในการจัดหาและติดตั้งระบบ ตลอดจนค่าใช้จ่ายในการบำรุงรักษาระบบ
6. ผลประโยชน์ที่คาดว่าจะได้ ทั้งที่คำนวณเป็นตัวเงินได้และไม่ได้
7. ภาคผนวก อาจเป็**น ตาราง กราฟ หรือเอกสารที่ได้จากการเก็บข้อมูล**

Flow Chart

**การเขียนผังงาน** (Flowchart)

ผังงานคือแผนภาพที่มีการใช้สัญลักษณ์รูปภาพและลูกศรที่แสดงถึงขั้นตอนการทำงานนิยมใช้ในการออกแบบโปรแกรม หรือ ระบบทีละขั้นตอน รวมไปถึงทิศทางการไหลของข้อมูลตั้งแต่แรกจนได้ผลลัพธ์ตามที่ต้องการ

**ประโยชน์ของผังงาน**

-ช่วยลำดับขั้นตอนการทำงานของโปรแกรม สามารถนำไปเขียนโปรแกรมได้โดยไม่สับสน

- ช่วยในการตรวจสอบ และแก้ไขโปรแกรมได้ง่าย เมื่อเกิดข้อผิดพลาด

- ช่วยให้การดัดแปลงแก้ไข ทำได้อย่างสะดวกรวดเร็ว

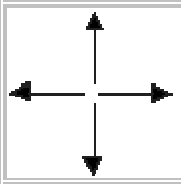
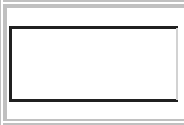
- ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้อย่างงายและรวดเร็วขึ้น

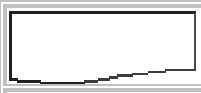
**วิธีการเขียนผังงานที่ดี**

1. ใช้สัญลักษณ์ตามที่กำหนดไว้
2. ใช้ลูกศรแสดงทิศทางการไหลของข้อมูลจากบนลงล่างหรือ ซ้ายไปขวา
3. คำอธิบายในภาพสั้นกะทัดรัดเข้าใจง่าย
4. ทุกแผนภาพต้อมีลูกศรแสดงทิศทางเข้า- ออก
5. ไม่ควรโยงเส้นโยงเชื่อมผังงานที่มีอยู่ไกลมาก ๆ ควรใช้สัญลักษณ์จุดเชื่อมต่อแทน
6. ผังงานควรมีการทดสอบความถูกต้องของการทำงานก่อนนำไปเขียนโปรแกรม

**ผังงานโปรแกรม** ( Program Flowchart )

การเขียนผังโปรแกรมจะประกอบไปด้วยการใช้สัญลักษณ์มาตรฐานต่าง ๆ ที่เรียกว่า สัญลักษณ์ ANSI ( American National Standards Institute ) ในการสร้างผังงาน ดังตัวอย่างที่แสดงในรูปต่อไปนี้

| | |
|---|---|
| | จุดเริ่มต้น / สิ้นสุดของโปรแกรม |
|  | ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล |
|  | ใช้แสดงคำสั่งในการประมวลผล หรือการกำหนดค่าข้อมูลให้กับตัวแปร |
| | แสดงการอ่านข้อมูลจากหน่วยเก็บข้อมูลสำรองเข้าสู่หน่วยความจำหลักภายในเครื่องหรือการแสดงผลลัพธ์จากการประมวลผลออกมา |
| | การตรวจสอบเงื่อนไขเพื่อตัดสินใจ โดยจะมีเส้นออกจารูปเพื่อแสดงทิศทางการทำงานต่อไป เงื่อนไขเป็นจริงหรือเป็นเท็จ |

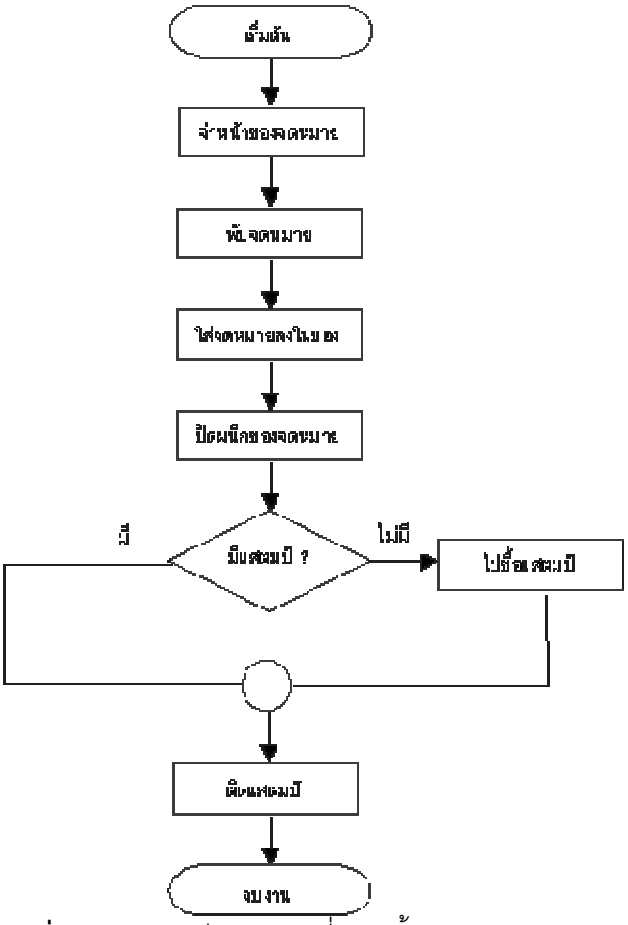
| | |
|---|---|
|  | แสดงผลหรือรายงานที่ถูกสร้างออกมา |
| | แสดงจุดเชื่อมต่อของผังงานภายใน หรือเป็นที่บรรจบของเส้นหลายเส้นที่มาจากหลายทิศทางเพื่อจะไปสู่การทำงานอย่างใดอย่างหนึ่งที่เหมือนกัน |
| | การขึ้นหน้าใหม่ ในกรณีที่ผังงานมีความยาวเกินกว่าที่จะแสดงพอในหนึ่งหน้า |

**รูปที่**1 แสดง สัญลักษณ์ในการเขียนผังงานโปรแกรม

**ผังงานกับชีวิตประจำวัน**

การทำงานหลายอย่างในชีวิตประจำวัน จะมีลักษณะที่เป็นลำดับขั้นตอน ซึ่งก่อนที่ท่านจะได้ศึกษาวิธีการเขียนผังงานโปรแกรม จะแนะนำให้ท่านลองฝึกเขียนผังงานที่แสดงการทำงานในชีวิตประจำวันวันก่อนเพื่อเป็นการสร้างความคุ้นเคยกับสัญลักษณ์รูปภาพต่าง ๆ ที่จะมีใช้ในผังงานโปรแกรมต่อไป ดัง **ตัวอย่าง** 1 เขียนผังงานที่แสดงขั้นตอนการส่งจดหมาย

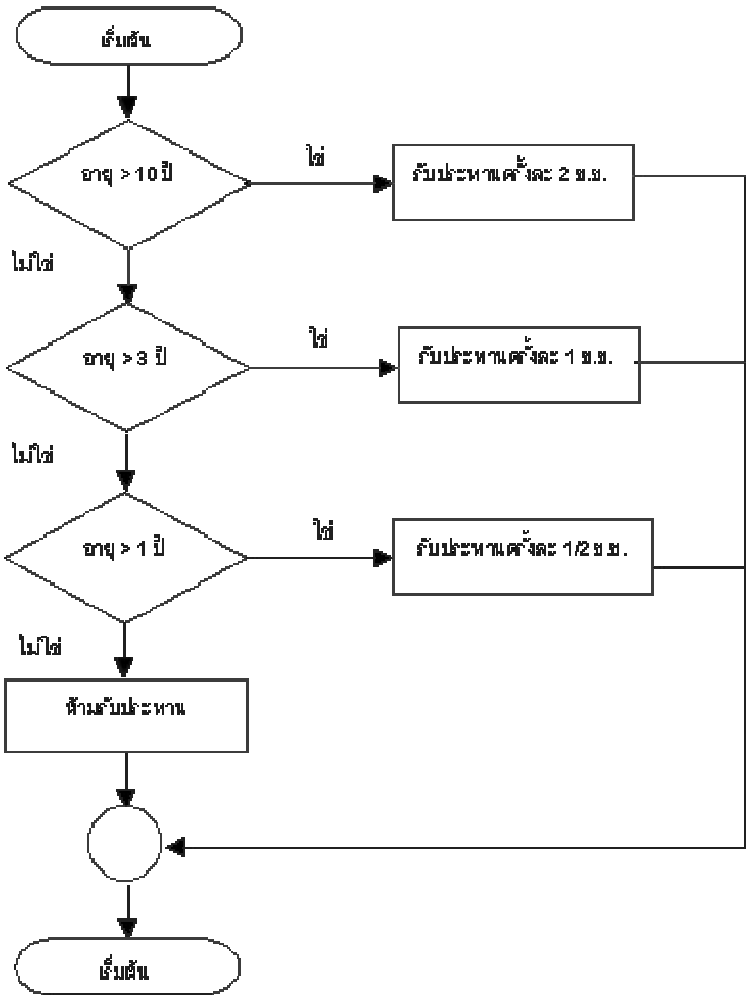
**ตัวอย่าง** 1 เขียนผังงานที่แสดงขั้นตอนการส่งจดหมาย




**รูปที่** 2 แสดงการเขียนผังงานที่แสดงขั้นตอนการส่งจดหมาย

**ตัวอย่างที่ 2 เขียนผังงานแสดงวิธีการรับประทานยา ที่แบ่งขนาดรับประทานตามอายุของผู้ทานดังนี้**

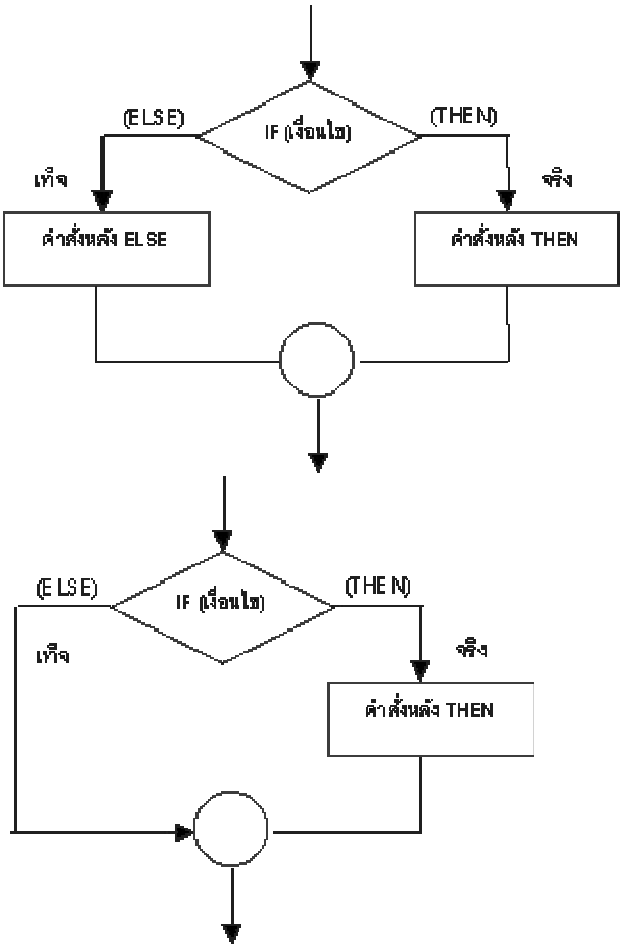
อายุมากกว่า 10 ปี รับประทานครั้งละ 2 ช้อนชา
อายุมากกว่า 3 ปี ถึง 10 ปี รับประทานครั้งละ 1 ช้อนชา
อายุมากกว่า 1 ปี ถึง 3 ปี รับประทานครั้งละ 1/2 ช้อนชา
แรกเกิดถึง 1 ปี ห้ามรับประทาน




**รูปที่** 3 แสดงการเขียนผังงานแสดงวิธีการรับประทานยา

**โครงสร้างการทำงานแบบมีการเลือก** ( Selection )

เป็นโครงสร้างที่ใช้การตรวจสอบเงื่อนไขเพื่อการทำงานอย่างใดอย่างหนึ่ง โดยโครงสร้างแบบนี้จะมีอยู่ด้วยกัน 2 รูปแบบ คือ IF - THEN - ELSE และ IF - THEN



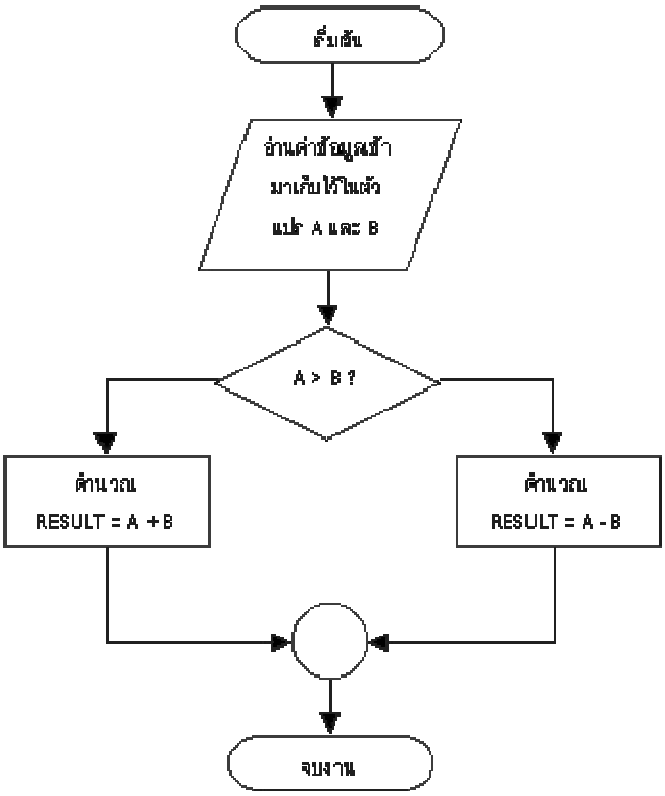

**รูปที่** 4 แสดงโครงสร้างผังงานแบบมีการเลือก

โครงสร้างแบบ IF - THEN - ELSE เป็นโครงสร้างที่จะทำการเปรียบเทียบเงื่อนไขที่ใส่ไว้ในส่วนหลังคำว่า IF และเมื่อได้ผลลัพธ์จากการเปรียบเทียบก็จะเลือกว่าจะทำงานต่อในส่วนใด กล่าวคือถ้าเงื่อนไขเป็นจริง ( TRUE ) ก็จะเลือกไปทำงานต่อที่ส่วนที่อยู่หลัง THEN แต่ถ้าเงื่อนไขเป็นเท็จ ( FALSE ) ก็จะไปทำงานต่อในส่วนที่อยู่หลังคำว่า ELSE

แต่ถ้าสำหรับโครงสร้างแบบ IF - THEN เป็นโครงสร้างที่ไม่มีการใช้ ELSE ดังนั้น ถ้ามีการเปรียบเทียบเงื่อนไขที่อยู่หลัง IF มีค่าเป็นจริง ก็จะไปทำส่วนที่อยู่หลัง Then แต่ถ้าเงื่อนไขเป็นเท็จ ก็จะไปทำคำสั่งที่อยู่ถัดจาก IF - THEN แทน

**ตัวอย่าง 3 การเขียนผังงานอ่านค่าข้อมูลเข้ามาเก็บไว้ในตัวแปร A และ B แล้วทำการเปรียบเทียบในตัวแปรทั้งสอง** โดยมีเงื่อนไขดังนี้

ถ้า A มากกว่า B ให้คำนวณหาค่า A - B และเก็บผลลัพธ์ไว้ในตัวแปรชื่อ RESULT
ถ้า A น้อยกว่าหรือเท่ากับ B ให้คำนวณหาค่า A + B และเก็บผลลัพธ์ไว้ในตัวแปรชื่อ RESULT



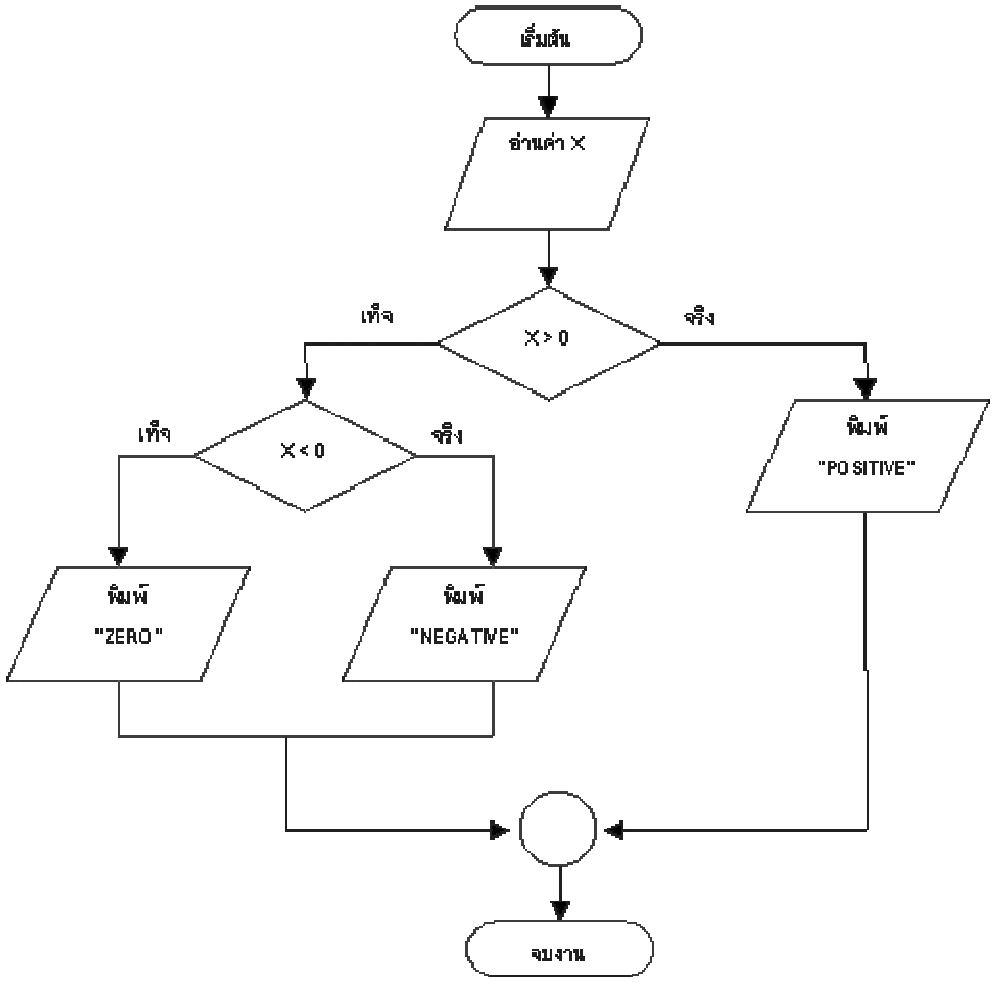

**รูปที่** 5 แสดงการเขียนผังงานอ่านค่าข้อมูล

**ตัวอย่าง** 4 **การเขียนผังงานเปรียบเทียบค่าข้อมูลที่เก็บอยู่ในตัวแปร** X โดยมีเงื่อนไขดังนี้

ถ้า X > 0 ให้พิมพ์คำว่า " POSITIVE NUMBER "

ถ้า X < 0 ให้พิมพ์คำว่า " NEGATIVE NUMBER "

ถ้า X = 0 ให้พิมพ์คำว่า " ZERO NUMBER "




**รูปที่** 6 แสดงการเขียนผังงานเปรียบเทียบค่าข้อมูล

## โครงสร้างการทำงานแบบมีการทำงานซ้ำ

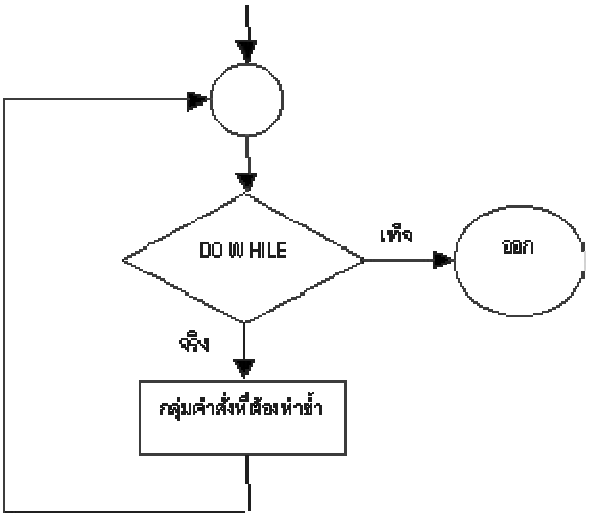
เป็นโครงสร้างที่มีการประมวลผลกลุ่มคำสั่งซ้ำหลายครั้ง ตามลักษณะเงื่อนไขที่กำหนด อาจเรียก การทำงานซ้ำแบบนี้ได้อีกแบบว่า การวนลูป ( Looping ) โครงสร้างแบบการทำงานซ้ำนี้จะมีอยู่ 2 ประเภท คือ

DO WHILE

DO UNTIL

### DO WHILE

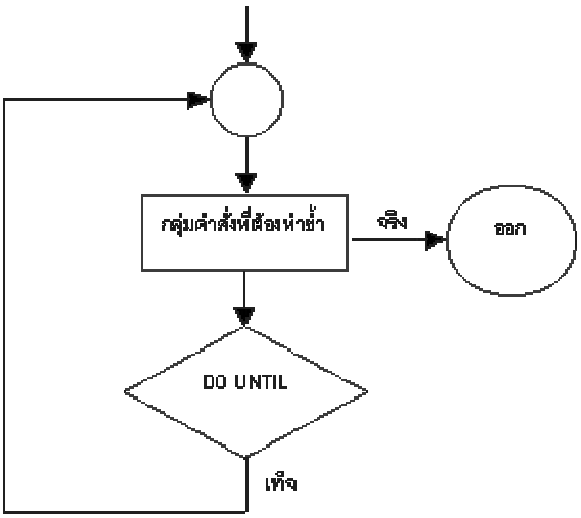
เป็นโครงสร้างที่มีการทดสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็นจริงก็จะเข้ามาทำงานในกลุ่มคำสั่งที่ต้องทำซ้ำ ซึ่งเรียกว่าการเข้าลูป หลังจากนั้นก็จะย้อนกลับไปตรวจสอบเงื่อนไขใหม่อีก ถ้าเงื่อนไขยังคงเป็นจริงอยู่ ก็ยังคงต้องทำกลุ่มคำสั่งซ้ำหรือเข้าลูปต่อไปอีก จนกระทั่งเงื่อนไขเป็นเท็จ ก็จะออกจากลูปไปทำคำสั่งถัดไปที่อยู่ถัดจาก DO WHILE หรืออาจเป็นการจบการทำงาน




**รูปที่** 7 แสดงโครงสร้างการทำงานซ้ำแบบ DO WHILE

DO UNTIL

เป็นโครงสร้างการทำงานแบบทำงานซ้ำเช่นกัน แต่มีการทำงานที่แตกต่างจาก DO WHILE คือ จะมีการเข้าทำงานกลุ่มคำสั่งที่อยู่ภายในลูปก่อนอย่างน้อย 1 ครั้ง แล้วจึงจะไปทดสอบเงื่อนไข ถ้าเงื่อนไขเป็นเท็จก็จะมีการเข้าทำกลุ่มคำสั่งที่ต้องทำซ้ำอีก หลังจากนั้นก็จะย้อนกลับไปตรวจสอบเงื่อนไขใหม่อีก ถ้าเงื่อนไขยังคงเป็นเท็จอยู่ ก็ยังต้องทำกลุ่มคำสั่งซ้ำหรือเข้าลูปต่อไปอีก จนกระทั่งเงื่อนไขเป็นจริง จึงจะออกจากลูปไปทำคำสั่งถัดจาก UNTIL หรืออาจเป็นการจบการทำงาน




**รูปที่** 8 แสดงโครงสร้างการทำงานซ้ำแบบ DO UNTIL

**สรุปข้อแตกต่างระหว่าง DO WHILE และ DO UNTIL มีดังนี้**

1. DO WHILE ในการทำงานครั้งแรกจะต้องมีการตรวจสอบเงื่อนไขก่อนทุกครั้ง ก่อนที่จะมีการเข้าลูปการทำงาน

2. DO UNTIL การทำงานครั้งแรกจะยังไม่มีการตรวจสอบเงื่อนไข แต่จะเข้าไปทำงานในลูปก่อนอย่างน้อย 1 ครั้งแล้วจึงจะไปตรวจสอบเงื่อนไข

3. DO WHILE จะมีการเข้าไปทำงานในลูปก็ต่อเมื่อตรวจสอบเงื่อนไขแล้วพบว่า เงื่อนไขเป็นจริง แต่เมื่อพบว่าเงื่อนไขเป็นเท็จ ก็จะออกจากลูปทันที

4. DO UNTIL จะมีการเข้าไปทำงานในลูปก็ต่อเมื่อตรวจสอบเงื่อนไขแล้วพบว่า เงื่อนไขเป็นเท็จ แต่เมื่อพบว่าเงื่อนไขเป็นจริง ก็จะออกจากลูปทันที

**สัปดาห์ที่** 3

- **การรวบรวม** Requirement
- Activity Diagram (UML 2.0)

.....

**การรวบรวม** Requirement

**ขอบเขตของข้อมูล**

เพื่อให้ได้ข้อมูลตามที่ต้องการ และเหมาะสม ต้องการมีกำหนดขอบเขตของข้อมูล ที่ต้องการซึ่งแบ่งเป็น 4 กลุ่มดังนี้

**1. ข้อมูลเกี่ยวกับองค์กร**

เป้าหมายขององค์กร (Goals of the Company)

เป้าหมายขององค์กรคือ เป้าหมายระยะยาวและแผนการกลยุทธ์ที่กำหนดไว้ ซึ่งเป็นตัวบอกให้รู้ถึงจุดมุ่งหมายใน 5 ถึง 7 ปี ข้างหน้า วัตถุประสงค์ในการศึกษา ถึงเป้าหมายขององค์กร เพื่อกำหนดทิศทางและขอบเขตการวิเคราะห์ระบบ

โครงสร้างขององค์กร (Organizational Structre)

การศึกษาถึงโครงสร้างองค์กรนั้น เพื่อต้องการให้ทราบถึงหลักการการบริหารงาน และทิศทางขององค์กร นอกจากจะได้ทราบว่าใครเป็นใคร ทำหน้าที่อะไร จะได้นำเสนอข้อมูลตาม ที่แต่ระบบต้องการ

วัตถุประสงค์ (Objectives and Purposes)

ในองค์กรแบ่งเป็นแผนกหรือหน่วยงานย่อย ๆ ซึ่งในแต่ละแผนก แต่ละหน่วยงาน ก็มีวัตถุประสงค์ของตนเองโดยต้องสอดคล้องกันและสนับสนุนเป้าหมายขององค์กร วัตถุประสงค์ของการศึกษาวัตถุประสงค์ของแต่ละแผนก คือ เพื่อเป็นแนวทางให้รู้ทิศทางของการไหล (รับ/ส่ง) ของข้อมูล

นโยบาย (Policies)

นโยบายเป็นกฎในการดำเนินงานขององค์กร ซึ่งต้องทำให้บรรลุเป้าหมายขององค์กร การศึกษาถึงนโยบายขององค์กรนั้น เพื่อให้ทราบถึงความต้องการระบบสารสนเทศขององค์กร

**2. ข้อมูลเกี่ยวกับบุคลากร**

อำนาจหน้าที่ความรับผิดชอบ (Authority and Responsibility Relationships)

จากโครงสร้างขององค์กรสามารถทราบได้ว่าใครเป็นใครในองค์กร และอำนาจหน้าที่ของเขา แต่บางครั้งการดำเนินงานจริง ๆ ก็แตกต่างจากโครงสร้างที่กำหนดวัตถุประสงค์ ของการศึกษาข้อมูลนี้คือ เพื่อให้ให้ทราบถึงการปฏิบัติงานจริง ๆ ของบุคลากรแต่ละระดับ และทราบว่า ใครมีหน้าที่ มีอำนาจในการตัดสินใจด้านใดบ้างตามที่ปฏิบัติอยู่

หน้าที่ (Job duties)

ศึกษาการปฏิบัติงาน หน้าที่การงานขององค์แต่ละตำแหน่งในองค์กร โดยการศึกษา จากเอกสารต่าง ๆ รวมทั้งคู่มือที่เกิดขึ้นในระบบ เพื่อให้ทราบความเป็นไปของการดำเนินงาน

ความสัมพันธ์ (Interpersonal relationships)

ในองค์กรใด ๆ ก็ตาม ความสัมพันธ์ ของบุคลากรเกิดขึ้นโดยที่ทุกคนชอบที่จะทำงานร่วมกันเพื่อให้งานเสร็จลุล่วงไปด้วยดี และให้เวลาน้อยที่สุด ซึ่งบางครั้งการทำงานดังกล่าว อาจทำให้แตกต่างไปจากระบบการทำงานที่ได้กำหนดไว้ในโครงสร้างขององค์กร นักวิเคราะห์ระบบ จำเป็นต้องหาข้อมูลจากเหตุการณ์ที่เกิดขึ้นจริง ๆ มากกว่าที่จะศึกษาจากข้อความในเอกสารที่เขียนไว้ว่า การทำงานควรจะเป็นอย่างไร วัตถุประสงค์ของการศึกษาความสัมพันธ์ของบุคลากรนั้น

ความต้องการสารสนเทศ(Information needs)

การศึกษาความต้องการสารสนเทศนั้นเพื่อให้ทราบว่าสารสนเทศที่ใช้ในระบบปัจจุบัน เป็นสารสนเทศที่ได้ตรงตามความต้องการมากแค่ไหน และทราบถึงความต้องการสารสนเทศจริง ๆ ของผู้บริหารแต่ละระดับ นักวิเคราะห์ต้องทำการเปรียบเทียบความต้องการของสารสนเทศที่ได้รับ เพื่อช่วยในการประเมินความสมดุลย์ของการไหลของข้อมูลภายในระบบได้

**3. ข้อมูลเกี่ยวกับงาน**

การไหลของข้อมูล (Task and Work flows)

วัตถุประสงค์ในการศึกษาการทำงานและระบบงาน คือ เพื่อทราบว่าการไหล หรือการรับ/การส่งข้อมูลภายในระบบ และการแปลงข้อมูลโดยหน้าที่ภายในระบบ การศึกษานี้ทำได้ โดยการรวบรวมเอกสารแบบฟอร์มต่างๆ ที่เกิดขึ้นภายในระบบงานปัจจุบันในแต่ละกระบวนการ โดยทั่วไปจะเน้นการศึกษาที่ข้อมูล และ การเปลี่ยนแปลงข้อมูลในโครงสร้างระบบ

วิธีการและกระบวนการทำงาน (Methods and procedures)

การศึกษาวิธีการ และกระบวนการทำงานนี้เป็นการศึกษาที่กระบวนการจริงๆ โดยมุ่งเน้นจุดศูนย์กลางของงานโดยศึกษาว่างานอะไร โดยใคร ด้วยเครื่องมืออะไร มีตารางการทำงานอย่างไร ภายใต้กฎเกณฑ์ใด ซึ่งแตกต่างจากการศึกษาการทำงานและระบบงาน การศึกษาข้อมูลดังกล่าวนี้เน้นที่การกระทำและกระบวนการ

ตารางการทำงานและปริมาณงาน (Work schedules Volumes)

ศึกษาว่าจำนวนงานเท่าไรที่ต้องการให้เสร็จ ในระยะเวลาที่กำหนดหรือไม่ ซึ่งจะทำให้รู้ว่าควรนำเอาระบบคอมพิวเตอร์มาช่วยในการพัฒนาระบบใหม่หรือไม่

มาตรการปฏิบัติงาน (Performance criteria)

ระบบการทำงานของทุกองค์กรมีการกำหนดมาตรฐานการปฏิบัติงาน เพื่อวัดและประเมินผลงาน ซึ่งในการวัดและประเมินงานนั้นไม่เพียงแต่ดูที่ตารางการทำงาน ปริมาณ แต่ดูที่คุณภาพ ความถูกต้อง ความเชื่อถือได้ และการยอมรับ สารสนเทศของงานนั้นด้วย วัตถุประสงค์ของการศึกษานี้ คือเปรียบเทียบการทำงานจริงๆ และมาตรฐานองงานว่าเป็นไป ตามที่กำหนดหรือไม่ เพื่อทราบถึงคุณภาพของระบบงาน ความบกพร่องของปัจจัยที่เกิดความบกพร่อง

เครื่องมือที่ใช้ควบคุม (Contorl Mechanisms)

ศึกษาว่าอะไรเป็นตัวควบคุมระบบการทำงาน

**4. ข้อมูลเกี่ยวกับสภาพแวดล้อม**

การกำหนดขอบเขตงาน (Resources avaible)

วัตถุประสงค์ของการศึกษาเกี่ยวกับทรัพยากรที่มีอยู่นั้น คือ ต้องการทราบว่า ทรัพยากรที่องค์กรมีอยู่นั้นมีอะไรบ้าง จัดสรรอย่างไร ใช้อย่างคุ้มค่าหรือไม่ และถ้าต้องการมีการออกแบบใหม่ต้องซื้ออุปกรณ์ใดเพิ่ม สามารถนำทรัพยากรเก่ามาใช้กับระบบงานใหม่ ได้หรือไม่ ทรัพยากรในที่นี้ รวมถึงบุคคลากรในองค์กรด้วย เพื่อเพิ่มฐานความรู้ของบุคลากร

---

**วิธีรวบรวมข้อมูล**

วิธีการรวบรวมข้อมูลอาจทำได้หลายวิธี ขึ้นอยู่กับชนิดข้อมูลที่เราต้องการ การรวบรวมข้อมูลแยกได้ดังต่อไปนี้

1. การรวบรวมจากเอกสาร (Documents)
2. แบบสอบถาม (Questionnaire)
3. การสัมภาษณ์ (Interview)
4. การสังเกต (Observation)

**1. การรวบรวมจากเอกสาร (Documents)**

แหล่งข้อมูลจากเอกสารที่มีอยู่ในองค์กรที่นักวิเคราะห์และออกแบบระบบ ควรจะศึกษาและรวบรวมข้อมูลมีดังต่อไปนี้

- โครงสร้างขององค์กร (Organization Charts)
- นโยบาย (Policy Manuals)
- คู่มือวิธีการปฏิบัติงาน (Methods and Prodeures Manuals)
- หน้าที่ความรับผิดชอบ (Job Descriptions)

• แบบฟอร์มรายงานต่าง ๆ (Forms and Reports)
• การรับ/ส่งเอกสารและกระบวนการทำงาน (Document Flow and Work Flow Diagrams)
• ระบบงาน (System Flowchats)

กรณีที่องค์กรมีระบบงานคอมพิวเตอร์อยู่แล้ว เอกสารที่ต้องศึกษาเพิ่มเติมคือ
• เอกสารเกี่ยวกับโปรแกรมคอมพิวเตอร์ (Computer Program Documentation)
• คำอธิบายข้อมูล (Data Dictionary listing)
• คู่มือการใช้ระบบงานคอมพิวเตอร์ (Computer Operations Manuals)

**2. แบบสอบถาม (Questionnaire)**

แบบสอบถามเป็นเอกสารที่ทำขึ้นเพื่อถามคำถามที่ให้ผู้ตอบตอบคำถาม ที่ผู้ออกแบบสอบถามต้องการ เมื่อเปรียบเทียบกับการสัมภาษณ์ แบบสอบถามจะเป็นแบบ Impersanal ซึ่งเป็นการหาข้อมูลได้ทีละมาก ๆ จากจำนวนมาก เหมาะกับการที่ต้องหาข้อมูลจากคนจำนวนมาก

คุณสมบัติของแบบสอบถามมีดังต่อไปนี้
1. แบบสอบถามที่ตรงประเด็น (Validity)
2. มีความเชื่อถือได้ (Reliability)
3. มีเหตุมีผล (Face validity)

**การวางแผนสำหรับการใช้แบบสอบถาม**

1. กำหนดวัตถุประสงค์ที่แน่นอนสำหรับการรวบรวมข้อมูลที่ต้องการ แบบสอบถามที่ได้นั้นต้องกำหนดทางเลือกข้อคิดเห็น เป็นทั้งคำถามปิดและคำถามเปิดเพื่อ ขอความคิดเห็นจากผู้ตอบ
2. กำหนดผู้ตอบแบบสอบถาม ถ้ามีจำนวนผู้ตอบมากกว่าควรใช้การสุ่ม
3. กำหนดแบบสอบถามเพื่อให้ผู้ตอบยอมรับแบบสอบถามนั้น
4. กำหนดรูปแบบ (Forms) ของแบบสอบถามและวิธีการประเมินผล

**การเขียนแบบสอบถาม ชนิดของคำถาม**

รูปแบบของคำถามในแบบสอบถามแบ่งเป็น 2 ชนิด ดังนี้
1. คำถามปลายเปิด (Open-ended questions)
เป็นแบบสอบถามที่ไม่มีทางเลือกให้เลือกตอบ แต่ให้ผู้ตอบแบบสอบถาม แสดงความคิดเห็นของตนเอง

**ตัวอย่างคำถามปลายเปิด**

คุณคิดว่าหน่วยงานต้องปรับปรุงอะไรเป็นอย่างแรกเพื่อให้สามารถให้บริการลูกค้าได้รวดเร็วขึ้น

.....

.....

2. คำถามปลายปิด (Closed-ended questions)

เป็นคำถามที่มีคำตอบให้ผู้ตอบแบบสอบถาม โดยการ เลือกคำตอบตามความคิดเห็น แบบสอบถามชนิดนี้มีหลายรูปแบบดังนี้

2.1 Multiple - choice

เป็นตัวเลือกให้เลือกตอบ ดังตัวอย่าง

1. คุณบริการลูกค้าวันละประมาณกี่คน

[] 0-5 คน

[] 6-10 คน

[] 11-15 คน

[] มากกว่า 15 คน

2.2 Rating - Scale

เป็นคำถามที่ให้ตอบคำถามที่เป็นอัตราในการตอบคำถามนั้น ๆ ดังตัวอย่าง

2. คุณเห็นด้วยกับนโยบายการนำเอาคอมพิวเตอร์มาใช้ในองค์กรมากแค่ไหน

[] เห็นด้วยอย่างมาก

[] เห็นด้วย

[] ไม่แน่ใจ

2.3 Ranking - Scale

เป็นคำถามที่ผู้ตอบแบบสอบถามต้องเลือกคำตอบระดับความคิดเห็น เรียงลำดับความสำคัญ ดังตัวอย่าง

3. เรียงลำดับตามความพอใจในการทำงานของคุณ

__ เงินเดือน

__ผลประโยชน์

__ เพื่อนร่วมงาน

__สภาพแวดล้อมในการทำงาน

__หัวหน้างาน

**หลักการเขียนแบบสอบถาม**

1. คำถามควรเป็นคำถามในเพียงหัวข้อเดียว
2. คำถามควรเหมาะสมกับผู้ตอบที่จะตอบได้
3. เรียงคำถามให้ต่อเนื่องสัมพันธ์กัน
4. คำถามควรออกแบบสำหรับคะเนที่ง่ายในการวิเคราะห์
5. คำถามควรเป็นคำถามที่ชัดเจน กะทัดรัด เข้าใจง่าย มีข้อความ

**ข้อดีของแบบสอบถาม**

1. ประหยัดเวลา
2. ง่าย สะดวก และรวดเร็ว

**ข้อเสียของแบบสอบถาม**

1. การทำแบบสอบถามที่ได้ผลตามความต้องนั้นยาก
2. มีข้อจำกัดในการได้ข้อมูลตามความต้องการ

3. **การสัมภาษณ์** (Interview)

เป็นการเก็บรวบรวมข้อมูลโดยการพูดคุยซักถามบุคลากร เพื่อให้ได้ข้อมูล ตามที่นักวิเคราะห์ระบบต้องการ ซึ่งเป็นวิธีหาข้อมูลที่สำคัญอีกวิธีหนึ่ง นักวิเคราะห์ต้องเตรียมตัวสัมภาษณ์ โดยศึกษาข้อมูลเกี่ยวกับองค์กรบางส่วนเพื่อเตรียมคำถามในการสัมภาษณ์ ผู้ที่จะสัมภาษณ์จะเป็นผู้บริหารระดับสูง ระดับกลาง ระดับล่าง เรื่องมาจนถึงพนักงาน

การสัมภาษณ์เริ่มจากผู้บริหารระดับสูงนั้น เพื่อต้องการทราบถึงวัตถุประสงค์ ขององค์กรและความต้องการของระบบของผู้บริหารแต่ละระดับ สิ่งที่ควรคำนึงถึงกาสัมภาษณ์นั้นคือ ควรทำให้ผู้ถูกสัมภาษณ์รู้สึกว่าเป็นการสนทนากันตามปกติ โดยมิให้เขามีความรู้สึกว่าเขาจะถูกแย่งงาน และพยายามเปิดโอกาสให้เขาออกความคิดเห็นและมีส่วนร่วมในการออกแบบระบบ เพื่อให้เขา มีความรู้สึกที่ดีต่อระบบใหม่ที่กำลังพัฒนาขึ้น

**ขั้นตอนการสัมภาษณ์**

1. กำหนดแหล่งที่มาของข้อมูล โดยทั่วไปนักวิเคราะห์ควรสัมภาษณ์ตั้งแต่ ผู้บริหารระดับสูงลงมา เพื่อทราบวัตถุประสงค์ขององค์การ ปัญหาความต้องการสารสนเทศและความต้องการระบบ ซึ่งการสัมภาษณ์ตั้งแต่ผู้บริหารระดับสูงลงมาจนถึงผู้ปฏิบัติงานนั้น ทำให้ผู้ปฏิบัติงานให้ความร่วมมือมากยิ่งขึ้น เพราะเขาได้รู้ว่าผู้บริหารเป็นผู้ต้องการให้มีการวิเคราะห์และออกแบบระบบ

2. เตรียมตัวสัมภาษณ์ก่อนการสัมภาษณ์นั้น ผู้สัมภาษณ์ต้องกำหนด ความต้องการโดยการเขียนวัตถุประสงค์ในการสัมภาษณ์ จากนั้นก็เขียน Outline เพื่อให้ครอบคลุม สิ่งที่ควรสัมภาษณ์หลังจากนั้นต้องทำการนัดหมายผู้ที่จะสัมภาษณ์ นักวิเคราะห์ระบบต้องศึกษาข้อมูล เกี่ยวกับผู้ที่เขาจะสัมภาษณ์ เช่น หน้าที่ความรับผิดชอบในองค์กร

ขั้นตอนการดำเนินการสัมภาษณ์ การดำเนินการสัมภาษณ์เป็นขั้นตอน ที่สำคัญที่สุดซึ่งจะต้องงทำให้การสัมภาษณ์เป็นไปตามความต้องการหรือไม่นั้น นักวิเคราะห์ควรจะมีทักษะ ในการสัมภาษณ์ โดยต้องให้การสัมภาษณ์เป็นไปอย่างต่อเนื่อง ควรสัมภาษณ์ตัวต่อตัว ยิ่งไปกว่านั้น นักวิเคราะห์ระบบควรจะพยายามพูดให้น้อยที่สุด เพื่อจะได้เปิดโอกาสให้ผู้ถูกสัมภาษณ์ได้แสดงความคิดเห็น ให้มากที่สุด

**ในการสัมภาษณ์นักวิเคราะห์ควรหลีกเลี่ยงสิ่งต่อไปนี้**

1. การตั้งคำถามนำ (Beware of Leading questions)

คำถามนำนั้นควรเป็นคำถามที่เป็นกลาง ไม่ควรเอียงไปทางใดทางหนึ่ง ซึ่งจะมีอิทธิพลต่อผู้ถูกสัมภาษณ์ ควรหลีกเลี่ยงคำถามต่อไปนี้

" จริงหรือไม่ที่ว่า..... "

" คุณเห็นด้วยหรือไม่ว่า..... "

2. หลีกเลี่ยงการนำเพื่อสรุป (Avoid premature conclusion)

นักวิเคราะห์ควรระวังในการสัมภาษณ์ ซึ่งควรแน่ใจว่าได้ข้อมูลครบถ้วน ตามความต้องการ แล้วค่อยถามนำเพื่อสรุป เพราะถ้าถามเพื่อนำสรุปโดยที่ยังได้ข้อมูลไม่ครบอาจทำให้ยาก ที่จะดำเนินการสัมภาษณ์ใหม่

3. ไม่ควรที่จะคล้อยตาม

กรณีผู้ถูกสัมภาษณ์แสดงความคิดเห็นในทางลบต่อองค์กรและเกิดความไม่พอใจ นักวิเคราะห์ไม่ควรคล้อยตาม เพียงแต่ทำความเข้าใจเท่านั้น

4. อย่าถูกจูงใจโดยใคร

ไม่ควรที่จะมีใครเป็นผู้มีอิทธิพลต่อเรา เช่น พยายามที่จะเอาในผู้บริหาร จนทำให้ปิดบังความเป็นจริงหรือปัญหาขององค์กร

**ข้อดีของการสัมภาษณ์**

1. ข้อมูลที่ได้เป็นข้อมูลที่มีคุณภาพ เนื่องจากเป็นข้อมูลที่ได้รับโดยการพูดคุยกัน ซึ่งเกิดขึ้นจากการดำเนินงานจริงๆ ในระบบ

2. นักวิเคราะห์ได้ความคิดเห็นและคำแนะนำในการออกแบบระบบ ทำให้ผู้ถูกสัมภาษณ์มีส่วนร่วมในการออกแบบ ซึ่งทำให้ลดการต่อต้านระบบใหม่

**ข้อเสียของการสัมภาษณ์**

1. เสียเวลามาก
2. นักวิเคราะห์อาจคล้อยตามความคิดเห็นที่เป็นอคติ (Bias) ของผู้ถูกสัมภาษณ์

**4. การสังเกต** (Observation)

การสังเกตเป็นการรวบรวมข้อมูล โดยการดูกระบวนการทำงานจริง ๆ ของระบบอย่างเดียว โดยไม่มีการสอบถามใด ๆ อาจใช้แบบสอบถามเป็นแนว ผู้สังเกตการณ์ต้องรู้ว่า จะสังเกตอะไร และต้องทำการบันทึกข้อมูลเกี่ยวกับการปฏิบัติงานในขณะสังเกต สรุปการสังเกต ข้อมูลที่ได้จะถูกต้องแม่นยำเพียงใด ขึ้นอยู่กับความสามารถของผู้สังเกตการณ์ ซึ่งผู้สังเกตการณ์ ต้องมีความละเอียดรอบคอบมีไหวพริบและความยุติธรรม

การสังเกตการปฏิบัติการในองค์การ อาจกระทำโดยให้ผู้สังเกตรู้ตัว หรือสังเกต โดยไม่ให้ผู้ถูกสังเกตรู้ตัวก็ได้

**ข้อดีของการสังเกต**

1. ข้อมูลที่ได้จากเหตุการณ์จริง ๆ (System-related tasks) ซึ่งเป็นข้อมูลที่เชื่อถือได้
2. ข้อมูลที่ได้จากการเก็บข้อมูล จากการสังเกตโดย ไม่มีการเตรียมตัว เหมือนการทำแบบสอบถาม หรือการสัมภาษณ์
3. ข้อมูลที่ได้เป็นข้อมูลที่น่าเชื่อมาก เนื่องจากผู้สังเกตการณ์เป็นผู้เห็นเหตการณ์จริงๆ ด้วยตา

**ข้อเสียของการสังเกต**

1. ไม่สะดวก กรณีที่กระบวนการเกิดขึ้นไม่บ่อยก็ต้องใช้เวลา
2. กรณีที่บุคลากรรู้ว่ามีคนสังเกตการทำงานของเขา เขาอาจจะไม่ได้ทำเหมือนปกติ ที่เคยทำ ทำให้ได้ข้อมูลที่ไม่ตรงกับความจริง
3. ต้องใช้คนที่มีความสามารถสูงในการสังเกต

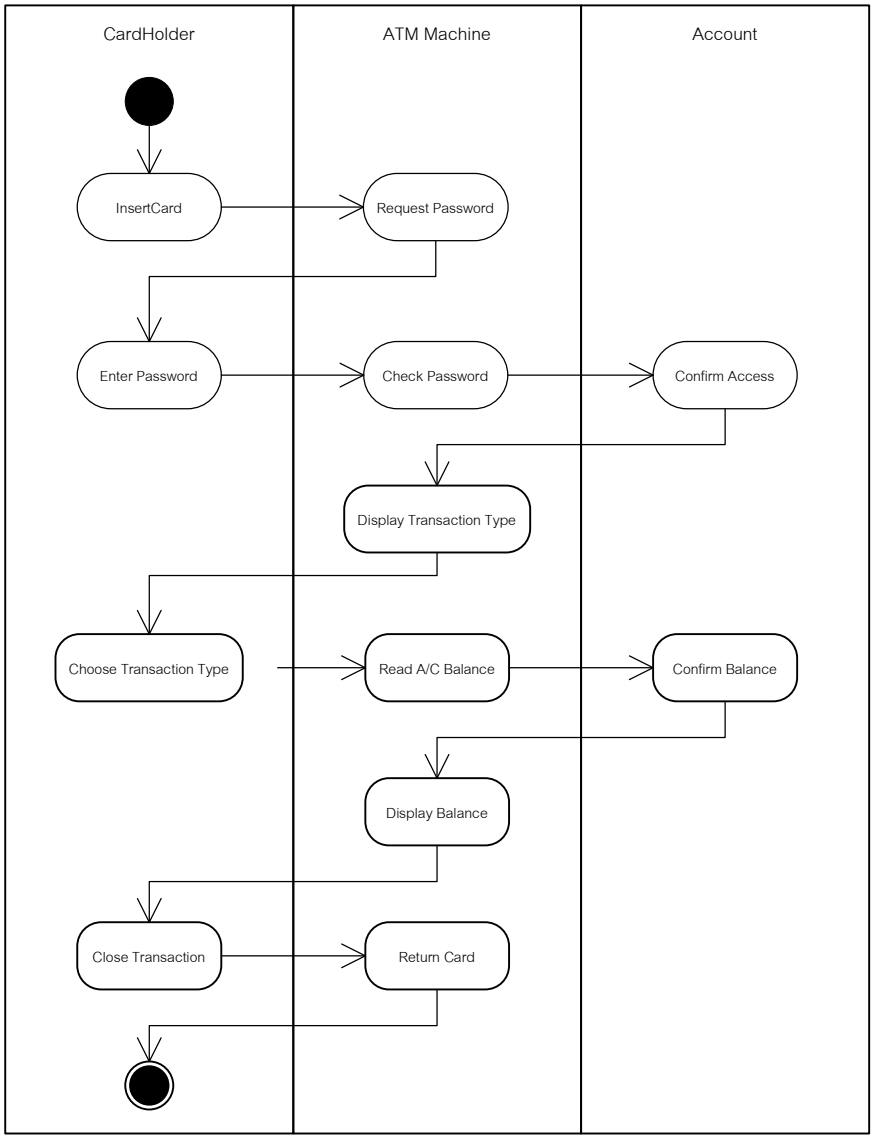
**5. การสุ่ม** (Sampling)

การสุ่มใช้การการหาข้อมูลที่มีบุคลากรจำนวนมาก เหตุการณ์มากและ มีการเปลี่ยนแปลง การทำงานมากไม่สามารศึกษาจากทุกกลุ่ม ทุกกระบวนการได้ ซึ่งทำให้เสียเวลา และค่าใช้จ่ายมาก จึงต้องนำเอาวิธีทางสถิติ เข้ามาช่วยโดยการใช้การสุ่มเอาข้อมูลบางส่วน วิธีนี้เป็นวิธีที่ได้ผลดีอีกวิธีหนึ่ง

## Activity Diagram

- Activity Diagram เป็นแผนภาพที่ใช้ที่แสดงขั้นตอนการทำงานของ use case (เช่นเดียวกับ Sequence Diagram และ Collaboration Diagram) แต่จะเน้นไปที่งานย่อยของวัตถุ โดยจะมีกระบวนการทำงานคล้ายกับ Flowchart
- Activity Diagram บางครั้งมีลักษณะคล้าย Swimlane โดยจะแบ่งกลุ่มกิจกรรมที่เกิดขึ้นเป็นช่อง โดยกำกับแต่ละช่องด้วยชื่อของ Object แต่ละ Swimlane แสดงถึงกิจกรรมที่เกิดขึ้นกับ Object นั้นๆ

**ตัวอย่าง Activity Diagram การสอบถามยอดบัญชีจากตู้ ATM**



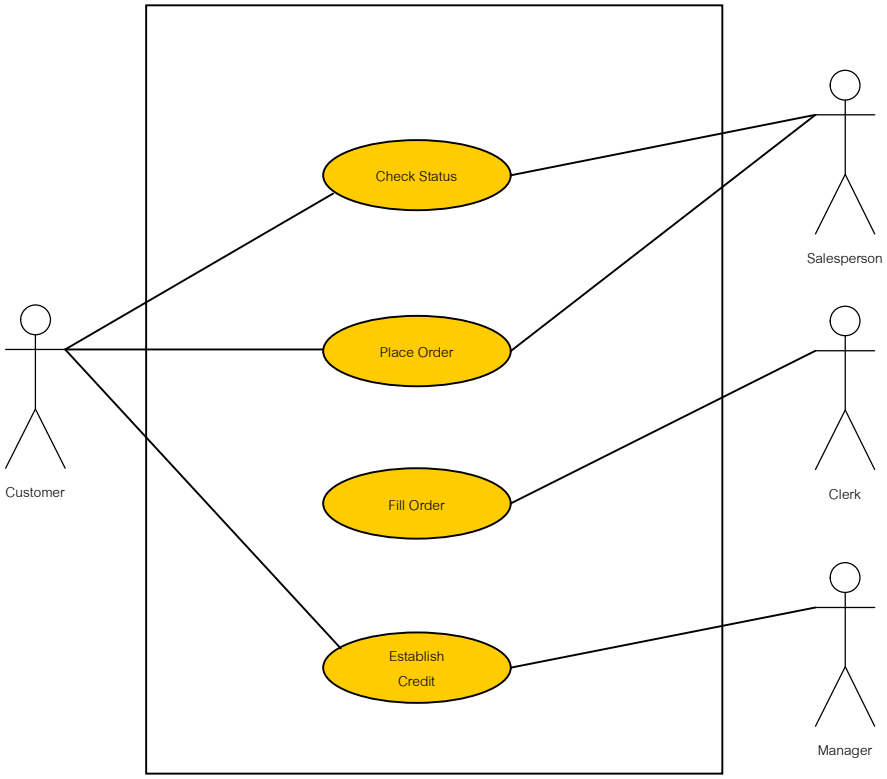
**สัปดาห์ที่** 4

**การวิเคราะห์ความต้องการของระบบ**

- Use Case Diagram (UML 2.0)

.....

- Use Case Diagram เป็นแผนภาพที่ใช้ที่แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน Use Case Diagram ประกอบด้วย
  - Actor คือ ผู้ที่กระทำกับระบบ อาจเป็นผู้ที่ทำการส่งข้อมูล, รับข้อมูล หรือ แลกเปลี่ยนข้อมูลกับระบบนั้นๆ เช่น ลูกค้ากับระบบสั่งซื้อสินค้าทางโทรศัพท์
  - Use Case คือ หน้าที่หรืองานต่างๆในระบบ เช่น การเช็คสต็อค การสั่งซื้อสินค้า เป็นต้น
  - Relationship คือ ความสัมพันธ์ระหว่าง Use Case กับ Actor




ตัวอย่าง Use Case การสั่งซื้อสินค้าทางโทรศัพท์

**สัปดาห์ที่** 5

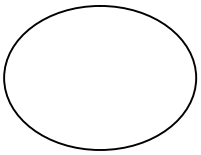
Context Diagram

.....

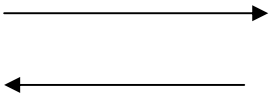
เป็นตัวแบบที่ใช้สำหรับการสร้างภาพรวมทั้งหมดของระบบ โดยจะประกอบด้วย

1. Entity ใช้แทน สิ่งที่อยู่นอกระบบ โดยใช้สัญลักษณ์

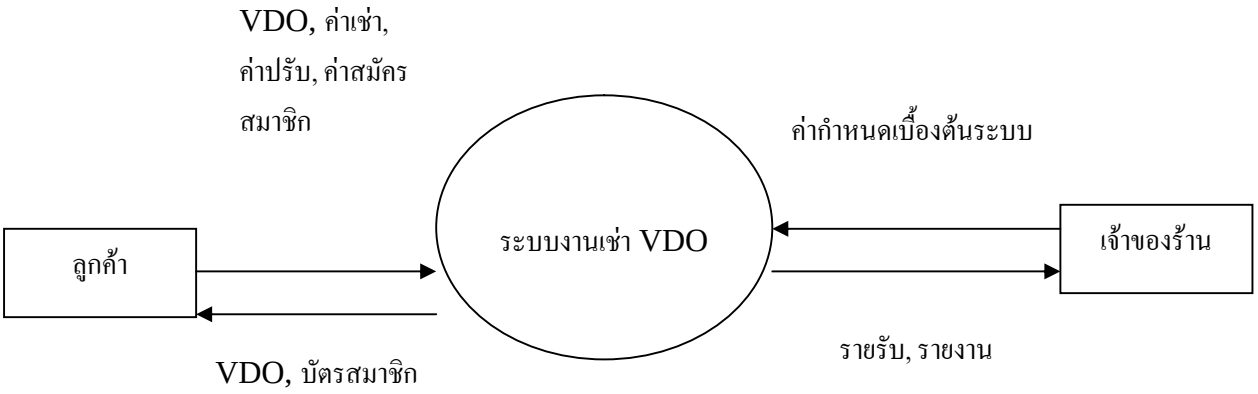
2. Context ใช้แทน ภาพรวมของระบบทั้งหมด โดยใช้สัญลักษณ์



3. ทิศทางการไหลของข้อมูล ใช้สัญลักษณ์



ตัวอย่างระบบงานการเช่า VDO



**สัปดาห์ที่** 6

**Process Modeling**

- **Waterfall, Spiral, RAD, CBSE, Agile**

……

**Software Process Model**

**Software Process model** คือตัวแบบหรือโครงร่าง ของ Software Process ต่างๆที่จะต้องทำ Process Model เช่น Waterfall Model , Iterative Development, Component- Base Software Engineering เป็นต้น ซึ่งมีมากมายกว่านี้ และแต่ละ model ก็ มีข้อดีข้อเสียแตกต่างกันไป
สิ่งที่อาจารย์ต้องการเพื่อสู่ Goal

- Process
- Technique
- Tool
- Doc
- วิเคราะห์ว่า Process ที่มีอยู่ มีข้อดีข้อเสียอย่างไร

Goal เพื่อสามารถดูในสิ่งที่ทำแล้วทราบปัญหาเพื่อจะสามารถแก้ปัญหาได้ เข้าใจ Business process เข้าใจกระบวนการและนำไปสู่การ apply เพื่อใช้งานได้จริง

**Waterfall Model**

| Communication -> planning -> modeling -> construction -> deployment |
|---|

คือ model ที่มีการทำเป็น process โดยเริ่มจากการติดต่อกับ user เพื่อรวมรวบ requirement ลูกค้าเข้ามา จากนั้นก็มาเริ่มวางแผนในการพัฒนา แล้วสร้างเป็น Model ตาม Requirement ที่ได้มา ต่อด้วยการเริ่มเขียนโปรแกรมและทดสอบโปรแกรม แล้วสุดท้ายก็ Deploy โดยจะเห็นว่า กระบวนการของ waterfall นั้นจะเป็นลักษณะเป็นขั้นเป็นตอนเรียงลำดับกันไป โดยจะเป็นลักษณะ Concurrent ซึ่งก็หมายความว่า เมื่อเสร็จขั้นตอนหนึ่งแล้วก็ส่งต่อไปอีกขั้นต่อไปเพื่อดำเนินการต่อไป

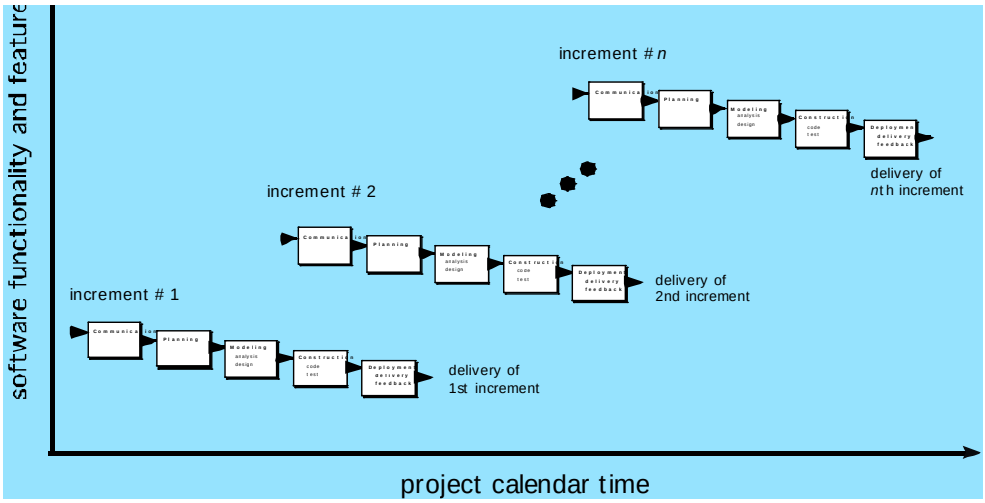
**ข้อดี**

- ง่ายต่อการบริหารจัดการ เพราะรู้ status และเป็น step

**ข้อเสีย**

- user ไม่สามารถบอก requirement ได้หมดในครั้งแรกได้
- เมื่อ requirement เปลี่ยน จะทำอย่างไรดี ? แน่นอนจะต้องมีการเริ่มกระบวนการใหม่ ซึ่งบางครั้งก็ไม่เหมาะนัก อาจจะต้องมีการรื้อกระบวนการใหม่ เพื่อให้เข้ากับ requirement ของ user ซึ่งก็ไม่ใช่วิธีการที่ดี

Incremental Process Model




มี 2 model ย่อย

1. Incremental
2. RAD (Rapid Application Development)

Increment คือ การแบ่งย่อย Process ที่เป็นอิสระจากกันแล้วนำมาทำในลักษณะ waterfall แล้วโอกาสในการ success ก็จะมีมากขึ้น แต่ปัญหาก็คือว่า Process นั้นจะมีความง่ายในการแยกมากน้อยเพียงใด เพราะบาง Process นั้นมีความซับซ้อนมากเกินกว่าจะแยกได้ แต่หลักการก็คือจะแยก Process นั่นแหละ โดยอาจจะแยกในลักษณะแบ่ง scope ก็ได้ เช่น ลักษณะการออกใบเสร็จ ซึ่งบางครั้งสามารถออกได้ ณ ขณะนั้นได้เลย ไม่ต้องมี transaction มาก เป็นต้น ทั้งนี้ต้องเลือกทำในสิ่งที่เป็น core product ให้ได้ก่อน

**ข้อดี**

- เห็นความคืบหน้า
- supplementary ทั้งหลาย ในสิ่งที่รู้และไม่รู้ (User ยังไม่แจ้ง) ก็ต้องสามารถรองรับได้ด้วย
- requirement ที่เปลี่ยนบ่อย ๆ สามารถที่จะรองรับและปรับปรุงต่อไปได้
- มี business value เนื่องจาก user สามารถเห็นความคืบหน้าของงานได้อย่างรวดเร็ว ซึ่งในลักษณะของกระบวนการนั้นก็สามารถสร้าง core ขึ้นมาก่อน แล้วค่อยสร้าง supplementary ตามมาก็จะทำให้เห็น business value ได้ดียิ่งขึ้น
- ไม่เหมาะกับ Business deadline
- จัดการกับ Technology ใหม่ ๆ ได้อย่างไร สมมติ ว่ากำลังพัฒนา 1 อยู่และใช้ tech 1 อยู่ แต่ซักพักมี tech 2 เข้ามา จะมีการจัดการ tech อย่างไรเพื่อให้สามารถจัดการได้อย่างมีประสิทธิภาพ
- Focus ที่ business value

พูดง่าย ๆ ก็คือ แบ่งงานย่อยแล้วสร้างแต่ละส่วน เพื่อนำเสนอ แล้วค่อยขยายเพื่อให้เห็น business value มากขึ้น

**ข้อเสีย**

- ในขณะที่ทำ core product เสร็จแล้วและมาทำ supplementary และถ้ามีการแก้ไข core ก็จะทำให้เกิดการแก้ไขทั้งหมดเช่นกัน
- ยังเป็น waterfall อยู่

**RAD Model**

RAD เป็น Rapid Application Development (RAD) ก็เป็น increment model อีก model หนึ่ง แต่ว่าเน้นที่มีการใช้ tool ต่างๆ เข้ามาช่วย เพื่อที่จะทำให้ process แต่ละ process เสร็จได้รวดเร็ว สิ่งที่สำคัญที่สุดของ RAD ก็คือ ความต้องการของ user ซึ่ง RAD จะมา result ปัญหา

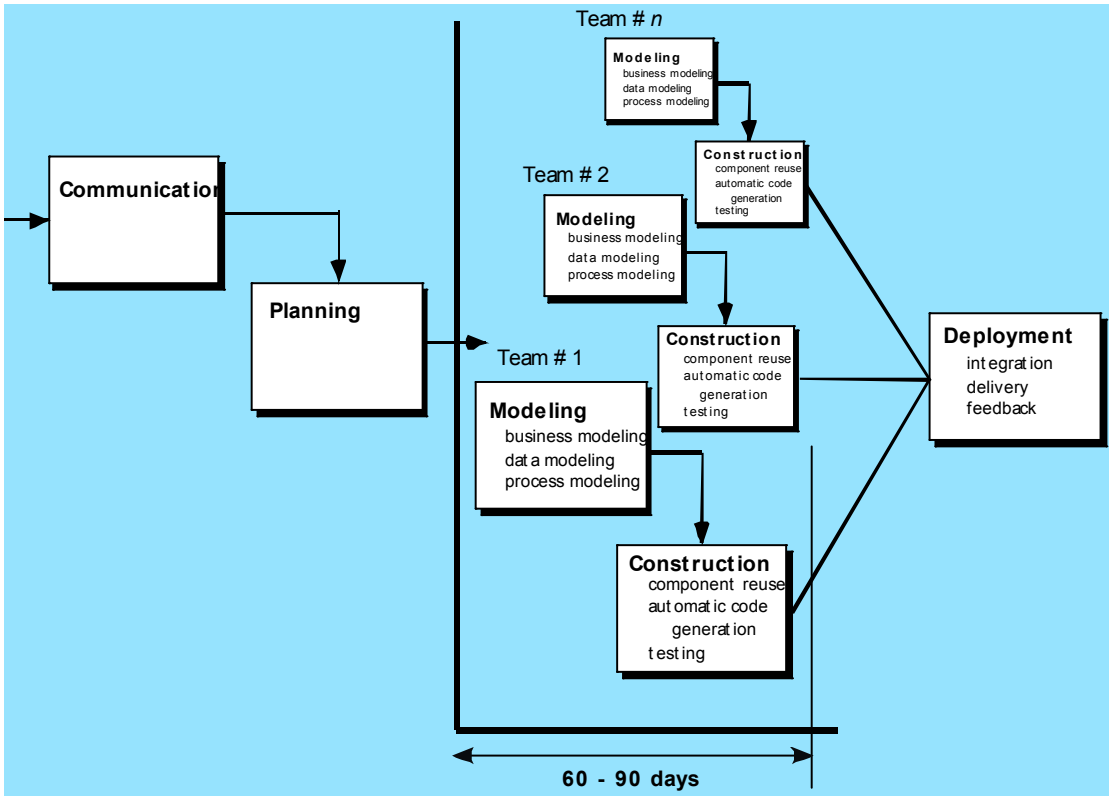
RAD เป็นincremental process Model ซึ่งเน้น Short development cycle หมายความเน้นให้ product ออกมารวดเร็ว การที่เราผลิตซอร์ฟแวร์ ออกมาแล้ว user ได้ใช้เลย สิ่งที่ได้กลับมาคือ business value สิ่งที่สำคัญที่สุดเลยคือ business value ครับ

RAD จะทำได้ก็คือใช้ 2 methology เข้าผสมกันเราเรียกว่าเป็น

- Component based approaches ก็คือการซื้อ software component ต่างๆ เข้ามาประกอบกัน
- มี tools ต่างๆเข้ามาช่วย อย่างเช่นพวก CASE tools, interface generator แล้วก้อ report Generator คำว่าพวก Report Generator หรือ interface generator Visual programming ซึ่งเป็น scripting language ในตัวอย่างเช่น visual basic ต่างๆ ที่คุณสามารถสร้าง user interface ได้จากพวก visual programming เหล่านี้

<u>รายละเอียดของ RAD model</u>

สิ่งที่สำคัญของ RAD model คือ เริ่มต้นก็ยังมีการ communication ก็คือมีการคุยกับ user มีการเก็บข้อมูลเก็บ requirement ต่างๆ จากนั้นก็มีการ planning เพื่อทำ project scheduling ต่างๆ เมื่อทำ planning เสร็จแล้วจะแบ่งเป็นหลายๆ ทีม เข้ามาทำงานพร้อมๆ กัน ในระยะเวลาที่ใกล้เคียงกัน อาจจะมีทีมแรกเข้ามาทำ modeling ของตัวเอง setup modeling ทำdata modeling process ขณะเดียวกันทีมที่ 2 ที่3 ก้อมีการทำแบบนี้เหมือนกันแต่ทุกอย่างอยู่ในช่วงเวลา 60-90 วันแล้วก้อ delivery ให้ user ใช้ หมายความในหน่วย application development เนี่ยเรามีการ develop software หลายๆ ส่วนกลุ่มตัวพร้อมๆ กันในหลายๆทีม แล้วก็มารวมกัน




อะไรที่เหมาะสมในการใช้ RAD model ก็คือ process ที่มี

Time constrain

- **scalable scope** คำว่า Scalable scope หมายความว่าเราสามารถตัดแบ่ง scope ของงานเราได้ ในบางงานจะตัดแบ่ง scope ไม่ได้ ถ้าเรามี time constrain แล้วเราสามารถ focus ได้ว่า scalable เราก็สามารถ delivery ได้
- **modularity** คือว่าเราสามารถแบ่งงานออกเป็น module ได้ เราพัฒนาเป็น module ได้ เพราะอย่าลืมว่าเรามีหลายทีม แต่ละทีมก็ทำงานแตกต่างกันไป เสร็จแล้วก็เอามา integrate ร่วมกัน นี่คือข้อดีของ RAD

**ข้อดี**

- ***ช่วยให้มีงานออกมาเพื่อนำเสนอได้เร็วขึ้น เป็นการเพิ่ม Business value ได้เป็นอย่างดี***
- สามารถเปลี่ยนแปลงได้ตาม user requirement ในเวลาที่ต้องการได้
- สามารถเปลี่ยนแปลงได้ตามความต้องการ
- สามารถรองรับ tech ที่เปลี่ยนแปลงได้ด้วย

**ข้อเสีย**

- A large scale project requires sufficient staff for RAD team: RAD team ใช้ employee จำนวนมาก แต่ละทีมก็จะมีการสร้าง Business model ของตัวเอง ถ้าต้องมีการ interface เข้าร่วมกัน ปัญหาก็คือว่า ถ้าเจอระบบมีความ complexity ถ้าแบ่ง scale ไม่ได้หรือแบ่ง modularity ไม่ได้ชัดเจน

การ interface ระหว่างทีมที่ 1 ที่ 2 จะทำยังงัย เพราะฉะนั้นจำนวนของ staff ที่มีอยู่ใน RAD team ต้องมีจำนวนมาก

-Lot of a activities for both Developer and Customer: เนื่องจากว่า RAD team ต้องการทำอะไรที่รวดเร็ว ดังนั้น แต่ละทีมจะติดต่อลูกค้าของตัวเองเลย แล้วพยายาม software จบให้เร็วที่สุด

-A Complex System that cannot be properly modularized: ถ้ามีระบบใหญ่ๆ แล้วไม่สามารถแบ่งเป็น module ได้ เราไม่สามารถใช้วิธีนี้ได้เลย

-High performance is an issue: RAD ในการสร้างของมัน ไม่มี architecture ไม่ได้พูดถึง software architecture ไม่ได้พูดถึง technical ไม่ได้พูดถึง design pattern คือแบ่งทีมแล้วทำให้จบ แต่ปัญหาก็คือ เนื่องจาก architecture ไม่มี ไม่ถูกต้อง จึงเกิดปัญหานี้

-Heavy use of new technology ถ้ามีการใช้เทคโนโลยีใหม่ๆ เข้ามาเยอะๆ มันจะมีปัญหากับ RAD

**ข้อดี** RAD model ที่เราน่าจะเอาไปใช้คือ การแบ่งทีมออกเป็นหลายทีม สามารถจะ modularize module ต่างๆ ออกจากันได้ แต่ไม่สามารถใช้ได้กับทุก application ที่เรา develop ถ้าเราแบ่ง module ไม่ได้ RAD ก็ไม่เกิด

RAD เป็น model อีกอันที่ว่า ถ้าซอร์ฟแวร์ค่อนข้างจะขนาดใหญ่ model นี้ไม่เหมาะในการพัฒนา

คล้าย ๆ กับ incremental แต่ cycle จะสั้นกว่า และ

- ใช้เวลาประมาณ 60-90 วัน สำหรับ
  - การกำหนดขนาดและ
  - กำหนดเกณฑ์วัดต่าง ๆ
  - โดยแบ่งเป็น section แล้วทำเป็น Modeling จากนั้นก็มา deploy แล้วค่อยมารวมกันได้
  - จะสามารถทำได้ก็ต่อเมื่อ business สามารถแบ่งออกเป็น component ได้
  - ถ้ามีหลาย Project ก็ต้องมีหลาย team และก็ต้องมี activities มากขึ้นระหว่าง user และ ทีมพัฒนา

**Spiral Model**

Incremental คือ ทำ software มารอบหนึ่ง เสร็จแล้ว ก็ค่อยขยาย แต่ Iterative process ซึ่งเป็นลักษณะของการทำซ้ำ หมายถึง เริ่มกระบวนการ

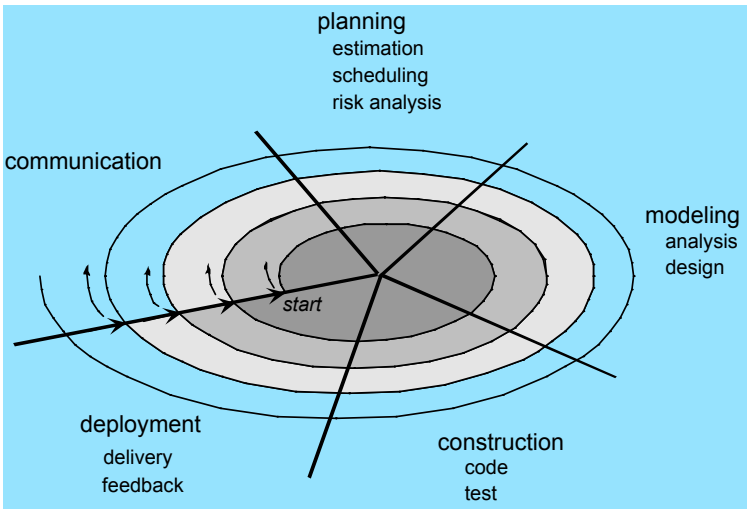
Communication -> Planning -> Modeling -> Construction -> Deployment

จากนั้นก็เริ่มทำซ้ำ วนไปเรื่อย ๆ ในขณะเดียวกันก็ increment ไปด้วย ซึ่งก็จะมีการขยายไปด้วยในกรณีที่ต้องขยาย โดยใช้เทคนิค Prototyping

- เป็น risk-driven process mode

  อันไหนที่เป็น risk ก็เอามาจัดการก่อน เพื่อให้ risk นั้นหมดไป โดย risk ที่สำคัญได้แก่

- requirement ที่คลุมเครือ ไม่ชัดเจน
- technical ต่าง ๆ ที่ยังไม่เคยใช้ ไม่เคยทดลอง ก็จะทำให้เกิด risk ได้




Component-Based Development (CBD)

บางครั้งเรียกว่า Component base software engineering

เป็นการนำ Component ต่าง ๆ ที่มีอยู่มาใช้รวมกันเพื่อพัฒนาโปรแกรม โดยเน้นในลักษณะของการ adap ในการใช้งาน

เราจะพิจารณาถึง COTS ( Commercial off-the shelf) หมายถึง หลาย ๆ สิ่งที่เป็น Component ที่เขามีอยู่แล้ว นำมาใช้งาน โดยไม่ต้องลงทุนสร้างเอง โดยนำ Component ต่าง ๆ มาประกอบกัน แล้วมองเป็น Software package ซึ่งเราเองก็จะอยู่ในกรอบของ Software เหล่านี้แหละ ถ้า customize เงื่อนไข business เข้ากับ component ได้ก็ไม่มีปัญหา แต่ในความจริง business จะมีความต้องการและเปลี่ยนแปลง ดังนั้น Component ที่ใช้อยู่อาจจะไม่สนองตอบต่อความต้องการที่เปลี่ยนไปได้ทั้งหมด

*** Development Model

- ต้องทำความรู้จักและประเมินค่า component ที่ต้องการนำมาใช้งาน โดยอาจจะปรึกษากับบริษัทที่ให้คำปรึกษาเกี่ยวกับ Component ต่าง ๆ ได้
- ดู feature ว่า เข้ากับ environment เราหรือไม่ ถ้าไม่เข้าก็ต้อง list issue เราขึ้นมาว่ากระทบหรือมีปัญหาอะไรบ้างหรือไม่
- จัดเตรียม architecture
- รวม component เข้าไปใน architecture ที่ได้จัดเตรียมไว้
- ทดสอบผลที่รวม component ทั้งหมด

จะสังเกตเห็นว่าแทบจะไม่มีการ coding เลย เพียงแต่จะเห็นแค่การประกอบกันเท่านั้น

จากอาจารย์

- best of the breed หมายถึง การเลือกเฉพาะ component ที่ดีที่สุดมารวมกัน
- suited หมายถึง ซื้อ 1 software ที่รวม component ต่าง ๆให้เราแล้ว เช่น Microsoft

**ข้อดี**

- Software Reusability
  - ไม่ต้องเขียนโปรแกรม ตั้งแต่เริ่มต้น
  - ความคงทนและ stable ของ Component library ที่สร้างมาจาก vendor มากกว่าเราสร้างเอง อาจจะมี bug มากกว่า ถึงแม้ราคาจะถูกกว่าก็จริง รวมทั้ง Component ต่าง ๆ ที่มีในท้องตลาด ก็มีคนใช้แล้ว ดังนั้น น่าจะดีกว่า
  - มี process น้อยลง
  - Software ที่เราจะเลือกใช้นั้น จะมี specialist อยู่แล้ว เมื่อเราซื้อมาใช้ก็น่าจะได้รับผลดีและได้ component ที่มีประสิทธิภาพ
  - เป็นมาตรฐาน เนื่องจาก component ที่เขาสร้างมานั้นจะมีการสร้างที่ตรงตาม requirement โดยทั่วไปอยู่แล้ว
  - เร่งให้มีการพัฒนา
  - ประหยัดเวลาในการพัฒนาโปรแกรมได้ 70%

**ข้อเสีย**

- Increased maintenance cost เมื่อใช้ไปซักพัก ก็จะเสียค่าใช้จ่ายในการ maintenance มากขึ้น เนื่องจาก เราเองไม่ทราบรายละเอียด code ที่มีความ complex ของแต่ละ component ดังนั้นต้องมีการจัดหาคนที่จะสามารถจัดการ code เหล่านี้ได้
- Lack of tool support เครื่องมือในการ support ในการเพิ่มเติม แก้ไข component ไม่มี
- Not-invented-here syndrome บางตัวไม่สามารถสร้าง component ใหม่ได้เลย และไม่ทราบว่า ข้างในมีรายละเอียดอะไรบ้าง
- Creating and maintaining a component library เมื่อมี component แล้ว ถ้าหากต้องการ reuse จะมีวิธีอย่างไร ซึ่งยากต่อการ catalog ว่าประกอบด้วยอะไรบ้าง
- Finding understanding and adapting reusable component การ adap เปลี่ยนแปลง , modify นั้นยาก และถ้ามีการเปลี่ยนแปลง component บางตัว อาจจะส่งผลกระทบต่อ component ตัวอื่นได้ด้วย
- Long term commitment ➔ component บางตัวจะ log ไว้ที่ framework ของมัน จนในที่สุดก็จะไม่สามารถสร้างหรือพัฒนาอะไรต่อได้เลย และ component บางตัวนั้นจะมีอายุยืนยาวได้ซักแค่ไหนกันเชียว

**สัปดาห์ที่** 6 – 7
**ปฏิบัติงานกลุ่มและนำเสนองานกลุ่ม ส่วนที่** 1

**สัปดาห์ที่** 10
**การออกแบบ** Output

.....

**หลักของการดีไซน์เอาต์พุต**

เอาต์พุตคือ ข้อมูลที่ถูกส่งมอบให้กับผู้ใช้ระบบ โดยระบบงานข้อมูล (Information System) ซึ่งข้อมูลเหล่านี้ บางครั้งก็ต้องผ่านกระบวนการต่างๆ มากมายภายในระบบงานเสียก่อนที่จะถูกส่งออกมาให้กับผู้ใช้ระบบหรือในบางครั้ง ข้อมูลบางประเภทก็อาจจะไม่จำเป็นต้องผ่านกระบวนการภายในระบบเลย หรือหากมีก็น้อยมากซึ่งก็อาจเป็นไปได้

เอาต์พุตสำหรับระบบงานคอมพิวเตอร์ในปัจจุบันมีหลายรูปแบบ ซึ่งอาจแบ่งได้เป็น 2 ลักษณะ คือแบบฮาร์ดก๊อปปี้ (Hard Copy) ซึ่งก็ได้แก่รายงานต่างๆ ที่ออกมาทางเครื่องพิมพ์ และแบบซอฟต์ก๊อปปี้ (Sofy Copy) ซึ่งมักหมายถึงข้อมูลที่แสดงผลออกทางจอภาพชนิดต่างๆ และไมโครฟอร์ม (Microform) เป็นต้น

เนื่องจากเอาต์พุตเป็นสิ่งสำคัญที่จะเรียกการยอมรับหรือเรียกคะแนนนิยมให้กับระบบงานที่นักวิเคราะห์กำลังพัฒนาและดีไซน์อยู่ นักวิเคราะห์ระบบจึงควรทราบหลักการสำคัญ 6 ข้อในการดีไซน์เอาต์พุต ดังนี้

1. ดีไซน์เอาต์พุต เพื่อตอบสนองต่อความต้องการของผู้ใช้ระบบ
2. ดีไซน์เอาต์พุต ให้เหมาะสมต่อผู้ใช้ระบบ
3. ส่งมอบเอาต์พุตตามจำนวนที่ผู้ใช้ระบบต้องการ
4. ให้แน่ใจว่าเอาต์พุตได้อยู่ในตำแหน่งที่เหมาะสม
5. เอาต์พุตถูกส่งมอบให้กับผู้ใช้ระบบตามเวลาที่กำหนด
6. เลือกวิธีการที่เหมาะสมสำหรับเอาต์พุตแต่ละแบบ

**1. ดีไซน์เอาต์พุตเพื่อตอบสนองต่อความต้องการของผู้ใช้ระบบ**

ทุกครั้งที่ระบบจะต้องออกเอาต์พุต ไม่ว่าจะเป็นการออกรายงานต่างๆ ทางเครื่องพิมพ์ หรือการแสดงข้อมูลออกทางจอภาพ (CRT) นักวิเคราะห์ระบบควรทราบว่า ทุกเอาต์พุตที่กล่าวมานั้นจะต้องมีจุดประสงค์ของมันเองอยู่ โดยจุดประสงค์ของการออกรายงานหรือเอาต์พุตนั้นเกิดขึ้นเพื่อตอบสนองต่อความต้องการของผู้ใช้ระบบในแง่ใดแง่หนึ่ง

ดังนั้น การดีไซน์เอาต์พุตจึงจะต้องคำนึงถึงจุดประสงค์และความต้องการของผู้ใช้ระบบ ไม่ใช่เอาความสวยงามของการจัดรายงานหรือจอภาพเป็นหลัก หากว่าการดีไซน์เอาต์พุตไม่สามารถครอบคลุมวัตถุประสงค์และความต้องการของผู้ใช้ระบบได้ นักวิเคราะห์ระบบก็ไม่ควรจะดันทุรังทำต่อ เพราะเป็นการสิ้นเปลืองทั้งเวลาและแรงงานโดยใช่เหตุ

**2. ดีไซน์เอาต์พุตให้เหมาะสมต่อผู้ใช้ระบบ**

ในลักษณะที่ระบบงานคอมพิวเตอร์ที่นักวิเคราะห์ระบบกำลังพัฒนาเป็นระบบที่ใหญ่ มีผู้ใช้ระบบอยู่หลายระดับหรือจำนวนมาก ซึ่งแต่ละคนอาจมีความต้องการเอาต์พุตที่ไม่เหมือนกัน ดังนั้นจึงเป็นการยากที่จะดีไซน์เอาต์พุตออกมาให้ตรงกับความต้องการของผู้ใช้ระบบทุกคนได้ แต่นั่นก็ไม่ใช่ว่า "ยาก" จะเท่ากับคำว่า "เป็นไปไม่ได้" เพียงแต่นักวิเคราะห์อาจจะต้องเพิ่มจำนวนเอาต์พุตให้เท่ากับจำนวนความต้องการที่แตกต่างออกไปของผู้ใช้ระบบแต่ละคนนั่นเอง ซึ่งอาจจะมีเอาต์พุตหรือรายงานบางอย่างที่มีลักษณะคล้ายกันเกิดขึ้น

ดังนั้น เพื่อที่จะลดลักษณะของเอาต์พุตที่คล้ายคลึงกัน ซึ่งในบางครั้งอาจจะเป็นการสิ้นเปลืองค่าใช้จ่ายเนื่องจากความซ้ำซ้อนกันของเอาต์พุต นักวิเคราะห์ควรทำหน้าที่ไกล่เกลี่ยและหาข้อยุติของเอาต์พุตกับกลุ่มผู้ใช้ระบบที่สามารถใช้เอาต์พุตร่วมกันได้ โดยข้อยุติของเอาต์พุตนั้นๆ จะต้องให้ความเหมาะสมต่อกลุ่มผู้ใช้ทั้งหมดด้วย

**3. ส่งมอบเอาต์พุตตามจำนวนที่ผู้ใช้ระบบต้องการ**

งานที่นักวิเคราะห์ระบบจะต้องทำก็คือ การดีไซน์เอาต์พุตให้มีจำนวนที่เหมาะสมและคล้องจองกันกับผู้ใช้ระบบ สิ่งนี้ดูๆ ไปก็ไม่ยากนัก แต่เวลาปฏิบัติจริงแล้วมักไม่เป็นเช่นนั้นเนื่องจากความต้องการของผู้ใช้ระบบเกิดขึ้นมาอย่างต่อเนื่อง ซึ่งจะส่งผลทำให้เอาต์พุตเกิดการขยายตัวสะสมมากขึ้นๆ เช่นกัน และในที่สุดก็จะเกิดปรากฏการณ์ที่เรียกว่า Information Overload ซึ่งหมายถึงระบบงานส่งมอบเอาต์พุตมากเกินกว่าความจำเป็นหรือเกินกว่าความสามารถที่ผู้ใช้ระบบจะใช้ได้หมดนั่นเอง

**4. ให้แน่ใจว่าเอาต์พุตได้อยู่ในตำแหน่งที่เหมาะสม**

ในระบบงานคอมพิวเตอร์นั้นเอาต์พุตมีอยู่ด้วยกันหลายรูปแบบ เช่น พิมพ์ออกเป็นรายงานบนกระดาษด้วยเครื่องพิมพ์ (Printer) แสดงผลออกทางจอภาพ (CRT) และเก็บอยู่ในรูปของไมโครฟอร์ม (microform) หรือแม้กระทั่งในรูปของเสียงที่ออกจากทางลำโพงที่ต่อกับเครื่องคอมพิวเตอร์ได้

โดยทั่วไป เอาต์พุตจะถูกสร้างขึ้น ณ ที่ใดที่หนึ่ง เช่น หน่วยงานประมวลผล (Data Processing department) แล้วค่อยถูกกระจายส่งต่อไปยังแผนกหรือหน่วยงานต่างๆ ซึ่งเป็นผู้ใช้ระบบอีกทอดหนึ่ง แม้ว่าในปัจจุบันการประมวลผลแบบ On-Line จะช่วยลดปัญหาการกระจายเอาต์พุตออกไปได้บ้าง แต่อย่างไรก็ตาม การกระจายเอาต์พุตไปยังผู้ใช้ระบบที่ถูกต้องเหมาะสม เป็นสิ่งที่นักวิเคราะห์ระบบจะต้องคำนึงถึง เพราะว่าไม่ว่าเอาต์พุตหรือรายงานต่างๆ ที่นักวิเคราะห์ระบบจะดีไซน์นั้นดีเพียงใด หากไม่ได้อยู่ในมือของบุคคลหรือผู้ใช้ระบบที่เหมาะสมกับมัน รายงานหรือเอาต์พุตนั้นก็ย่อมไม่เกิดคุณค่าอะไรเลย

**5. เอาต์พุตถูกส่งมอบให้กับผู้ใช้ระบบตามเวลา**

หนึ่งในสาเหตุสำคัญที่ระบบงานนั้นไม่ได้รับการยอมรับจากผู้ใช้ระบบก็คือ ระบบงานไม่สามารถจะให้ข้อมูลหรือเอาต์พุตกับผู้ใช้ระบบได้ตามเวลาอย่างทันท่วงทีที่เขาต้องการ เวลาจึงเป็นสิ่งสำคัญต่อเอาต์พุตอย่างมาก โดยเฉพาะผู้บริหารที่ต้องการข้อมูลจากเอาต์พุตเพื่อมาใช้ในการตัดสินใจ

**6. เลือกวิธีการที่เหมาะสมสำหรับเอาต์พุตแต่ละแบบ**

เอาต์พุตมีได้หลายรูปแบบ เช่น ออกทางเครื่องพิมพ์ ออกทางจอภาพ หรือไมโครฟอร์ม ฯลฯ การเลือกวิธีการออกแบบเอาต์พุตที่เหมาะสมจึงเป็นสิ่งที่นักวิเคราะห์ระบบจะต้องคำนึงถึงด้วย

การเลือกวิธีการที่เหมาะสมสำหรับเอาต์พุตจะยังผลให้ต้นทุนหรือค่าใช้จ่ายของระบบงานแตกต่างออกไปเช่นกัน ตัวอย่างเช่น ต้นทุนของการออกรายงานทางเครื่องพิมพ์สูงกว่าการแสดงผลทางจอภาพ เนื่องจากต้นทุนของกระดาษที่ใช้ในการพิมพ์เริ่มสูงขึ้นทุกขณะ

ในสหรัฐอเมริกา เริ่มมีแนวความคิดของการพัฒนาระบบงานในลักษณะที่เรียกว่า Paperless System ซึ่งหากแปลกันเป็นภาษาไทยคงจะได้คำแปลกๆ ว่า "ระบบงานไร้กระดาษ" ซึ่งระบบงานชนิดนี้จะพยายามให้เอาต์พุตที่ได้เก็บอยู่ในลักษณะอื่น โดยพยายามให้สิ้นเปลืองกระดาษน้อยที่สุด เช่น ให้เอาต์พุตเก็บอยู่ในแฟ้มข้อมูลที่เก็บอยู่ในออปติคัลดิสก์หรือฮาร์ดดิสก์แทน ซึ่งเมื่อผู้ใช้ระบบต้องการใช้ ก็จะดึงข้อมูลนั้นออกมาดูทางจอภาพแทน เป็นต้น

**การเลือกใช้รูปแบบที่เหมาะสมสำหรับเอาต์พุต**

เอาต์พุตที่เกิดขึ้นในระบบธุรกิจจะต้องได้รับการจัดการที่เหมาะสม โดยทั่วไปเอาต์พุตสามารถแบ่งออกเป็น 2 ประเภทคือ เอาต์พุตที่ใช้ภายในธุรกิจ (Internal Output) และเอาต์พุตที่ใช้ภายนอกธุรกิจ (External Output)

เอาต์พุตที่ใช้ภายในธุรกิจได้แก่ รายงานต่างๆ ที่ใช้ภายในแผนกหรือหน่วยงาน เช่น รายงานการขายประจำวัน รายงานสรุปผลการขายประจำเดือน หรือรายงานเวลาการทำงานของพนักงาน เป็นต้น ส่วนเอาต์พุตที่ใช้ภายนอก ได้แก่ เอาต์พุตที่จะต้องถูกส่งออกไปยังบุคคลภายนอกธุรกิจ เช่น ใบกำกับสินค้า ใบเสร็จรับเงิน แบบฟอร์มการเสียภาษีของสรรพากร เป็นต้น

วิธีการดีไซน์เอาต์พุตที่ใช้ภายนอกอาจจะต้องแตกต่างกับเอาต์พุตที่ใช้ภายใน เช่น อาจจะต้องมีการอธิบายถึงความหมายของแต่ละช่องที่กรอกลงไปให้ละเอียดขึ้น การจัดรูปแบบอาจจะต้องกำหนดตามมาตรฐานสากลหรือตามที่กฎหมายบังคับไว้ เป็นต้น

วิธีการดีไซน์เอาต์พุตมีส่วนสัมพันธ์กับรูปแบบของเอาต์พุตที่ออกมาอย่างมาก เช่น ประเทศไทยเอาต์พุตที่ใช้ภายนอกหรือที่ธุรกิจจะต้องให้กับหน่วยงานราชการทั้งหมดยังคงอยู่ในรูปแบบที่จะต้องพิมพ์ลงกระดาษทั้งสิ้น ดังนั้นรูปแบบของเอาต์พุตที่ใช้ย่อมต้องเลือกวิธีการพิมพ์ลงเครื่องพิมพ์อย่างเดียว เป็นต้น ดังตารางต่อไปนี้ได้แสดงให้เห็นถึงข้อดีข้อเสียของการใช้รูปแบบต่างๆ เพื่อออกเอาต์พุต

| รูปแบบที่ใช้ | ข้อดี | ข้อเสีย |
|---|---|---|
| **เครื่องพิมพ์** | เป็นที่ยอมรับกันโดยทั่วไปในธุรกิจมีความยืดหยุ่นและความสามารถในการรองรับข้อมูลใหญ่ๆ ได้ดี | มักจะก่อให้เกิดเสียงรบกวน โดยเฉพาะเมื่อใช้กับเครื่องพิมพ์แบบดอตแมทริกซ์ โดยทั่วไปมักให้ผลลัพธ์ช้าและต้นทุนของเครื่องพิมพ์ |

| | | |
|---|---|---|
| | มีความเชื่อถือได้สูง และตรวจสอบได้ง่าย | ค่อนข้างสูง แต่ปัจจุบันมีเครื่องพิมพ์แบบ Laser ซึ่งให้งานพิมพ์รวดเร็ว ไม่มีเสียงรบกวน |
| **จอภาพ** | เงียบและมีความสามารถที่จะทำงานแบบออนไลน์ และส่งผ่านเอาต์พุตไปในระบบเครือข่ายได้ดี สามารถที่จะใช้ความสามารถของคอมพิวเตอร์ได้อย่างเต็มที่ในการที่จะเข้าถึงแฟ้มข้อมูลหรือฐานข้อมูลต่างๆ ได้อย่างรวดเร็ว และบ่อยครั้งตามต้องการ | ยังคงต้องการระบบเดินสายที่ดีโดยเฉพาะในระบบเครือข่าย<br><br>ไม่สามารถตอบสนองต่อความต้องการบางลักษณะ โดยเฉพาะเอาต์พุตที่ใช้ภายนอก ที่ยังคงต้องการให้มีการพิมพ์ออกมาทางเครื่องพิมพ์อยู่ดี |
| **เสียง** | เหมาะสำหรับระบบงานที่ต้องการส่งข้อความที่จะส่งให้กับผู้ใช้ระบบและให้ผู้ใช้ระบบตอบสนองหรือกระทำต่อระบบทันที<br><br>เหมาะสำหรับผู้ใช้ระบบที่ต้องการความอิสระของการใช้มือ (Hands Free) เพื่อการปฏิบัติงานอย่างอื่น | ในปัจจุบันเทคโนโลยีทางด้านนี้ยังค่อนข้างแพงอยู่ดี<br><br>ระดับของเสียงอาจไปรบกวนคนอื่นที่อยู่ในพื้นที่เดียวกัน<br><br>การใช้วิธีนี้ค่อนข้างจะจำกัดอยู่มาก ไม่ใช่สามารถใช้กันได้โดยทั่วไป |
| **ไมโครฟอร์ม** | ประหยัดเนื้อที่ในการเก็บอย่างมาก มีความคงทนต่อการใช้งาน สามารถใช้ได้บ่อยครั้งและค้นหาข้อมูลได้สะดวก | ต้องการซอฟต์แวร์พิเศษสำหรับการใช้ไมโครฟอร์มโดยเฉพาะ<br><br>ต้องการอุปกรณ์พิเศษอีกเช่นกันหากต้องการสำเนา (Copy) ออกทางเครื่องพิมพ์<br><br>ค่าใช้จ่ายตอนตั้งต้นจะค่อนข้างสูง |

**การดีไซน์การพิมพ์รายงาน**

เนื่องจากการพิมพ์รายงานออกทางเครื่องพิมพ์ ยังคงเป็นที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน ซึ่งรายงานต่างๆ นักวิเคราะห์ระบบจะต้องดีไซน์ให้รายงานสามารถแสดงข้อมูลได้อย่างเพียงพอ และจัดให้เหมาะสมกับที่ผู้ใช้ระบบต้องการ

**พื้นฐานของการรายงาน**

มีลักษณะพื้นฐานของการรายงานที่นักวิเคราะห์พึงทราบเป็นอย่างดีคือ ประเภทของข้อมูลที่ต้องแสดงในตำแหน่งใดตำแหน่งหนึ่งนั้นเป็นประเภทอะไร เช่น ตัวเลข หรือตัวอักษร และจะต้องใช้ความกว้างเท่ากับที่ตัวอักษรจึงจะเพียงพอ ในพื้นฐานของการรายงาน ข้อมูลที่เราจะแสดงในรายงานสามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ ด้วยกัน คือ

1. **ข้อมูลที่เป็นค่าคงที่** (Constant Information) หมายถึงข้อมูลที่จะต้องออกมาเหมือนกันทุกครั้งที่มีการพิมพ์รายงาน โดยปกตินักวิเคราะห์ระบบจะทำการดีไซน์รายงานลงในแบบฟอร์มที่ใช้สำหรับการดีไซน์รายงานซึ่งเรียกว่า Report Layout Form โดยข้อมูลที่เป็นค่าคงที่นั้น นักวิเคราะห์จะระบุด้วยซึ่งทุกครั้งที่ออกรายงานก็จะพิมพ์ข้อความที่เหมือนกันเช่นนี้โดยตลอด

2. **ข้อมูลที่เป็นตัวแปร** (Variable Information) หมายถึงข้อมูลที่อาจจะเปลี่ยนแปลงไปในแต่ละครั้งที่มีการพิมพ์รายงาน ตัวอย่างเช่น ยอดขาย (Sales) กำไรขั้นต้น (Gross Profit) เป็นต้น หลักที่ปฏิบัติกันโดยทั่วไปที่ใช้ระบบประเภทของตัวแปรในเลย์เอาต์ฟอร์มก็คือ เรามักจะใช้ X แทนตัวแปรที่เป็นตัวอักษร (Character/String) และ 9 แทนตัวแปรที่เป็นตัวเลข ตัวอย่างเช่น หากต้องการพิมพ์ข้อมูลที่เป็นตัวแปรสำหรับลูกค้ารายหนึ่ง ซึ่งประกอบไปด้วยรหัสลูกค้า ชื่อลูกค้า และยอดสั่งซื้อต่อปีตามลำดับ โดยรหัสลูกค้าจะเป็นตัวอักษร 10 ตำแหน่ง และชื่อลูกค้าจะเป็นตัวอักษร 20 ตำแหน่ง และยอดสั่งซื้อจะเป็นตัวเลขหลักล้านที่มีจุดทศนิยม 2 ตำแหน่ง

**การคำนวณความกว้างของรายงาน** เนื่องจากเครื่องพิมพ์แต่ละชนิดมีขีดความสามารถในการพิมพ์รายงานไม่เหมือนกัน บางเครื่องสามารถพิมพ์รายงานได้มากกว่า 200 ตัวอักษร ในขณะที่บางเครื่องพิมพ์ได้ไม่ถึง 130 ตัวอักษร ดังนั้น เมื่อจะทำการดีไซน์รายงานทุกครั้ง นักวิเคราะห์ก็ควรที่จะทำการคำนวณโดยคร่าวๆ ว่าข้อมูลที่อยากจะให้พิมพ์ออกมานั้น พอดีกับความกว้างของหน้ากระดาษหรือไม่ และเครื่องพิมพ์สามารถพิมพ์ได้หรือ

**ลำดับขั้นการดีไซน์รายงาน** มี 11 ขั้นตามลำดับดังนี้

1. พิจารณาถึงวัตถุประสงค์ของรายงาน
2. พิจารณาว่า ใครเป็นผู้ใช้รายงาน
3. พิจารณาว่า มีข้อมูลอะไรบ้างที่จะต้องแสดงหรือพิมพ์ในรายงาน

4. นับจำนวนช่องว่างและความกว้างของข้อมูลในฟิลด์เพื่อนำมาพิจารณาถึงขนาดของรายงานที่จะพิมพ์

5. ตั้งชื่อรายงาน

6. รายงานควรจะต้องมีการพิมพ์หมายเลขหน้าไว้เสมอ

7. ควรจะแสดงวันที่ที่พิมพ์รายงานไว้ในตัวรายงานด้วย

8. สำหรับหัวข้อรายงานในแต่ละแถว ควรใช้คำพูดที่ชัดเจน

9. ในแบบฟอร์มที่ใช้สำหรับการดีไซน์รายงาน ควรระบุชนิดของข้อมูลว่าเป็นตัวเลขหรือตัวอักษรให้ชัดเจน

10. ระบุตำแหน่งที่ใช้สำหรับพิมพ์ข้อความสรุปรายงาน เช่น ตำแหน่งต่างๆ ในบรรทัดของยอดรวมต่างๆ ในรายงาน

11. นำตัวอย่างที่ได้ออกแบบมาให้กับผู้ใช้รายงาน เพื่อตรวจดูอีกครั้ง เพื่อความถูกต้องว่ารายงานได้รับการดีไซน์ตรงตามวัตถุประสงค์ดีแล้วก่อนนำแบบไปเขียนโปรแกรมจริง

**การดีไซน์เอาต์พุตทางจอภาพ**

การดีไซน์เอาต์พุตที่ออกทางจอภาพ (Disigning Screen Output) จะมีข้อแตกต่างกับเอาต์พุตที่ออกจาเครื่องพิมพ์อยู่หลายจุดด้วยกันคือ สำหรับจอภาพ (Screen) นั้น ข้อมูลที่แสดงออกมานั้นจะไม่ติดตายตัว (Permanect) เหมือนกับการใช้เครื่องพิมพ์พิมพ์รายงาน ลักษณะของเอาต์พุตจึงมีความยืดหยุ่นค่อนข้างมาก แต่ก็มีข้อจำกัดคือการออกเอาต์พุตทางจอภาพเหมาะสำหรับผู้ใช้ระบบที่ต้องมีจอภาพด้วย เราไม่สามารถจะแจกจ่ายเอาต์พุตให้กับผู้ใช้อื่นที่ขาดอุปกรณ์ดังกล่าวได้

การที่กล่าวว่าจอภาพมีความยืดหยุ่นในการแสดงข้อมูลสูงกว่าเครื่องพิมพ์ จะหมายถึงความสามารถที่จะลบข้อมูลออกไปแล้วใส่ข้อมูลใหม่ได้ ความสามารถในการที่เลื่อนข้อมูลขึ้นลง (Scroll) หรือเลื่อนไปทางซ้ายหรือทางขวา หรือการเก็บตำแหน่งของข้อมูลบนจอภาพ (Save Screen) ไว้ชั่วคราว แล้วดึงออกมาแสดงใหม่ (Restore Screen) ในภายหลัง เป็นต้น ซึ่งสิ่งเหล่านี้เป็นความสามารถเฉพาะตัวของการแสดงผลทางจอภาพเลยทีเดียว

จากการที่จอภาพมีความสามารถในการทำงานหลายด้าน จึงจำเป็นอยู่เองที่นักวิเคราะห์ระบบจะต้องทำให้ผู้ใช้ระบบสามารถที่จะควบคุมจอภาพบางประการตามที่เขาต้องการได้ เช่น หากผู้ใช้ต้องการพิมพ์ข้อมูลออกทางจอภาพ ระบบงานอาจจะต้องพิมพ์ออกทีละ 24 บรรทัด และในบรรทัดที่ 25 อาจจะต้องถามผู้ใช้ระบบเสียก่อนว่า "กรุณากด (C) = ดำเนินการกด หรือกด (S) = หยุดงาน" ลักษณะเช่นนี้เป็นตัวอย่างที่ผู้ใช้ระบบมีความสามารถที่จะควบคุมจอภาพให้พิมพ์รายงานต่อออกทางจอภาพเมื่อกด (C) และเมื่อต้องการจะหยุด ก็สามารถทำได้โดยการกด (S)

**ข้อแนะนำในการดีไซน์จอภาพ**

ข้อแนะนำเกี่ยวกับการดีไซน์จอภาพอยู่ 4 หัวข้อ โดยสรุปจะได้ดังนี้

1. พยายามให้การแสดงข้อมูลบนจอภาพดูเรียบง่าย ไม่ซับซ้อน

2. พยายามให้การแสดงบนจอภาพมีมาตรฐานแบบเดียวกัน เพื่อให้ผู้ใช้เกิดความคุ้นเคยได้เร็ว

3. สำหรับข้อมูลบางอย่างที่ต้องการจะเน้นให้เห็นถึงความแตกต่าง ให้ใช้สีที่แตกต่างออกไปจากปกติ เพื่อดึงดูดความสนใจของผู้ใช้

4. ให้การโต้ตอบระหว่างผู้ใช้ระบบกับจอภาพเป็นไปโดยธรรมชาติมากที่สุด เช่น การเลื่อนเคอร์เซอร์ (Cursor Movement) ควรจะเลื่อนจากบนลงล่างหรือจากซ้ายมาขวา ซึ่งเป็นไปตามธรรมชาติและมาตรฐานสากล เป็นต้น

**การดีไซน์ลงแบบฟอร์มสำหรับดีไซน์จอภาพ** (Screen Layout)

มีลักษณะเช่นเดียวกับการดีไซน์ลงในแบบฟอร์มสำหรับรายงาน (Report Layout) แบบฟอร์มสำหรับดีไซน์จอภาพจะมีขนาดเล็กกว่า เนื่องมาจากโดยปกติจอภาพทั่วไปจะมีความกว้าง 80 ตัวอักษร และมีความยาวได้เท่ากับ 25 บรรทัด ซึ่งโดยปกติ เราจะพบว่าแบบฟอร์มสำหรับการดีไซน์จอภาพจอให้เรากรอกได้ 24 บรรทัด เหตุที่เป็นเช่นนี้มักจะเกิดจากการที่ในเครื่องคอมพิวเตอร์บางชนิดจะสงวนบรรทัดไว้ 1 บรรทัดสำหรับแสดงผลเฉพาะระบบปฏิบัติการเท่านั้น ดังนั้น บรรทัดในการดีไซน์จึงได้หายไป จากเหตุผลดังกล่าวจึงทำให้โปรแกรมเมอร์และนักวิเคราะห์ระบบเองพยายามหลีกเลี่ยงการใช้บรรทัดที่ 1 หรือบรรทัดที่ 25 อย่างใดอย่างหนึ่ง เพื่อลดปัญหาการแสดงข้อมูลที่อาจจะไปทับกับข้อความที่แสดงโดยระบบปฏิบัติการดังที่กล่าวมาแล้ว

ในการดีไซน์จอภาพลงในฟอร์มนั้น เรายังคงยึดวิธีเดียวกันกับการดีไซน์รายงาน นั่นก็คือเราจะใช้ตัวอักษร X แทนข้อมูลตัวแปรที่เป็นตัวอักษร และ 9 แทนข้อมูลตัวแปรที่เป็นตัวเลข ส่วนข้อมูลใดที่เป็นข้อมูลตายตัว (Fixed) หรือเป็นค่าคงที่ (Constant) เราก็จะเขียนเข้าไปโดยตรงในแบบฟอร์มนั้นเลย

**หลักของการดีไซน์อินพุต**

แม้ว่าโดยทั่วไป นักวิเคราะห์ระบบจะเน้นหนักถึงความสำคัญของเอาต์พุตมาก เนื่องเพราะเอาต์พุตของระบบงานถือว่าเป็นผลลัพธ์อันสำคัญในอันที่จะตอบสนองต่อความต้องการของผู้ใช้ระบบ

อย่างไรก็ดี สิ่งหนึ่งที่นักวิเคราะห์ระบบจะลืมเสียไม่ได้ก็คือ หากอินพุตที่เข้ามาในระบบไม่ดีเพียงพอหรือเกิดความผิดพลาดได้ง่าย ย่อมต้องส่งผลทำให้เอาต์พุตที่ได้ออกมาจากระบบพลอยเสียหายไปด้วย ดังนั้น การดีไซน์อินพุตที่ดีจึงเป็นสิ่งสำคัญอีกสิ่งหนึ่งที่จะทำให้ระบบงานคอมพิวเตอร์มีความสมบูรณ์ยิ่งขึ้น

**การดีไซน์แบบฟอร์มสำหรับกรอกข้อมูลที่เป็นอินพุต**

แม้ว่าโดยทั่วไปธุรกิจจะได้มีการดีไซน์แบบฟอร์มของธุรกิจไว้เพื่อใช้ปฏิบัติงานอยู่แล้ว นักวิเคราะห์ระบบก็ควรจะมีความสามารถทางด้านนี้ด้วยไม่แพ้กัน นอกจากนี้นักวิเคราะห์ระบบจะต้องสามารถรู้ด้วยตนเองว่า แบบฟอร์มแบบไหนที่ยังไม่ดีเพียงพอและจะต้องปรับปรุง และแบบฟอร์มแบบไหนที่ควรจะยกเลิกหรือยุบมารวมกัน เพื่อให้เกิดประโยชน์สูงสุด

แบบฟอร์มต่างๆ ที่ใช้อยู่ในธุรกิจ เป็นเครื่องมืออย่างหนึ่งที่จะทำให้ธุรกิจดำเนินต่อไป ซึ่งแบบฟอร์มเหล่านี้ โดยปกติมักจะถูกดีไซน์และตีพิมพ์ออกมาไว้ก่อน เมื่อต้องการจะใช้ ผู้ใช้ก็จะเขียนข้อความอันเป็นข้อมูลลงในแบบฟอร์มเป็นเบื้องต้น ซึ่งต่อจากนั้น แบบฟอร์มต่างๆ จึงค่อยถูกนำมาบันทึกลงในระบบคอมพิวเตอร์ ในลักษณะที่ธุรกิจดำเนินการด้วยระบบงานคอมพิวเตอร์ แบบฟอร์มต่างๆ โดยส่วนใหญ่จึงถือเป็นต้นกำเนิดของข้อมูลที่จะนำมากรอกเข้าสู่ระบบงานโดยพนักงานคีย์ข้อมูล (Data Entry Personnel)

**หลักสำคัญที่ใช้ในการดีไซน์แบบฟอร์ม มีอยู่ด้วยกัน 4 หัวข้อ คือ**

1. **แบบฟอร์มควรมีลักษณะที่ง่ายต่อการกรอก** จะทำให้ลดข้อผิดพลาดในการกรอกข้อมูล และในขณะเดียวกันก็ลดเวลาในการกรอกลงไปด้วย นักวิเคราะห์ไม่ควรลืมว่าต้นทุนของกระดาษนั้นถูกกว่าค่าแรงของผู้ใช้ที่เสียไปในการกรอกข้อมูล หรือที่เสียไปในการนำแบบฟอร์มไปบันทึกเข้าในระบบคอมพิวเตอร์มากนัก

การดีไซน์แบบฟอร์มที่ดีจะต้องคำนึงถึงลำดับในการกรอกข้อมูลให้คล้องจองกับความเป็นจริง ตัวอย่างเช่น แบบฟอร์มกำหนดให้พนักงานกรอกชื่อเป็นอันดับที่ 1 และนามสกุลเป็นอันดับที่ 2 หรือการกรอกวันที่ ให้กรอกวันก่อนแล้วมากรอกเดือน และท้ายสุดก็คือ ปี พ.ศ. อย่างนี้ก็ถือว่าลำดับในการกรอกข้อมูลได้คล้องจองไปกับมาตรฐานที่ใช้กันโดยทั่วไป ในทางตรงข้าม หากเราสลับตำแหน่งให้พนักงานกรอกนามสกุลก่อนแล้วค่อยกรอกชื่อหรือกรอกปีก่อน แล้วค่อยกรอกเดือนหรือวันตามลำดับ แบบนี้เราคนไทยที่จะกรอกฟอร์มนี้ก็คงสับสนและมีการกรอกผิดพลาดกันแน่ๆ

2. **แบบฟอร์มต้องตรงกับวัตถุประสงค์ที่ต้องการ** แบบฟอร์มที่ได้ถูกดีไซน์ขึ้นมานั้นจะมีวัตถุประสงค์ของมันด้วยกันทั้งนั้น ซึ่งบางแบบฟอร์มอาจจะต้องถูกสำเนาและกระจายส่งผ่านไปยังแผนกต่างๆ อีกหลายแผนก ดังนั้น ก่อนที่จะทำการดีไซน์แบบฟอร์มใดๆ นักวิเคราะห์ระบบ หรือผู้ที่จะดีไซน์แบบฟอร์มจะต้องคำนึงถึงวัตถุประสงค์ของแบบฟอร์มนั้นเสียก่อนว่ามีขึ้นเพื่อประโยชน์อันใด และจะต้องมีข้อมูลอะไรบ้างที่จะต้องถูกบันทึกลงไป เอกสารจะถูกกระจายไปยังหน่วยงานไหนบ้าง และหน่วยงานนั้นจะเอาข้อมูลในส่วนไหนไปทำอะไร เป็นต้น

ตัวอย่างเช่น ใบกำกับภาษี ซึ่งอาจจะต้องประกอบไปด้วยสำเนาหลายฉบับกระจายออกไปยังหน่วยงานต่างๆ เช่น คลังสินค้า เพื่อใช้ในการตัดสต๊อก หรือแผนกขนส่งเพื่อนำสินค้าไปส่งให้ลูกค้าได้ถูกต้อง แผนกบัญชีเพื่อใช้บันทึกเป็นภาษีขาย และฝ่ายการเงินเพื่อนำไปใช้เรียกเก็บเงินในภายหลังจากการที่มีการทำสำเนาแต่ละฉบับไปยังหน่วยงานที่มีจุดประสงค์ไม่เหมือนกัน สำเนาแบบฟอร์มใบกำกับภาษีอาจจะต้องให้มีการบันทึกรายละเอียดไม่เหมือนกัน เช่น ใบที่ส่งไปให้กับฝ่ายการเงินเพื่อจะนำไปเรียกเก็บเงิน อาจจะมีช่อง "ชื่อผู้เก็บเงิน" ในขณะที่ใบที่อยู่กับแผนกขนส่ง อาจจะมีช่อง "ชื่อพนักงานขับรถ" หรือใบที่อยู่กับสต๊อกอาจมีช่อง "ชื่อผู้เบิกสินค้า" แทน เป็นต้น

3. **แบบฟอร์มควรมีการดีไซน์ให้ตรวจสอบความถูกต้องได้** ในการบันทึกข้อมูลนั้น อัตราการเกิดข้อผิดพลาดจะขึ้นอยู่กับการดีไซน์แบบฟอร์มด้วย หากแบบฟอร์มได้รับการดีไซน์ที่ดี โอกาสที่จะเกิดข้อผิดพลาดก็ลดลง การดีไซน์แบบฟอร์มจึงควรที่จะให้ความสำคัญในอันที่จะทำให้ผู้ใช้แบบฟอร์มสามารถกรอกข้อมูลได้อย่างถูกต้องและสะดวกที่สุดเท่าที่จะทำได้ แม้ว่าผู้ใช้แบบฟอร์มจะมีโอกาสใช้แบบฟอร์มนั้นแค่ครั้งเดียวหรืออาจจะเป็น 1,000 ครั้งก็ตาม

4. **แบบฟอร์มควรดีไซน์ให้มีลักษณะที่ดึงดูดต่อผู้ใช้** การดีไซน์แบบฟอร์มให้เป็นที่ดึงดูดใจต่อผู้ใช้นั้นอาจถือเป็นงานศิลปะอย่างหนึ่ง แต่ก็มีความสำคัญในตัวของมันเองอยู่ เป็นหลักจิตวิทยาอย่างหนึ่งที่เราไม่สามารถจะโต้เถียงได้ว่า หากแบบฟอร์มมีจุดดึงดูดแล้ว มันมักจะช่วยให้ผู้ใช้กรอกข้อมูลที่เราต้องการได้ดีขึ้น และผู้กรอกก็จะรู้สึกพอใจที่จะกรอกมากขึ้น

หลักการในการออกแบบฟอร์มนี้ เราจะต้องเน้นในเรื่องของความเป็นระเบียบของแบบฟอร์ม โดยจัดให้ข้อมูลที่ควรจะอยู่ด้วยกันเป็นกลุ่มๆ เพื่อให้เกิดความต่อเนื่องและง่ายต่อการกรอก บรรทัดและช่องว่างระหว่างบรรทัดจะต้องกว้างเพียงพอที่จะกรอก ในแบบฟอร์มอาจจะใช้ตัวอักษรที่มีขนาดแตกต่างกัน เพื่อให้สามารถจะเน้นจุดต่างๆ ได้ การใช้กรอบตารางและความหนาของตัวอักษรและเส้นต่างๆ เหล่านี้ล้วนเป็นเทคนิคที่จะช่วยดึงดูดความสนใจจากผู้ใช้แบบฟอร์มได้เป็นอย่างดี

**การดีไซน์อินพุตทางจอภาพ**

หลักเกณฑ์ที่เราจะทำการดีไซน์อินพุตทางจอภาพนั้นไม่ได้แตกต่างกับการดีไซน์เอาต์พุตทางจอภาพแต่อย่างไร ซึ่งจะใช้หลักเกณฑ์สำคัญ 4 ข้อในการดีไซน์เช่นกัน คือ

1. **พยายามให้การแสดงข้อมูลบนจอภาพดูเรียบง่ายไม่ซับซ้อน** ก่อนที่จะทำการดีไซน์จอภาพนักวิเคราะห์ระบบควรจะเข้าใจลักษณะพื้นฐานโดยทั่วไปของการจัดวางข้อมูลบนจอภาพเสียก่อน โดยพื้นที่ที่ใช้แสดงข้อมูลบนจอภาพ จะถูกแบ่งออกเป็น 3 ส่วนด้วยกัน คือ

**พื้นที่ส่วนหัวของจอภาพ** (Heading) โดยส่วนใหญ่จะเป็นส่วนที่แสดงข้อมูลให้ผู้ใช้ระบบได้รับทราบว่ากำลังทำงานอยู่ในระบบงานอะไร เช่น ระบบงานสินค้าคงคลัง ระบบงานบัญชี ฯลฯ

นอกจากนี้ ในปัจจุบันการออกแบบระบบงานแบบพูลดาวน์เมนูกำลังเป็นที่นิยมใช้กันอย่างแพร่หลาย โดยตัวเมนูก็จะแสดงอยู่ในส่วนหัวของจอภาพด้านบน ผู้ใช้สามารถเลือกเมนูได้โดยการเลื่อนเคอร์เซอร์หรือ Light Bar ไปที่เมนูที่ต้องการแล้วกด (Enter) หรือในกรณีที่ผู้ใช้เกิดความชำนาญอาจจะใช้การคีย์ตัวอักษรของเมนูนั้นเพื่อเลือกเมนูก็ทำได้เช่นกัน ตัวอย่างซอฟต์แวร์ที่ใช้ระบบพูลดาวน์เมนู ได้แก่ CU-Writer ของจุฬาลงกรณ์มหาวิทยาลัย IRC Standard Word และ Quatro Pro

**พื้นที่ส่วนกลางของจอภาพ** (Body) โดยทั่วไปพื้นที่ส่วนนี้จะใช้แสดงรายละเอียดของข้อมูลหรือหัวข้อต่างๆ ที่ผู้ใช้ระบบจะต้องทราบเพื่ออินพุตข้อมูลลงไปให้ถูกตำแหน่ง เทคนิคของการดีไซน์ในส่วนนี้ยังคง

ใช้ตามแบบมาตรฐาน คือพยายามให้ผู้ใช้ระบบอินพุตหรือกรอกข้อมูลลงในลักษณะจากบนลงล่างหรือจากซ้ายไปขวา

**พื้นที่ส่วนล่างของจอภาพ** (Ending) โดยทั่วไปพื้นที่ส่วนนี้จะใช้ประโยชน์ในด้านของการบอกให้ผู้ใช้ทราบถึงคำสั่งต่างๆ ที่ระบบงานกำหนดให้ผู้ใช้สามารถกระทำได้ เช่น กด (F1) เพื่อเรียกคำช่วยอธิบายวิธีการใช้ระบบ (Help-Text Sensitivity) หรือกด (F8) เพื่อเก็บข้อมูล (Save)

ในปัจจุบัน เทคโนโลยีทางด้านซอฟต์แวร์ได้รับการพัฒนาไปอย่างมาก โดยเฉพาะแนวทางที่จะพยายามให้ระบบงานหรือซอฟต์แวร์มีความเป็นมิตรกับผู้ใช้ระบบ (User Friendly) ให้มากที่สุด เช่น ระบบ Graphical User Interface (GUI) ที่ใช้รูปภาพหรือไอคอน (Icon) แทนคำสั่ง โดยผู้ใช้อาจใช้เมาส์ (Mouse) แทนคีย์บอรืด (Keyboard) ในการปฏิบัติงานก็ได้

นอกจากนี้ ก็ยังมีเทคนิคประเภทอื่นที่ยังคงนิยมกันมากก็คือ การใช้เทคนิคของการซ้อนกันของหน้าต่าง (Overlay Windows หรือ Pop-Up Windows) บนจอภาพก็เป็นอีกวิธีหนึ่งที่จะทำให้ระบบงานดูง่าย และเป็นที่ดึงดูดต่อผู้ใช้ด้วย

2. **พยายามให้การแสดงผลบนจอภาพมีมาตรฐานแบบเดียวกัน** เพื่อให้ผู้ใช้เกิดความคุ้นเคยได้เร็ว การทำให้จอภาพมีมาตรฐานนั้นนอกจากจะทำให้ผู้ใช้สามารถเรียนรู้ได้เร็วแล้ว ยังทำให้ลดข้อผิดพลาดลงได้อย่างมากอีกด้วย หากผู้ใช้ระบบจะต้องใช้เอกสารในการกรอกข้อมูลลงบนจอภาพแล้ว นักวิเคราะห์ระบบก็ควรจะดีไซน์จอภาพให้คล้องจองกันกับเอกสารที่ผู้ใช้ระบบจะต้องใช้ในการกรอกด้วย

การแสดงผลจะมีมาตรฐานได้ก็ด้วยวิธีการง่ายๆ คือ ตำแหน่งของข้อมูลควรจะปรากฏอยู่ในที่เดียวกันทุกครั้ง หากว่าข้อมูลนั้นเป็นข้อมูลอันเดียวกัน รวมทั้งข้อมูลไหนที่ควรจะอยู่ด้วยกันก็ควรจะจัดแบ่งออกให้เป็นกลุ่มๆ อย่างชัดเจน

3. **สำหรับข้อมูลบางอย่างที่ต้องการจะเน้นให้เห็นถึงความแตกต่าง** ให้ใช้สีที่แตกต่างออกไปจากปกติเพื่อดึงดูดความสนใจของผู้ใช้ ในปัจจุบันจอสีกำกับเป็นที่นิยมใช้กันมากขึ้นทุกขณะ และมีแนวโน้มที่จะมาครองตลาดแทนจอภาพแบบขาวดำหรือโมโนโครม สาเหตุที่เป็นเช่นนี้เนื่องมาจากจอภาพสีสามารถทำให้ระบบงานคอมพิวเตอร์มีความดึงดูดมากขึ้นด้วยสีสันที่แตกต่าง ความละเอียดของภาพที่ได้ก็ดีกว่าอย่างเห็นได้ชัด การแสดงผลทางกราฟฟิกในแบบต่างๆ ก็ทำได้โดยสะดวกและชัดเจนกว่าซอฟต์แวร์ในตลาดก็เริ่มปรับตัวให้ใช้กับจอภาพแบบสีกันอย่างมากมาย ดังนั้น ความสำคัญของการใช้สีจึงเป็นอีกจุดหนึ่งที่นักวิเคราะห์ระบบควรจะให้ความสำคัญด้วย

การเลือกใช้สีควรจะใช้ให้เหมาะสมด้วย เช่น พื้นสีแดง โดยทั่วไปมักจะใช้ในการบอกถึงอันตรายต่างๆ ที่อาจเกิดขึ้นในระบบงานคอมพิวเตอร์ พื้นสีน้ำเงินจะใช้ในการแสดงผลทางปกติ พื้นสีเขียวอาจใช้ในการแสดงข้อมูลความช่วยเหลือแบบต่างๆ ดังนั้นในซอฟต์แวร์หรือในระบบงานหนึ่งหากใช้สีปนเปกันไป โดยไม่คำนึงถึงความหมายของแต่ละสีแล้ว ก็อาจจะทำให้ผู้ใช้ระบบเกิดความสับสนและจะก่อให้เกิดผลเสียหายตามมาในภายหลังได้เช่นกัน

4. **ให้การโต้ตอบระหว่างผู้ใช้ระบบกับจอภาพเป็นไปโดยธรรมชาติมากที่สุด** เช่น การเลื่อนเคอร์เซอร์ (Cursor Movement) ควรจะเลื่อนจากบนลงล่างหรือจากซ้ายมาขวา ซึ่งเป็นไปตามธรรมชาติและมาตรฐานสากล

**สัปดาห์ที่** 11

Database **และการสร้าง** Database Diagram

.....

## การออกแบบข้อมูล

Data Design

### แนวความคิดการออกแบบข้อมูล

ก่อนที่จะสร้างระบบสารสนเทศ นักวิเคราะห์ระบบต้องมีความเข้าใจแนวความคิดพื้นฐานของการออกแบบข้อมูล ซึ่งรวมการสร้างโครงสร้างข้อมูล คุณสมบัติพิเศษของแฟ้มข้อมูลและฐานข้อมูล นอกจากนี้ นักวิเคราะห์ระบบยังต้องมีความเข้าใจในเรื่องดาต้าแวร์เฮาส์ ดาต้าไมนนิ่ง และองค์ประกอบของระบบจัดการฐานข้อมูล

### โครงสร้างข้อมูล

แฟ้มข้อมูลหรือไฟล์ (File) บรรจุด้วยข้อมูล เกี่ยวกับคน สถานที่ สิ่งของ หรือเหตุการณ์ที่โต้ตอบกับระบบสารสนเทศ เช่น แฟ้มข้อมูลอาจประกอบด้วยข้อมูลเกี่ยวกับลูกค้า สินค้า คำสั่งซื้อหรือผู้ขาย ระบบสารสนเทศสามารถออกแบบได้ทั้ง ระบบเชิงแฟ้มข้อมูล (File-oriented System) และระบบฐานข้อมูล (Database System) ทั้งนี้ขึ้นอยู่กับความต้องการของธุรกิจ

ระบบเชิงแฟ้มข้อมูลเป็นการประมวลผลของหนึ่งแฟ้มข้อมูลหรือมากกว่า แยกจากกันโดยใช้วิธีการที่เรียก การประมวลผลแฟ้มข้อมูล (File Processing)

ฐานข้อมูล ประกอบด้วยแฟ้มข้อมูลที่เชื่อมโยงกัน เรียกว่า ตาราง (Table) ซึ่งเป็นโครงสร้างของข้อมูลทั้งหมด เมื่อเปรียบเทียบกับการประมวลผลแฟ้มข้อมูล สภาพแวดล้อมของฐานข้อมูลจะมีความยืดหยุ่นและมีประสิทธิภาพดีกว่า

### ภาพรวมการประมวลผลแฟ้มข้อมูล

ระบบบางระบบใช้การประมวลผลแฟ้มข้อมูล ในการจัดการโครงสร้างข้อมูลที่มีปริมาณมากๆ เพราะเหมาะสมกับฮาร์ดแวร์ของเมนเฟรมและวิธีการป้อนข้อมูลเข้าแบบกลุ่ม (Batch Input) ถึงแม้ว่าในปัจจุบันจะมีจำนวนการใช้งานลดน้อยลง แต่ในบางสถานการณ์การประมวลผลแบบแฟ้มข้อมูลจะเกิดประสิทธิภาพที่สูงกว่า

ในสภาพแวดล้อมการประมวลผลแฟ้มข้อมูลโดยทั่วไป หากมีสามหน่วยงาน ที่มีระบบสารสนเทศและแฟ้มข้อมูลแยกของแต่ละหน่วยงาน อาจทำให้เกิดหลายปัญหาขึ้นได้ ปัญหาแรก คือ ข้อมูลที่ซ้ำซ้อน (Data Redundancy) ข้อมูลที่เหมือนกันจะถูกจัดเก็บไว้ในหลายๆ ที่ ซึ่งต้องการเนื้อที่ในการจัดเก็บ การบำรุงรักษา และการปรับปรุงข้อมูล อันทำให้เกิดค่าใช้จ่ายที่สูงขึ้น ปัญหาที่สอง คือ บูรณการของข้อมูล (Data Integrity) ถ้าการปรับปรุงนั้นไม่ได้กระทำในทุกแฟ้มข้อมูล การเปลี่ยนข้อมูลในระบบหนึ่งจะเป็นสาเหตุให้ไม่ตรงกับข้อมูลในอีกระบบหนึ่ง ปัญหาที่สาม คือ ความไม่ยืดหยุ่นของโครงสร้างข้อมูล (Rigid Data Structure) โดยปกติธุรกิจจะอาศัยการตัดสินใจบนข้อมูลที่ธุรกิจมีอยู่และจากสารสนเทศของหน่วยงานต่างๆ ซึ่งในสภาพแวดล้อมแบบนี้ จะทำให้การดึงสารสนเทศจากระบบกระทำได้ช้าและไม่มีประสิทธิภาพ

ระบบสารสนเทศแบบเชิงแฟ้มข้อมูล เป็นการใช้แฟ้มข้อมูลหลายประเภท ได้แก่ แฟ้มหลัก แฟ้มตาราง แฟ้มรายการเปลี่ยนแปลง แฟ้มทำการ แฟ้มความปลอดภัย และแฟ้มประวัติ

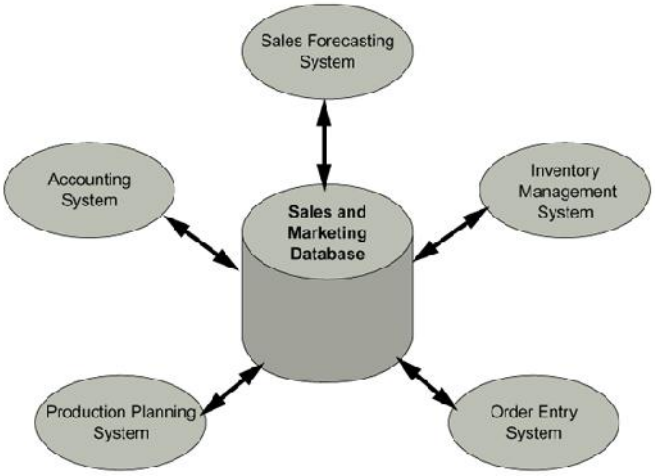
- แฟ้มหลักหรือมาสเตอร์ไฟล์ (Master File) ใช้เก็บข้อมูลเชิงถาวรที่เกี่ยวข้องกับเอนทิตี

- แฟ้มตารางหรือเทเบิลไฟล์ (Table File) เป็นแฟ้มบรรจุข้อมูลอ้างอิงที่ใช้ในระบบสารสนเทศ เป็นแฟ้มข้อมูลเชิงถาวร แต่จะไม่ถูกปรับปรุงข้อมูลด้วยระบบสารสนเทศ
- แฟ้มรายการเปลี่ยนแปลงหรือทรานเซคชั่นไฟล์ (Transaction File) ใช้เก็บเรคคอร์ดของการดำเนินกิจการประจำวัน มีจุดมุ่งหมายเพื่อเป็นแฟ้มข้อมูลเข้าสำหรับใช้ในการปรับปรุงมาสเตอร์ไฟล์ จึงเป็นแฟ้มชั่วคราว (Temporary File)
- แฟ้มทำการหรือเวิร์กไฟล์ (Work File) เป็นแฟ้มชั่วคราว ซึ่งสร้างขึ้นโดยระบบสารสนเทศเพื่องานหนึ่งงานใด มักจะถูกสร้างในการประมวลผลหนึ่งของระบบสารสนเทศ และถูกใช้โดยอีกการประมวลผลหนึ่งในระบบสารสนเทศ จึงเรียกอีกชื่อหนึ่งว่า แฟ้มครั้งคราวหรือสะแค็ชไฟล์ (Scratch File)
- แฟ้มความปลอดภัยหรือซีเคียวริตีไฟล์ (Security File) ถูกสร้างและเก็บไว้เพื่อการสำรองข้อมูลและการกู้ข้อมูล เช่น แฟ้มหลักฐานการตรวจสอบ (Audit Trail File) และแฟ้มสำรองสำหรับมาสเตอร์ไฟล์ เทเบิลไฟล์ หรือทรานเซคชั่นไฟล์
- แฟ้มประวัติหรือฮิซทอรีไฟล์ (History File) เป็นแฟ้มสำเนาที่ถูกสร้างและเก็บไว้เป็นประวัติหรือเก็บถาวร ซึ่งไม่เหมือนแฟ้มความปลอดภัย คือ แฟ้มใหม่ไม่ได้แทนที่แฟ้มเก่า

**ภาพรวมของระบบฐานข้อมูล**

ฐานข้อมูลหรือดาต้าเบส (Database) ได้ถูกออกแบบมาสำหรับแก้ไขปัญหาที่พบจากการประมวลผลแฟ้มข้อมูล ฐานข้อมูลจะจัดทำโครงร่าง (Framework) ทั้งหมด เพื่อหลีกเลี่ยงปัญหาของการซ้ำซ้อนข้อมูลและสนับสนุนสภาพแวดล้อมแบบกระทำทันที (Real-time) และพลวัต (Dynamic)

ในสภาพแวดล้อมแบบการประมวลผลแฟ้มข้อมูล แฟ้มข้อมูลถูกออกแบบมาให้ตรงกับระบบธุรกิจหนึ่ง ในทางตรงกันข้ามสภาพแวดล้อมแบบฐานข้อมูล หลายๆ ระบบจะใช้ข้อมูลจากฐานข้อมูลเดียวกัน ดังรูปที่ 8-4 แสดงถึงสภาพแวดล้อมแบบฐานข้อมูลที่ให้บริการถึง 5 ระบบ




รูปที่ 1 สภาพแวดล้อมแบบฐานข้อมูลจะประกอบด้วยฐานข้อมูลที่สามารถให้บริการ 5 ระบบที่แยกจากกัน

จากมุมมองของผู้ใช้ ข้อได้เปรียบหลักของระบบจัดการฐานข้อมูล คือ เวลา การโต้ตอบ และความยืดหยุ่นในการเข้าถึงข้อมูล และอื่นๆ ดังนี้

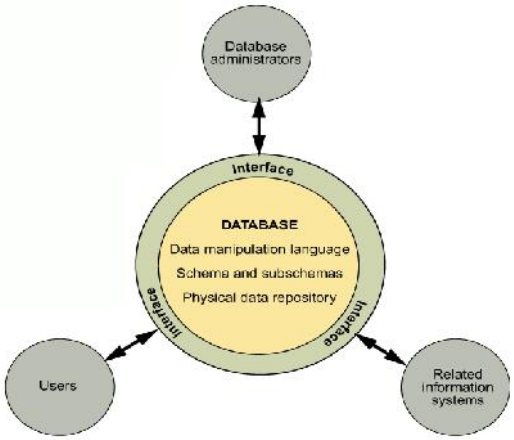
- ความสามารถในการปรับขนาด
- ให้การสนับสนุนระบบไคลเอ้นท์เซิร์ฟเวอร์
- ช่วยประหยัด

- การใช้ข้อมูลร่วมกันได้ทั้งองค์กร
- สร้างดุลยภาพให้แก่ความต้องการที่ขัดแย้งกัน
- บังคับให้เป็นไปตามมาตรฐานแบบเดียวกันทั้งองค์กร
- ควบคุมการซ้ำซ้อน
- มีความปลอดภัย
- เพิ่มผลผลิตของนักเขียนโปรแกรม
- ความเป็นอิสระของข้อมูล

แม้ว่าระบบจัดการฐานข้อมูลจะมีข้อได้เปรียบอยู่มาก แต่ก็มีข้อเสียบ้าง เนื่องจากว่าระบบจัดการฐานข้อมูลมีความสามารถมาก ดังนั้น จึงต้องการฮาร์ดแวร์ ซอฟต์แวร์ และเครือข่ายที่มีราคาแพงเพื่อสนับสนุนสภาพแวดล้อมแบบหลายๆ ผู้ใช้พร้อมกัน และยังมีความซับซ้อนมากกว่าระบบประมวลผลแฟ้มข้อมูล ดังนั้น หากเปรียบกับเส้นโค้งในการเรียนรู้ของนักวิเคราะห์ระบบ ผู้บริหารระบบ หรือผู้ใช้งานจะมีความชันขึ้น ซึ่งทำให้ค่าใช้จ่ายในส่วนของเจ้าความเป็นของ (Total Cost of Ownership) สูงขึ้นด้วย สุดท้ายเป็นเรื่องของระบบความปลอดภัย การสำรองข้อมูล และการกู้ข้อมูล ซึ่งจะมีความยุ่งยากและวิกฤตตามแบบสภาพแวดล้อมแบบฐานข้อมูล หากเกิดความขัดข้องของระบบจัดการฐานข้อมูลจะทำให้การดำเนินงานของธุรกิจยุ่งยากอย่างยิ่ง

**ส่วนประกอบของระบบจัดการฐานข้อมูล**

ระบบจัดการฐานข้อมูลจะจัดเตรียมส่วนต่อประสานระหว่างฐานข้อมูลกับผู้ใช้ที่ต้องการเข้าถึงข้อมูล นักวิเคราะห์ระบบควรมีความเข้าใจในส่วนประกอบของระบบจัดการฐานข้อมูล นอกเหนือจากส่วนต่อประสานผู้ใช้ การบริหารฐานข้อมูล และระบบสารสนเทศที่เกี่ยวข้องแล้ว ระบบจัดการฐานข้อมูลยังประกอบด้วย ภาษาถ่ายเทข้อมูล ซีคีม่า และแหล่งรวบรวมข้อมูลเชิงกายภาพ ดังแสดงในรูป 8-5




รูปที่ 2 นอกเหนือจากส่วนต่อประสานผู้ใช้ การบริหารฐานข้อมูล และระบบสารสนเทศที่เกี่ยวข้องแล้ว ระบบจัดการฐานข้อมูลยังประกอบด้วย ภาษาถ่ายเทข้อมูล ซีคีม่า และแหล่งรวบรวมข้อมูลเชิงกายภาพ

**ส่วนต่อประสานผู้ใช้ การบริหารฐานข้อมูล และระบบสารสนเทศที่เกี่ยวข้อง**

เมื่อผู้ใช้งาน ผู้บริหารระบบ และระบบสารสนเทศที่เกี่ยวข้อง ร้องขอข้อมูลและการบริการ DBMS จะทำการประมวลคำขอถ่ายเทข้อมูลและจัดเตรียมการสนองตอบ ดังนี้

- ผู้ใช้งาน โดยปกติจะใช้คำถามหรือคำสั่งที่ได้มีการจัดเตรียมไว้แล้ว และสามารถใช้ภาษาสอบถาม (Query Language) เพื่อเข้าถึงแหล่งเก็บข้อมูล ภาษาสอบถามจะทำให้ผู้ใช้สามารถระบุงานได้โดย

ไม่ต้องระบุวิธีการทำงาน ภาษาสอบถามบางภาษาใช้คำสั่งแบบภาษาธรรมชาติที่เหมือนประโยคในภาษาอังกฤษทั่วไป

- ผู้บริหารฐานข้อมูลหรือ DBA มีหน้าที่รับผิดชอบต่อการจัดการและสนับสนุน DBMS นอกจากนี้ ยังดูแลความปลอดภัยของข้อมูลและบูรณการ ป้องกันการเข้าถึงข้อมูลจากผู้ไม่ได้รับสิทธิ์ จัดเตรียมการสำรองและการกู้ข้อมูล เก็บหลักฐานการตรวจสอบ การบำรุงรักษาฐานข้อมูล และสนับสนุนความต้องการของผู้ใช้ ระบบจัดการฐานข้อมูลส่วนใหญ่จะมีโปรแกรมอรรถประโยชน์ (Utility Program) เพื่อช่วยผู้บริหารฐานข้อมูลในการสร้างและปรับปรุงโครงสร้างข้อมูล การเก็บและจัดทำรายงานรูปแบบการใช้ฐานข้อมูล ตลอดจนการตรวจสอบและรายงานความผิดปกติของฐานข้อมูล
- ระบบสารสนเทศที่เกี่ยวข้อง โดย DBMS จะต้องให้การสนับสนุนระบบสารสนเทศที่เกี่ยวข้องได้ในหลายๆ ระบบ

**ภาษาถ่ายเทข้อมูล**

ภาษาถ่ายเทข้อมูล (Data Manipulation Language : DML) ใช้ควบคุมการทำงานของฐานข้อมูล รวมถึงการจัดเก็บ การดึงข้อมูล การปรับปรุง และการลบข้อมูล ระบบจัดการฐานข้อมูลส่วนใหญ่ ใช้ภาษาถ่ายเทข้อมูล

**เค้าร่าง**

นิยามที่สมบูรณ์ของฐานข้อมูล รวมถึงคำอธิบายฟิลด์ เรคคอร์ดและความสัมพันธ์ทั้งหมด จะเรียกว่า เค้าร่างหรือซคีมา (Schema) ซึ่งสามารถจะกำหนด เค้าร่างย่อยหรือซับซคีม่า (Subschema) เพิ่มเติมได้ โดยที่ซับซคีม่านั้น เป็นภาพของฐานข้อมูลที่ถูกใช้โดยระบบหรือผู้ใช้ ซับซคีม่าสามารถกำหนดส่วนของฐานข้อมูลที่ระบบหรือผู้ใช้ต้องการหรืออนุญาตให้เข้าถึงได้ เช่น ในการป้องกันความเป็นส่วนตัว โดยไม่ให้ระบบบริหารโครงการสามารถดึงข้อมูลอัตราค่าจ้างได้ ในกรณีนี้ ซับซคีม่าของระบบบริหารโครงการควรจะไม่กำหนดฟิลด์อัตราค่าจ้างไว้ เป็นต้น นอกจากนั้น ผู้ออกแบบฐานข้อมูลยังใช้ซับซคีม่าในการจำกัดระดับของการอนุญาตในการเข้าถึงข้อมูล เฉพาะผู้ใช้ระบบหรือเฉพาะสถานที่ที่ได้รับอนุญาต ให้สร้าง ดึง แก้ไข และลบข้อมูลได้ ทั้งนี้ขึ้นอยู่กับความต้องการและนโยบายด้านความปลอดภัย

**แหล่งรวบรวมข้อมูลเชิงกายภาพ**

จากบทที่ 4 ได้เรียนรู้เกี่ยวกับพจนานุกรมข้อมูล (Data Dictionary) ซึ่งอธิบายถึง ส่วนของข้อมูลย่อย (Data Element) ทั้งหมด รวมทั้งการออกแบบเชิงตรรกะ ในขั้นตอนของกรรมวิธีการพัฒนาระบบนี้ พจนานุกรมข้อมูลจะถูกโอนย้ายไปเป็นแหล่งรวบรวมข้อมูลเชิงกายภาพ ซึ่งบรรจุด้วยซคีม่าและซับซคีม่า แหล่งรวบรวมเชิงกายภาพนี้ อาจจะรวมศูนย์หรือกระจายไปในหลายๆ สถานที่ได้

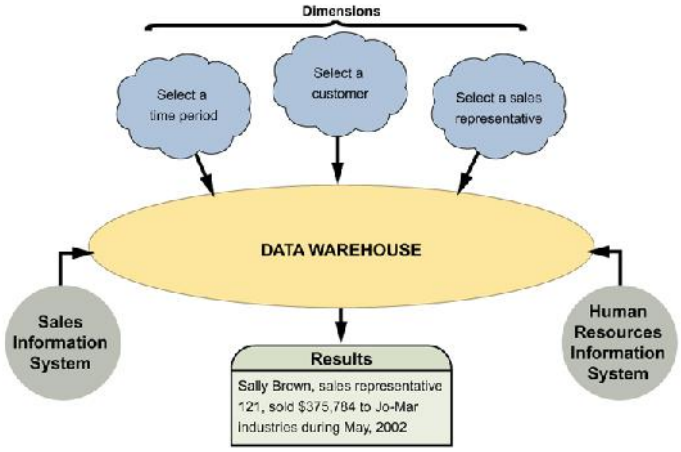
**ดาต้าแวร์เฮาส์**

ธุรกิจขนาดใหญ่หลายธุรกิจใช้โปรแกรมสำเร็จ ซึ่งจัดและเก็บข้อมูลไว้ในโครงแบบชนิดพิเศษที่เรียกว่า โกดังข้อมูลหรือดาต้าแวร์เฮาส์ (Data Warehouse) ซึ่งเป็นการรวบรวมข้อมูลแบบเบ็ดเสร็จเพื่อ

สนับสนุนการวิเคราะห์และตัดสินใจของฝ่ายบริหาร ยกตัวอย่างของธุรกิจโดยทั่วไป ข้อมูลจะถูกสร้างจากระบบเชิงรายการเปลี่ยนแปลง (Transaction-based Systems) เช่น การป้อนข้อมูลคำสั่งซื้อ คลังสินค้า ลูกหนี้ หรือเงินเดือน ถ้าต้องการทราบรหัสลูกค้าของใบสั่งซื้อเลขที่ 4071 ก็สามารถดึงข้อมูลได้โดยง่ายจากระบบป้อนข้อมูลใบสั่งซื้อ เป็นต้น

อีกนัยหนึ่ง หากต้องการดูยอดขายในเดือนพฤษภาคมของตัวแทนขายที่ได้รับมอบหมายให้ดูแลลูกค้ารายหนึ่งโดยเฉพาะ ถึงแม้ว่าระบบสารสนเทศจะเป็นระบบแบบโต้ตอบ แต่ก็ยากที่จะให้ผู้ใช้คัดเอาข้อมูลเฉพาะที่เจาะจง ซึ่งทอดข้ามไปยังหลายๆ ระบบออกมาได้ภายในกรอบเวลาที่กำหนด ซึ่งผู้ใช้ส่วนมากจะต้องอาศัยความช่วยเหลือจากเจ้าหน้าที่ไอที

เพื่อแทนการเข้าถึงระบบที่แยกจากกัน ดาต้าแวร์เฮาส์จะเก็บข้อมูลรายการการเปลี่ยนแปลงไว้ในรูปแบบที่ยอมให้ผู้ใช้เข้าถึงรวมไว้ด้วยกันและสามารถวิเคราะห์ข้อมูลนั้นได้ นอกจากนี้ยังอนุญาตให้ผู้ใช้กำหนดมิติ (Dimension) ที่แน่นอน หรือลักษณะพิเศษได้ สำหรับดาต้าแวร์เฮาส์ของสินค้าอุปโภคบริโภค มิติจะหมายถึง เวลา ลูกค้า หรือตัวแทนขาย โดยการเลือกค่าสำหรับแต่ละลักษณะ เพื่อให้ผู้ใช้ได้รับสารสนเทศจากข้อมูลที่จัดเก็บได้หลายมิติ




รูปที่ 3 ดาต้าแวร์เฮาส์อาจเก็บข้อมูลมาจากหลายระบบสารสนเทศ เพื่อให้ผู้ใช้ได้รับสารสนเทศจากข้อมูลที่จัดเก็บได้หลายมิติ

**ดาต้าไมนนิง**

ซอฟต์แวร์เกี่ยวกับการทำเหมืองข้อมูลหรือดาต้าไมนนิง (Data Mining) แสดงถึง รูปแบบที่มีความหมายและมีความสัมพันธ์กันระหว่างข้อมูล ยกตัวอย่าง ดาต้าไมนนิงซอฟต์แวร์ที่ช่วยให้ธุรกิจสินค้าอุปโภคบริโภคสามารถระบุลูกค้าที่คาดหวังบนพื้นฐานของสถิติการสั่งซื้อของลูกค้า

การเติบโตอย่างรวดเร็วของพาณิชย์อิเล็กทรอนิกส์ได้เน้นไปที่ดาต้าไมนนิง เพื่อเป็นเครื่องมือในการวิเคราะห์ผู้เข้าใช้เว็บไซต์และแนวโน้มการใช้งาน

**ศัพท์เฉพาะของการออกแบบข้อมูล**

แนวความคิดของการออกแบบข้อมูลได้กล่าวแล้วในหัวข้อที่ผ่านมา นักวิเคราะห์ระบบสามารถเลือกกลยุทธ์และเริ่มสร้างระบบการจัดการข้อมูล โดยขั้นตอนแรกคือ การทำความเข้าใจในศัพท์เฉพาะของการออกแบบข้อมูล

- เอนทิตี (Entity) หมายถึง คน สถานที่ สิ่งของ หรือเหตุการณ์ ที่ข้อมูลถูกรวบรวมและบำรุงรักษา

- เขตข้อมูลหรือฟิลด์ (Field) เรียกอีกชื่อหนึ่งว่า ลักษณะประจำหรือแอททริบิวท์ (Attribute) ซึ่งเป็นคุณลักษณะพิเศษอย่างหนึ่งหรือเป็นข้อเท็จจริงของเอนทิตี
- ระเบียนข้อมูลหรือเรคคอร์ด (Record) หรือที่เรียกว่า ทูเพอร์ (Tuple) เป็นชุดของฟิลด์ที่สัมพันธ์กัน ที่ใช้อธิบายถึงหนึ่งสมาชิกหรืออินสแตนส์ (Instance) ของเอนทิตี
- แฟ้มข้อมูลและตาราง เรคคอร์ดที่ถูกจัดกลุ่มรวมเป็นแฟ้มข้อมูล (File) หรือ ตาราง (Table) โดยขึ้นอยู่กับแบบจำลองข้อมูลที่ได้ทำไว้ในระบบเชิงแฟ้มข้อมูล

ในระบบแฟ้มข้อมูลจะประกอบด้วยกลุ่มของเรคคอร์ดที่มีความสัมพันธ์รวมเป็นแฟ้ม เช่น ถ้าระบบสินค้าคงคลังมีสินค้า 1500 สินค้า แฟ้มสินค้าจะมี 1500 เรคคอร์ด โดยหนึ่งเรคคอร์ดสำหรับหนึ่งสินค้า

ในสภาพแวดล้อมแบบฐานข้อมูล กลุ่มของเรคคอร์ดที่เกี่ยวข้องจะถูกจัดกลุ่มเป็นตารางที่เก็บข้อมูลที่เกี่ยวกับเอนทิตีเฉพาะ ตารางแสดงถึงโครงสร้างแบบสองมิติที่ประกอบด้วย แถวในแนวตั้งแสดงฟิลด์และแถวในแนวนอนสำหรับเรคคอร์ด

**เขตหลัก**

ระหว่างระยะของการออกแบบระบบ จะใช้ เขตหลักหรือคีย์ฟิลด์ (Key Field) ในการรวบรวมการเข้าถึงและการบำรุงรักษาโครงสร้างข้อมูล โดยแบ่งออกเป็นสี่ประเภท คือ

- กุญแจหลักหรือไพรมาริคีย์ (Primary Key) เป็นฟิลด์หนึ่งหรือการผสมกันของฟิลด์ ซึ่งไม่ซ้ำกันหรือเป็นหนึ่งเดียว (Unique) และเพื่อระบุสมาชิกเฉพาะของเอนทิตี
- กุญแจให้เลือก บางครั้งอาจเลือกฟิลด์หรือฟิลด์ผสมเพื่อใช้เป็นไพรมาริคีย์ ฟิลด์ใดที่สามารถนำมาใช้เป็นไพรมาริคีย์ได้เรียกว่า กุญแจให้เลือกหรือแคนดิเดทคีย์ (Candidate key) ฟิลด์ใดที่ไม่ใช่ไพรมาริคีย์หรือแคนดิเดทคีย์ จะเรียกว่า ฟิลด์ที่ไม่เป็นกุญแจหรือน็อนคีย์ฟิลด์ (Non key field)
- กุญแจต่างลักษณะ เป็นฟิลด์ในตารางหนึ่งที่ต้องตรงกับค่าของไพรมาริคีย์ในอีกตารางหนึ่ง เพื่อสร้างความสัมพันธ์ระหว่างสองตารางนั้น ซึ่งจะต่างจากไพรมาริคีย์ ตรงที่สามารถซ้ำกันได้
- กุญแจรองหรือเซคคันเดริคีย์ (Secondary Key) เป็นฟิลด์หรือการผสมของฟิลด์ที่นำมาใช้ในการเข้าถึงหรือดึงเรคคอร์ด ค่าเซคคันเดริคีย์สามารถซ้ำกันได้

**การบูรณการอ้างอิง**

การบูรณการอ้างอิง (Referential Integrity) คือ ชุดของกฎเกณฑ์ที่หลีกเลี่ยงความไม่ต้องกันของข้อมูลและปัญหาเรื่องคุณภาพ ในฐานข้อมูลเชิงสัมพันธ์จะหมายถึงค่าของฟอรินคีย์ที่ไม่สามารถป้อนเข้าสู่ตารางหนึ่ง ถ้าไม่ตรงกับไพรมาริคีย์ที่มีอยู่แล้วในอีกตารางหนึ่ง เป็นอีกประเภทหนึ่งของการตรวจสอบความสมเหตุสมผล

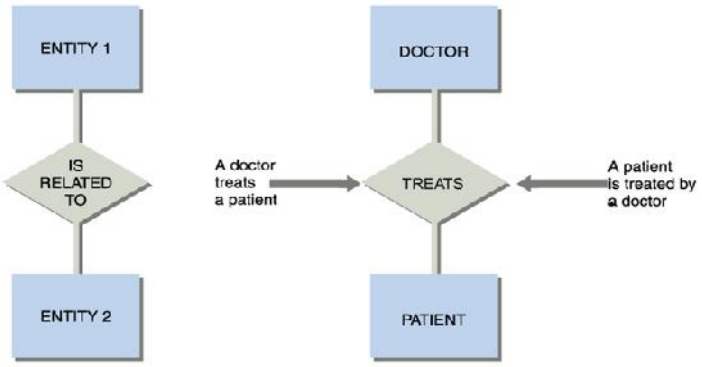
**ความสัมพันธ์ของข้อมูล**

ความสัมพันธ์ (Relationship) จะเป็นการเชื่อมโยงเชิงตรรกะระหว่างเอนทิตีที่เกี่ยวข้อง เช่น ความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตี PRODUCT และ WAREHOUSE เนื่องจากสินค้าถูกเก็บอยู่ในโกดังสินค้า

**แผนภาพความสัมพันธ์เอนทิตี**

แผนภาพความสัมพันธ์เอนทิตี (Entity-Relationship Diagrams : ERD) เป็นรูปจำลองภาพของระบบสารสนเทศที่บรรยายความสัมพันธ์ระหว่างเอนทิตีของระบบ รูปที่ 8-12 แสดงรูปแบบพื้นฐานของ ERD แต่ละเอนทิตีแสดงด้วยรูปสี่เหลี่ยมผืนผ้า และรูปสี่เหลี่ยมเพชรแสดงถึงความสัมพันธ์หรือความเกี่ยวพันที่เชื่อมระหว่างเอนทิตี ในสี่เหลี่ยมผืนผ้าเอนทิตีจะระบุด้วยชื่อที่เป็นคำนามเดี่ยวและในความสัมพันธ์รูปสี่เหลี่ยมเพชรจะระบุด้วยข้อความที่เป็นคำกิริยา เช่น รูป 8-12 แพทย์รักษาผู้ป่วย หรือ

ผู้ป่วยถูกรักษาโดยแพทย์ แต่ใช้กรรตุวาจก (Active Voice) จะดีกว่า ซึ่งบางเคสทูลสามารถแสดงความสัมพันธ์ได้ทั้งสองทิศทาง




รูปที่ 4 ในแผนภาพความสัมพันธ์เอนทิตี เอนทิตีจะระบุด้วยชื่อที่เป็นคำนามเดี่ยว และความสัมพันธ์ที่ระบุด้วยคำกริยา

ERD จะแตกต่างจากแผนภาพกระแสข้อมูล และผังระบบงาน ตรงที่ไม่กำหนดกระแสของข้อมูลหรือสารสนเทศ ไม่มีหัวลูกศร ให้เอนทิตีหนึ่งจะอยู่ในตำแหน่งด้านบนหรือด้านซ้ายของอีกเอนทิตีหนึ่ง โดยตำแหน่งนั้นไม่ได้หมายถึงความเกี่ยวพันที่เหนือกว่าอีกเอนทิตีหนึ่ง หรือการไหลจากเอนทิตีแรกไปสู่เอนทิตีที่สอง

ประเภทของความสัมพันธ์แบ่งออกได้เป็นสามประเภทหลัก คือ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่งความสัมพันธ์ (One-to-one Relationship) ใช้คำย่อว่า 1:1 เกิดขึ้นเมื่อเอนทิตีที่สอง เป็นเพียงหนึ่งอินสแตนส์ของเอนทิตีแรก

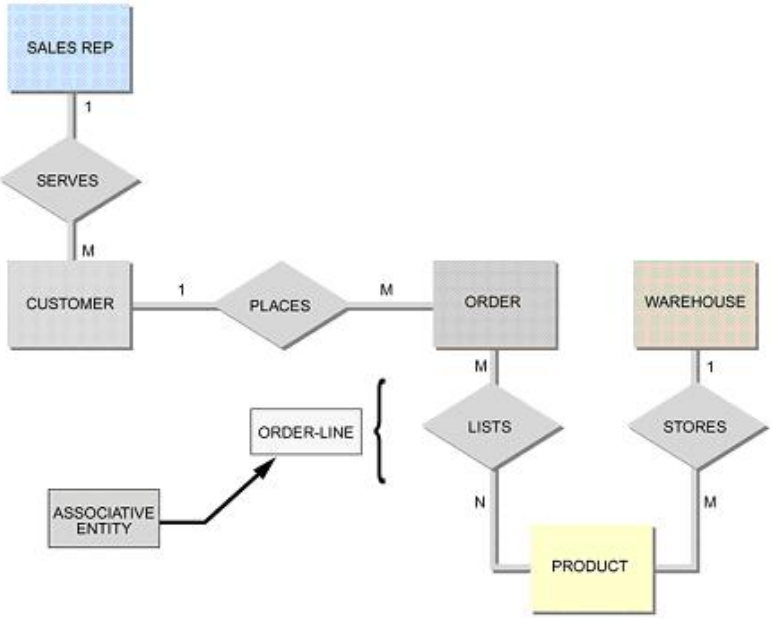
- ความสัมพันธ์แบบหนึ่งต่อหลายความสัมพันธ์ (One-to-many Relationship) ใช้ชื่อย่อว่า 1:M เกิดขึ้นเมื่อหนึ่งเอนทิตีแรกเกี่ยวข้องกับหลายเอนทิตีที่สอง และเอนทิตีที่สองสามารถเกี่ยวข้องเพียงหนึ่งเอนทิตีแรกเท่านั้น บางซอฟต์แวร์แสดงความสัมพันธ์หลายๆ ด้านด้วยสัญลักษณ์ไม่รู้จบ (Infinity) หรือสัญลักษณ์ตีนกา (Crow's Foot Notation)

- ความสัมพันธ์แบบหลายต่อหลายความสัมพันธ์ (Many-to-many Relationship) ใช้อักษรย่อว่า M:N เกิดขึ้นเมื่อหนึ่งอินสแตนส์ของเอนทิตีแรกสัมพันธ์กับหลายอินสแตนส์ของเอนทิตีที่สอง และหนึ่งอินสแตนส์ของเอนทิตีที่สองสัมพันธ์กับหลายอินสแตนส์ของเอนทิตีแรก

ให้สังเกตว่าความสัมพันธ์ M:N จะแตกต่างจาก 1:1 หรือ 1:M เนื่องจากเหตุการณ์หรือรายการเปลี่ยนแปลงที่เชื่อมโยงสองเอนทิตีเกิดเป็นเอนทิตีที่สามที่เรียกว่า เอนทิตีร่วมหรือแอซโซซิเอทิฟเอนทิตี (Associative Entity) ซึ่งจะมีชุดของแอททริบิวท์และลักษณะพิเศษ (Characteristics) ของตนเอง ในตัวอย่างแรกของรูป 8-15 ความสัมพันธ์ของการลงทะเบียนเรียน แสดงถึงเอนทิตี REGISTRATION ที่บันทึกแต่ละอินสแตนส์ของนักศึกษาที่เข้าเรียนในวิชาที่ต้องการ เช่นเดียวกันความสัมพันธ์การสำรองที่นั่ง แสดงถึงเอนทิตี RESERVATION ที่บันทึกแต่ละอินสแตนส์ของผู้โดยสารที่สำรองที่นั่งในเที่ยวบินที่ต้องการ และในตัวอย่างที่สาม ความสัมพันธ์ LISTS แสดงถึงเอนทิตี ORDER-LINE ที่บันทึกแต่ละอินสแตนส์ของรายการสินค้าที่ต้องการ ดังนั้นเมื่อสร้างตารางและออกแบบเรคคอร์ดสำหรับความสัมพันธ์แบบ M:N จะต้องรวมตารางและการออกแบบตารางสำหรับแต่ละเอนทิตีสาระด้วย

ERD ที่สมบูรณ์แสดงเอนทิตีและความสัมพันธ์ของระบบทั้งหมด รูปที่ 8-16 มีห้าเอนทิตีคือ SALES REP, CUSTOMER, PRODUCT และ WAREHOUSE กับความสัมพันธ์ระหว่าง SALES REP กับ CUSTOMER ซึ่งเป็นแบบหนึ่งต่อหลายความสัมพันธ์ ในบางองค์กรลูกค้าอาจจะรับบริการจาก

ตัวแทนขายมากกว่าหนึ่งรายได้ ความสัมพันธ์จะเปลี่ยนไปเป็นแบบหลายต่อหลายความสัมพันธ์ และความสัมพันธ์ระหว่าง WAREHOUSE และ PRODUCT อาจจะเป็นความสัมพันธ์แบบหลายต่อหลายความสัมพันธ์ได้ ถ้าธุรกิจมีการจัดเก็บสินค้าหลายสินค้าในคลังสินค้าหลายแห่ง



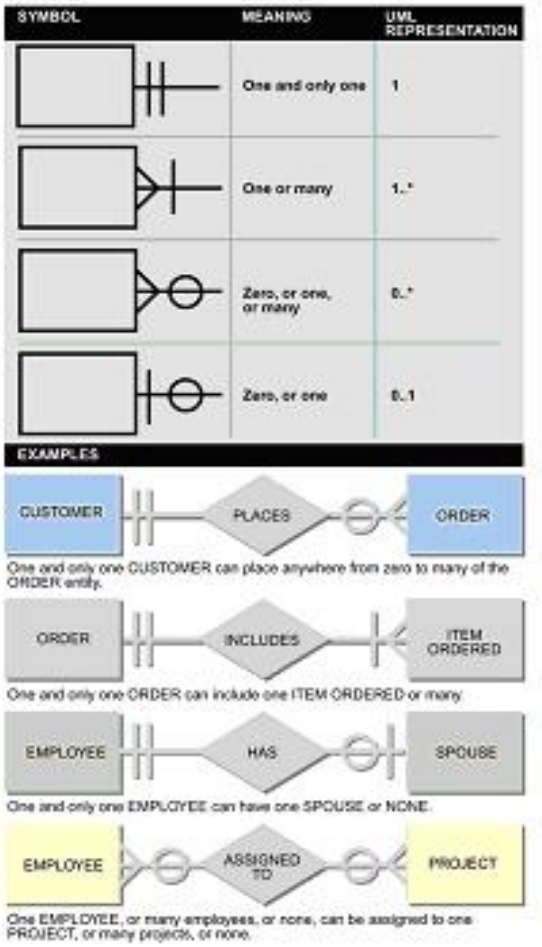

รูปที่ 5 แผนภาพความสัมพันธ์เอนทิตี ตัวแทนขาย ลูกค้า คำสั่งซื้อ สินค้า และคลังสินค้า

**คาดิแน็ลลิตี**

ธรรมชาติของความสัมพันธ์เหล่านั้นเรียกได้อีกอย่างหนึ่งว่า คาดิแน็ลลิตี (Cardinality) ในฐานะนักวิเคราะห์ระบบต้องเข้าใจคาดิแน็ลลิตี เพื่อการออกแบบแฟ้มและฐานข้อมูลที่สะท้อนถึงความสัมพันธ์ทั้งหมดระหว่างเอนทิตีได้อย่างถูกต้อง

คาดิแน็ลลิตี อธิบายถึงการที่อินสแตนส์ของหนึ่งเอนทิตีมีความสัมพันธ์กับอินสแตนส์ของอีกเอนทิตีหนึ่งได้อย่างไร ในความสัมพันธ์ที่เจาะจง เอนทิตีสามารถถูกควบคุม (Mandatory) คือ เพียงหนึ่งอินสแตนส์อนุญาตให้มีความสัมพันธ์หรือหลายอินสแตนส์ที่อนุญาตให้มีความสัมพันธ์ เช่น ความสัมพันธ์ของระหว่างสองเอนทิตี CUSTOMER และ ORDER ลูกค้ารายหนึ่งสามารถมีคำสั่งซื้อได้หลายคำสั่งหรือไม่มีการสั่งซื้อเลย และในแต่ละคำสั่งซื้อต้องเกิดจากลูกค้าเพียงรายเดียวเท่านั้น

สัญลักษณ์ของคาดิแน็ลลิตี (Cardinality Notation) มีหลายประเภท วิธีการที่นำมาใช้แสดงความสัมพันธ์ระหว่างเอนทิตี วิธีพื้นฐานแบบหนึ่งเรียกว่า สัญลักษณ์ตีนกา เนื่องจากรูปร่างที่รวมเอาวงกลม แท่ง (bar) และสัญลักษณ์ที่กำหนดความเป็นไปได้ต่างๆ โดยแท่งเดี่ยว (Single Bar) หมายถึงหนึ่ง แท่งคู่ (Double Bar) หมายถึงมีเพียงหนึ่งเดียวเท่านั้น วงกลม (Circle) หมายถึงไม่มีเลยหรือศูนย์ ส่วนตีนกาหมายถึงมีมากกว่าหนึ่ง ดังรูปที่ 8-17 แสดงสัญลักษณ์ของคาดิแน็ลลิตี ความหมาย และการใช้ UML




รูปที่ 6 ภาพบน แสดงสัญลักษณ์ตีนกา วิธีพื้นฐานของการกำหนดคาดิแน็ลลิตี และภาพล่าง แสดงสี่ตัวอย่างของการใช้สัญลักษณ์ในการอธิบายความสัมพันธ์ระหว่างสองเอนทิตี

จากรูปที่ 6 แสดงสี่ตัวอย่างของสัญลักษณ์ของคาดิแน็ลลิตี ในตัวอย่างแรก ลูกค้าหนึ่งราย สามารถสั่งสินค้าได้จากศูนย์ถึงหลายคำสั่งซื้อ ตัวอย่างที่สอง คำสั่งซื้อหนึ่งสามารถมีรายการสินค้าตั้งแต่หนึ่งรายการขึ้นไป ตัวอย่างที่สาม พนักงานหนึ่งคนสามารถมีคู่สมรสได้เพียงหนึ่งคนหรือไม่มีเลย และตัวอย่างที่สี่ หนึ่ง พนักงานหนึ่งคนหรือหลายคนหรือไม่มีเลย ได้รับการมอบหมายงานหนึ่งงานหรือหลายงานหรือไม่ได้รับการมอบหมายงานเลย

**การสร้างแผนภาพความสัมพันธ์เอนทิตี**

การสร้างแผนภาพความสัมพันธ์เอนทิตี จะง่ายขึ้น หากปฏิบัติตามสี่ขั้นตอนนี้

1. ระบุเอนทิตี ทบทวน DFD และจัดทำรายการ คน สถานที่ สิ่งของ และเหตุการณ์ สำหรับข้อมูลที่รวบรวมได้ ให้แน่ใจว่าได้กำหนดหน่วยเก็บข้อมูลทั้งหมด
2. กำหนดเหตุการณ์ รายการเปลี่ยนแปลง หรือกิจกรรมที่มีนัยสำคัญที่เกิดขึ้นระหว่างสองเอนทิตีหรือมากกว่านั้น
3. วิเคราะห์ธรรมชาติของการโต้ตอบ
4. วาด ERD สามารถวาดด้วยมือหรือเคสทูล

## การสร้างรูปแบบปกติ

การสร้างรูปแบบปกติหรือนอร์มอลไลสเซชั่น (Normalization) เป็นกระบวนการที่ระบุและแก้ปัญหาที่ฝังติดและซับซ้อนในการออกแบบเรคคอร์ด การออกแบบเรคคอร์ด (Record Design) เป็นการระบุฟิลด์และไพรมาริคีย์ ถ้ามีสำหรับเรคคอร์ดทั้งหมดในแฟ้มหรือตาราง การทำงานกับชุดแรกของการออกแบบเรคคอร์ด จะใช้นอร์มอลไลสเซชั่นในการพัฒนาการออกแบบฐานข้อมูลทั้งหมด ให้ง่าย ยืดหยุ่น และอิสระจากการซ้ำซ้อนของข้อมูล

กระบวนการนอร์มอลไลสเซชั่น แบ่งเป็นสามขั้นตอนคือ รูปแบบปกติที่หนึ่ง ที่สอง และที่สาม ทั้งสามรูปแบบปกตินั้นเป็นไปตามกฎต่อเนื่องกัน โดยการออกแบบเรคคอร์ดในรูปแบบปกติที่หนึ่งจะดีกว่ารูปแบบไม่ปกติ ในรูปแบบปกติที่สองจะดีขึ้น และในรูปแบบที่สามแสดงการออกแบบที่ดีที่สุด

### การออกแบบระเบียน

การออกแบบระเบียน (Record Design) จะง่ายขึ้นหากใช้วิธีมาตรฐานในการแสดงโครงสร้างของระเบียน ฟิลด์ และไพรมาริคีย์ จากตัวอย่างแสดงการออกแบบเรคคอร์ดที่เริ่มต้นด้วยชื่อของแฟ้มหรือตาราง ตามด้วยข้อความในวงเล็บประกอบด้วยชื่อฟิลด์แยกกันโดยจุลภาค ฟิลด์ที่เป็นไพรมาริคีย์จะมีการขีดเส้นใต้ และกลุ่มฟิลด์ซ้ำจะแสดงไว้ในวงเล็บที่สอง โดยรูปแบบจะเป็นดังนี้

NAME (FIELD 1, FIELD 2, FIELD 3, (REPEATING FIELD 4, REPEATING FIELD 5))

### รูปแบบปกติที่หนึ่ง

เรคคอร์ดจะเป็นรูปแบบปกติที่หนึ่งหรือเฟิซท์นอมอลฟอร์ม (First Normal Form : 1NF) ถ้าไม่มีกลุ่มฟิลด์ซ้ำหรือรีพีสติงกรุ๊ป (Repeating Group) ซึ่งเป็นชุดของหน่วยข้อมูลที่สามารถเกิดขึ้นได้หลายๆ ครั้งในเรคคอร์ดหนึ่ง หรือชุดของเรคคอร์ดปลีกย่อยที่บรรจุอยู่ในเรคคอร์ดหลัก

ในการแปลงเรคคอร์ดที่รูปแบบไม่ปกติให้เป็น 1NF นั้น ต้องขยายไพรมาริคีย์ โดยการรวมคีย์ของกลุ่มฟิลด์ซ้ำเข้าด้วยกันและลบกลุ่มฟิลด์ซ้ำออก

### รูปแบบปกติที่สอง

เพื่อให้เข้าใจรูปแบบปกติที่สองหรือเซคคัลนอมอลฟอร์ม (Second Normal Form : 2NF) จะต้องเข้าใจหลักการของทำงานที่ขึ้นอยู่ต่อกัน (Functionally Dependent) คือ ค่าของ X จะขึ้นอยู่กับค่าของ Y เช่น วันที่คำสั่งซื้อขึ้นอยู่กับหมายเลขคำสั่งซื้อ ในคำสั่งซื้อหนึ่งจะมีค่าเพียงค่าเดียวสำหรับวันที่คำสั่งซื้อ ในทางตรงข้ามคำอธิบายสินค้าไม่ได้ขึ้นอยู่กับเลขที่คำสั่งซื้อ สำหรับหมายเลขคำสั่งซื้อหนึ่งอาจจะมีหลายคำอธิบายสินค้า - สำหรับแต่ละคำสั่งซื้อ

ในการออกแบบเรคคอร์ด 2NF ถ้าเป็น 1NF และถ้าฟิลด์ทั้งหมดที่ไม่ได้เป็นส่วนหนึ่งของไพรมาริคีย์และขึ้นอยู่กับไพรมาริคีย์ ถ้าฟิลด์ใดๆ ในเรคคอร์ด 1NF ขึ้นอยู่กับฟิลด์หนึ่งที่เป็นส่วนของไพรมาริคีย์แล้ว เรคคอร์ดนั้นไม่เป็น 2NF เรคคอร์ด 1NF กับไพรมาริคีย์ที่เป็นฟิลด์เดียวจะอยู่ใน 2NF โดยอัตโนมัติ

ทำไมจึงต้องทำ 1NF ให้เป็น 2NF เนื่องจากปัญหาสี่อย่างที่พบกับเรคคอร์ดใน 1NF แต่ไม่พบใน 2NF

- ปัญหาแรก พิจารณาถึงงานที่จำเป็นในการเปลี่ยนคำอธิบายสินค้า สมมุติมี 1,000 คำสั่งซื้อเกิดขึ้นสำหรับสินค้าหมายเลข 304 การปรับปรุงข้อมูลทั้ง 1,000 เรคคอร์ด จะไม่สะดวกและมีค่าใช้จ่ายสูง
- ปัญหาที่สอง เรคคอร์ด 1NF ประกอบด้วยข้อมูลที่ไม่สอดคล้องกัน ไม่มีการป้องกันสินค้าหมายเลข 304 จากการมีคำอธิบายสินค้าที่แตกต่างกัน คือถ้าสินค้าหมายเลข 304 เกิดขึ้นในเรคคอร์ด ORDER 30 เรคคอร์ด อาจเกิดคำอธิบายสินค้าที่แตกต่างกัน 30 คำอธิบายขึ้นได้

- ปัญหาที่สาม เมื่อต้องการเพิ่มสินค้าใหม่สำหรับให้ลูกค้าสั่งซื้อ เนื่องจากไพรมาริคีย์ต้องรวม ORDER-NUM และ PRODUCT-NUM จึงต้องใช้ค่าของทั้งสองฟิลด์ในการสร้างเรคคอร์ดใหม่ แต่ถ้าสินค้าใหม่ที่ยังไม่มีการสั่งซื้อจากลูกค้ารายใด จะทำอย่างไร ซึ่งอาจใช้ค่าดัมมี่ (Dummy) สำหรับหมายเลขคำสั่งซื้อและแทนด้วยหมายเลขจริงเมื่อมีการสั่งซื้อ แต่ทางแก้วิธีนี้ไม่ค่อยเหมาะสมสร้างความยุ่งยาก
- ปัญหาที่สี่ เกี่ยวกับการลบข้อมูล ถ้าเรคคอร์ดที่เกี่ยวข้องถูกลบ เมื่อมีการสั่งซื้อและชำระเงินอะไรจะเกิดขึ้น หากลบเรคคอร์ดของสินค้าหมายเลข 633 ข้อมูลสารสนเทศของหมายเลขสินค้าและคำอธิบายจะหายไป

สุดท้าย วางฟิลด์ที่เหลือให้เหมาะสมกับไพรมาริคีย์ ซึ่งเป็นคีย์ที่น้อยที่สุด ให้ชื่อแต่ละเรคคอร์ด เมื่อเสร็จการวางฟิลด์ทั้งหมด ให้ลบเรคคอร์ดที่ไม่มีการวางฟิลด์เพิ่มเติม เรคคอร์ดที่เหลืออยู่จึงเป็น 2NF รูปแบบปกติที่สาม

เมื่อตารางทั้งหมดอยู่ในรูปแบบปกติที่สามหรือเธิดนอมอลฟอร์ม (Third Normal Form : 3NF) การออกแบบฐานข้อมูลจะหลีกเลี่ยงการซ้ำซ้อนของข้อมูลและปัญหาความถูกต้องตรงกันของข้อมูล กฎเกณฑ์จากการเรียนรู้ที่นิยมคือการออกแบบจะอยู่ใน 3NF ได้ถ้าทุกๆ น็อนคีย์ฟิลด์ขึ้นอยู่กับคีย์ ของคีย์ทั้งหมด และไม่เป็นอย่างอื่นนอกจากเป็นคีย์

เรคคอร์ดจัดเป็น 1NF เพราะว่าไม่มีกลุ่มซ้ำ และจัดเป็น 2NF ด้วย เพราะว่าไพรมาริคีย์เป็นฟิลด์เดียว แต่การออกแบบยังคงมีปัญหาเหมือนสี่ปัญหาของ 1NF ที่ได้อธิบายมาแล้ว การเปลี่ยนชื่อของตัวแทนขายยังคงต้องเปลี่ยนทุกๆ เรคคอร์ดที่ปรากฏชื่อตัวแทนขายนั้นอยู่ การออกแบบไม่ได้ห้ามตัวแทนขายให้มีหลายๆ ชื่อในหลายๆ เรคคอร์ด นอกจากนี้เนื่องจากชื่อตัวแทนขายยังเป็นส่วนหนึ่งของเรคคอร์ด CUSTOMER จึงต้องสร้างเรคคอร์ด CUSTOMER ที่เป็นตัวปลอมหรือดัมมี่ (Dummy) เพื่อให้สามารถเพิ่มตัวแทนขายใหม่ที่ยังไม่ได้รับการมอบหมายลูกค้าได้ แต่สุดท้าย หากลบเรคคอร์ดลูกค้าทั้งหมดของตัวแทนขายหมายเลข 22 ออก จะทำให้ชื่อและหมายเลขของตัวแทนขายหายไป

ปัญหาเหล่านี้ เนื่องจากการออกแบบเรคคอร์ดไม่เป็น 3NF การออกแบบจะเป็น 3NF ถ้าเป็น 2NF และถ้าไม่มีน็อนคีย์ฟิลด์ที่ขึ้นอยู่กับน็อนคีย์ฟิลด์อื่น จำไว้ว่าน็อนคีย์ฟิลด์เป็นฟิลด์ที่ไม่ใช่แคนดิเดทคีย์สำหรับไพรมาริคีย์ การแปลงเรคคอร์ดให้เป็น 3NF นั้น จะต้องลบทุกฟิลด์จากเรคคอร์ด 2NF ที่ขึ้นอยู่กับน็อนคีย์ฟิลด์อื่น และให้วางในเรคคอร์ดใหม่ที่ใช้น็อนคีย์ฟิลด์เป็นไพรมาริคีย์

**ขั้นตอนการออกแบบฐานข้อมูล**

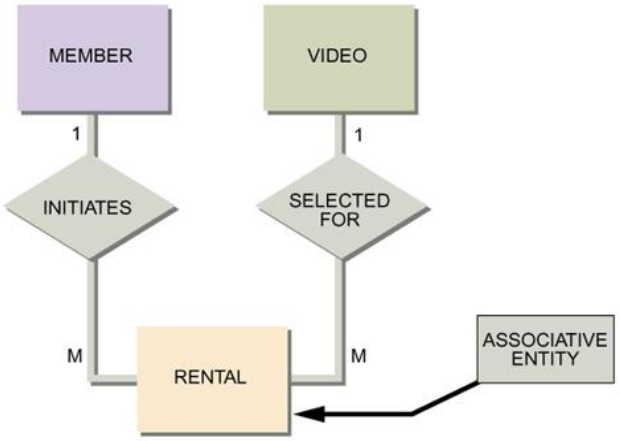
การสร้างฐานข้อมูลและออกแบบแฟ้มข้อมูล อาศัยการวิเคราะห์และออกแบบสี่ขั้นตอนต่อไปนี้

1. เริ่มต้นสร้าง ERD โดยตรวจทาน DFD และแผนภาพคลาส เพื่อระบุเอนทิตีของระบบ นอกจากนี้พิจารณาหน่วยเก็บข้อมูลที่แสดงใน DFD เพื่อดูว่าเป็นเอนทิตีหรือไม่ ถัดไปสร้างแบบร่างของ ERD โดยระมัดระวังการวิเคราะห์แต่ละความสัมพันธ์เพื่อดูว่าเป็น 1:1, 1:M หรือ M:N รูปที่ 8-30 แสดง ERD เริ่มต้นสำหรับเอนทิตี MEMBER และ VIDEO ในระบบให้เช่าวีดีโอ

MEMBER (MEMBER-NUMBER, NAME, ADDRESS, CITY, STATE, ZIP, HOME-TELEPHONE,
WORK-TELEPHONE, CREDIT-CARD-CODE, CREDIT-CARD-NUMBER, (VIDEO-ID, TITLE,
DATE-RENTED, DATE-RETURNED))
VIDEO (VIDEO-ID, TITLE)

รูปที่ 7 แผนผังความสัมพันธ์เอนทิตีเริ่มแรก และการออกแบบเรคคอร์ดที่ยังไม่ได้จัดรูปแบบปกติ ของระบบให้เช่าวีดีโอ

2. กำหนดหน่วยย่อยข้อมูลให้กับเอนทิตี ตรวจทานว่าทุกๆ หน่วยย่อยข้อมูลในพจนานุกรมข้อมูลมีความสัมพันธ์เชิงตรรกะกับเอนทิตี สำหรับระบบให้เช่าวีดีโอ การออกแบบเริ่มต้นด้วยหน่วยย่อยข้อมูลที่แสดงด้วย ERD ดังรูปที่ 8-30
3. การออกแบบ 3NF สำหรับเรคคอร์ดทั้งหมด ใช้ความระมัดระวังในการระบุทั้งไพรมาริคีย์ เซคคันเดริคีย์ และฟอรินคีย์ ผลิต ERD สุดท้ายที่รวมเอนทิตีใหม่ที่เกิดขึ้นในระหว่างนอร์มอลไลสเซชั่น รูปที่ 8-31 ที่แสดง ERD สุดท้ายและเรคคอร์ดที่เป็นรูปแบบปกติ สังเกตเอนทิตีที่สัมพันธ์ใหม่ที่เกิดระหว่างการสร้างรูปแบบปกติและความสัมพันธ์แบบ M:N ที่กลายเป็นสองความสัมพันธ์แบบ 1:M




MEMBER (MEMBER-NUMBER, NAME, ADDRESS, CITY, STATE, ZIP, HOME-TELEPHONE,
WORK-TELEPHONE, CREDIT-CARD-CODE, CREDIT-CARD-NUMBER)
VIDEO (VIDEO-ID, TITLE)
RENTAL (MEMBER-NUMBER, VIDEO-ID, DATE-RENTED, DATE-RETURNED)

รูปที่ 8 แผนผังความสัมพันธ์เอนทิตี หลังจากออกแบบเรคคอร์ดให้เป็นรูปแบบปกติ ของระบบให้เช่าวีดีโอ

4. ตรวจทานพจนานุกรมข้อมูลเอนทิตีทั้งหมด ให้แน่ใจว่าพจนานุกรมข้อมูลเอนทิตีสำหรับหน่วยเก็บข้อมูล เรคคอร์ด และส่วนย่อยข้อมูลทั้งหมดและได้จัดทำเอกสารเป็นที่เรียบร้อยอย่างถูกต้องแล้ว

## ตัวแบบฐานข้อมูล

ตัวแบบฐานข้อมูล (Database Models) ประกอบด้วย 4 แบบพื้นฐาน คือ แบบเชิงลำดับชั้น (Hierarchical) แบบเครือข่าย (Network) แบบเชิงสัมพันธ์ (Relational) และแบบเชิงวัตถุ (Object-Oriented) ฐานข้อมูลเชิงลำดับชั้นและฐานข้อมูลแบบเครือข่าย โดยปกติจะนำมาใช้ในเมนเฟรม ฐานข้อมูลเชิงสัมพันธ์สามารถนำมาใช้ในหลายๆ ฐานงานหรือเพลตฟอร์ม (Platforms) รวมทั้ง

คอมพิวเตอร์ส่วนบุคคล ฐานข้อมูลเชิงสัมพันธ์นั้นเหมาะกับไคลเอ้นท์เซิร์ฟเวอร์ เพราะว่ามีประสิทธิภาพและยืดหยุ่น ฐานข้อมูลเชิงวัตถุเป็นการใช้ส่วนจำเพาะ (Modular) ลดต้นทุนและส่วนขยายเชิงตรรกะของขบวนการวิเคราะห์เชิงวัตถุ

**ฐานข้อมูลเชิงลำดับชั้นและฐานข้อมูลแบบเครือข่าย**

สำหรับฐานข้อมูลเชิงลำดับชั้น (Hierarchical Databases) นั้น โครงสร้างของข้อมูลจะคล้ายกับต้นไม้หรือ แผนภูมิองค์กรกับสาขา ซึ่งแสดงถึงเรคคอร์ดพ่อแม่หรือแพเรนท์เรคคอร์ด (Parent Record) และเรคคอร์ดลูกหรือชายด์เรคคอร์ด (Child Record) แพเรนท์เรคคอร์ดสามารถมีได้หลายชายด์เรคคอร์ด ในแต่ละชายด์เรคคอร์ดจะมีแพเรนท์เรคคอร์ดได้เพียงเรคคอร์ดเดียว โครงสร้างถูกสร้างพร้อมกับตัวชี้ที่อยู่ เพื่อเชื่อมแพเรนท์เรคคอร์ดกับชายด์เรคคอร์ด

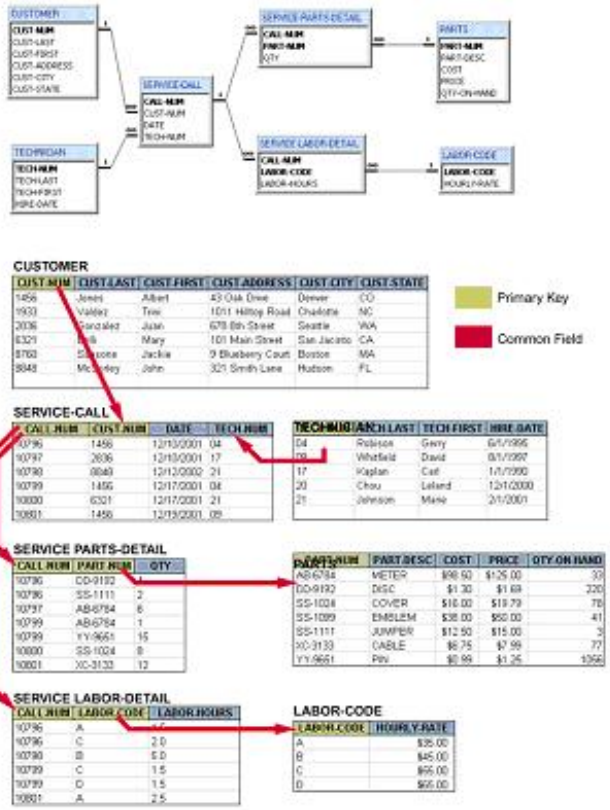
ฐานข้อมูลแบบเครือข่าย (Network Databases) ใกล้เคียงกับฐานข้อมูลเชิงลำดับชั้น แตกต่างที่มีความยืดหยุ่นกว่าในฐานข้อมูลเครือข่าย

**ฐานข้อมูลเชิงสัมพันธ์**

แบบจำลองเชิงสัมพันธ์ (Relational Model) ได้ถูกแนะนำขึ้นระหว่างปี 1970 และเป็นที่นิยมเพราะยืดหยุ่นและมีประสิทธิภาพ

เนื่องจากตารางเชื่อมโยงกัน ผู้ใช้สามารถร้องขอข้อมูลที่ตรงกับเงื่อนไขที่กำหนดได้ เพื่อให้มองเห็นวิธีการที่ระบบจัดการฐานข้อมูลเชิงสัมพันธ์จัดการคำถามที่ซับซ้อนได้ โดยอ้างถึงรูป 8-32 และพิจารณาสามตัวอย่างต่อไปนี้ สมมุติว่าผู้ใช้ต้องการที่จะเห็น ดังนี้

1. ลูกค้าทุกรายได้รับการบริการหลังจากวันที่ 12/15/2001
2. ทุกการบริการที่ใช้เวลามากกว่าสี่ชั่วโมง ของช่างเทคนิคชื่อ Marie Johnson




รูปที่ 9 ฐานข้อมูลเชิงสัมพันธ์ สำหรับธุรกิจที่ให้บริการ ณ สถานที่ของลูกค้า ที่ใช้คอมมันฟิลด์ในการจัดทำโครงสร้างข้อมูล

ในตัวอย่างแรก DBMS มองไปที่ตาราง SERVICE-CALL และค้นพบสามเรคคอร์ดที่มีวันที่หลัง 12/15/2001 แล้วใช้ CUSTOMER-NUM ซึ่งเป็นฟอรินคีย์ในตาราง SERVICE-CALL ให้ตรงกับค่าของไพรมาริคีย์ในตาราง CUSTOMER เพื่อได้ข้อมูลลูกค้า Albert Jone และ Mary Belli

ในตัวอย่างที่สอง DBMS ค้นหา Marie Johnson ในตาราง TECHNICIAN และดึงหมายเลขประจำตัวซึ่งเป็นไพรมาริคีย์ออกมา ซึ่งมีค่าเท่ากับ 21 และใช้หมายเลขนี้ให้ตรงกับ TECH-NUM ในตาราง SERVICE-CALL ค้นหาได้สองรายการเรียกใช้บริการ ถัดไประบบจัดการฐานข้อมูลใช้ค่าของ CALL-NUM คือ 10798 และ 10800 ตรวจสอบเพื่อหาเรคคอร์ดที่มีเวลามากกว่าสี่ชั่วโมงในตาราง SERVICE-LABOR-DETAIL สำหรับตัวอย่างนี้เฉพาะการเรียกใช้บริการหมายเลข 10798 เท่านั้นที่ตรงตามเงื่อนไข

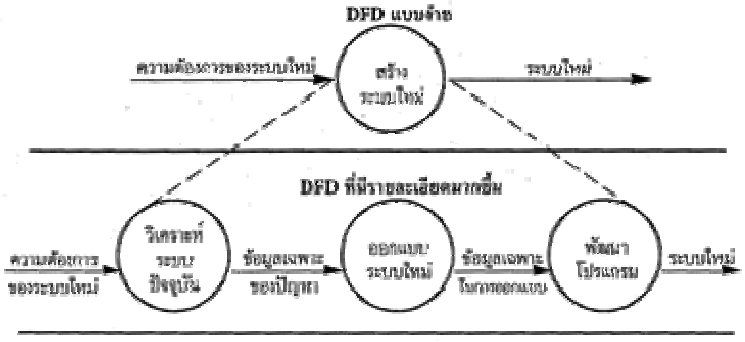
**สัปดาห์ที่** 12

Data Flow Diagram

.....

**แผนภาพกระแสข้อมูล**

แผนภาพกระแสข้อมูล (DFD) เป็นเครื่องมือที่ใช้กันอย่างแพร่หลายในการเขียนแบบระบบใหม่ โดยเฉพาะกับระบบที่ "หน้าที่" ของระบบมีความสำคัญและมีความสลับซับซ้อนมากกว่าข้อมูลที่ไหลเข้า

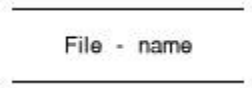



**ส่วนประกอบของ DFD** DFD มีองค์ประกอบ 4 อย่าง ซึ่งใช้สัญลักษณ์ต่าง ๆ แทน ดังต่อไปนี้

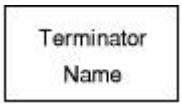
1. สัญลักษณ์แทนการประมวลผล (Process) เป็นวงกลม

2. สัญลักษณ์แทนกระแสข้อมูลเป็นลูกศร

3. สัญลักษณ์แทนแหล่งเก็บข้อมูลเป็นเส้นขนาน 2 เส้น โดยมีชื่อกำกับ




4. สี่เหลี่ยมผืนผ้าเป็นสัญลักษณ์แทนสิ่งที่อยู่นอกระบบ




การประมวลผลโพรเซส (Process) การประมวลผลโพรเซส (Process) คือ งานที่จะต้องทำแทนด้วยวงกลมและมีชื่ออยู่ภายในวงกลม เช่น

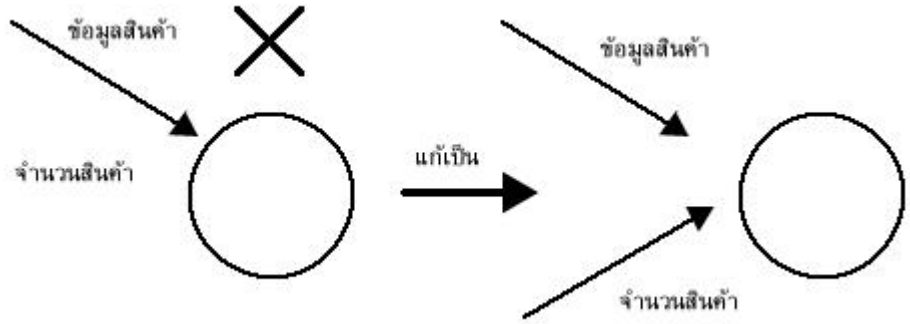
การประมวลผลจะเปลี่ยนข้อมูลขาเข้าเป็นผลลัพธ์ นั่นหมายความว่าจะต้องมีการกระทำบางอย่างต่อข้อมูลทำให้เกิดผลลัพธ์ขึ้นมา โดยปกติแล้วข้อมูลที่นำเข้าสู่โพรเซสจะแตกต่าง จากข้อมูลเมื่อออกจากโพรเซส

โพรเซสเป็นตัวอย่างหนึ่งของ "กล่องดำ" หมายถึงว่า เราทราบว่าข้อมูลเป็นอะไรผลลัพธ์อะไรที่เราต้องการ และหน้าที่โดยทั่วๆ ไปของโพรเซส แต่จะไม่ทราบว่าโพรเซสนั้นทำงาน อย่างไร หลักการของกล่องดำหลักการของกล่องดำมีประโยชน์ในการเขียน แผนภาพแสดงการเปลี่ยนแปลงของข้อมูลโดยที่ยังไม่ต้องทราบในรายละเอียดว่าโพรเซสนั้นมีรายละเอียด อะไรบ้าง ซึ่งสามารถหารายละเอียดเหล่านี้ได้ในภายหลัง

ชื่อโพรเซสเป็นตัวบอกว่าโพรเซสนั้นทำหน้าที่อะไร คำที่ใช้ควรมีความหมายที่แน่นอน ควรจะใช้คำกริยา เช่น คำนวณ แก้ไข พิมพ์ เป็นต้น ถ้าการทำงานใดที่เราไม่สามารถ หาคำแทนได้อย่างเหมาะสม อาจจะหมายความว่า งานนั้นๆ ไม่ใช่โพรเซสก็ได้

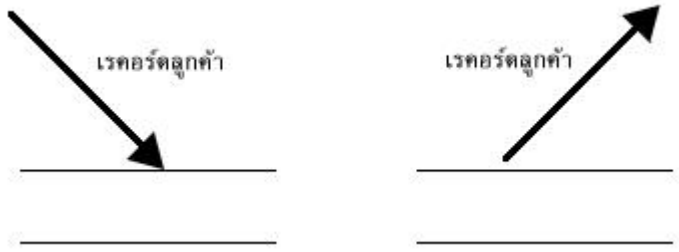
กระแสข้อมูล (Data Flow) กระแสข้อมูลแทนด้วยลูกศรโดยที่มีชื่อข้อมูลกำกับอยู่บนลูกศรนั้น

ข้อมูลที่ไหลระหว่างโพรเซสต่าง ๆ และอาจเคลื่อนที่มาจากสิ่งที่อยู่นอกระบบก็ได้ ข้อมูลที่เคลื่อนที่อาจจะเป็นเพียงข้อมูลเดี่ยวๆ เช่น เลขที่สินค้า หรือกลุ่มของข้อมูล เช่น ข้อมูล พนักงาน ข้อมูลลูกค้า เป็นต้น กลุ่มของข้อมูลควรจะเป็นเรื่องเดียวกัน หรือสัมพันธ์กัน ตัวอย่างเช่น ข้อมูลลูกค้าอาจจะมีรายละเอียดเป็นชื่อลูกค้า เลขที่ ที่อยู่ แต่ไม่ควรรวมจำนวน สินค้าในคลังอยู่ในข้อมูลเดียวกัน ถ้าต้องการอ้างอิงข้อมูลทั้งสองที่ไม่เกี่ยวข้องกันให้เขียนแยกเป็นลูกศร 2 อัน




ข้อมูลแต่ละอันหรือกลุ่มข้อมูลควรจะมีชื่อของตัวเองที่ไม่เหมือนกัน ควรหลีกเลี่ยงใช้ชื่อที่กว้างเกินไป เช่น "ข้อผิดพลาด" เพราะว่าในระบบหนึ่งๆ อาจจะมี "ข้อผิดพลาด" เกิดขึ้นหลายๆ แห่ง เราควรใช้ชื่อเฉพาะเจาะจงมากกว่านี้ เช่น "เลขที่ลูกค้าไม่ถูกต้อง" "ไม่มีสินค้านี้ในคลัง" หรือ "ไม่มีสินค้าในคลัง" เป็นต้น ในระบบใหญ่ๆ ต้องแยก รายละเอียดเหล่านี้ออกให้ชัดเจน

*แหล่งเก็บข้อมูล* (Data Store) แทนด้วยเส้นขนานสองเส้นและมีชื่อกำกับ ข้อมูลจะถูกเก็บในไฟล์และถูกเรียกใช้เมื่อต้องการ โดยปกติแล้วไฟล์อาจจะอยู่ในจานแม่เหล็ก หรือเทปแม่เหล็ก ถ้าหัวลูศรวิ่งเข้าสู่ไฟล์แสดงว่า มีการเขียนข้อมูล หรือการแก้ไขข้อมูลในไฟล์ดังในรูปข้างล่างนี้ ถ้าลูกศรวิ่งออกจากไฟล์แสดงว่ามีการอ่านข้อมูล การตั้งชื่อ ไฟล์ควรเป็นคำนาม




*สิ่งที่อยู่นอกระบบ* (Terminator) สิ่งที่อยู่นอกระบบแทนด้วยสี่เหลี่ยมผืนผ้า ซึ่งมีชื่อกำกับอยู่ด้วย ส่วนใหญ่จะเป็นตัวบุคคล หรือองค์กรต่างๆ สิ่งที่อยู่นอกระบบอาจจะเป็นที่ส่งข้อมูล เข้าสู่ระบบ หรืออาจจะเป็นที่รับข้อมูลจากระบบก็ได้ เราไม่สนใจการทำงานภายในของสิ่งที่อยู่นอกระบบ ถึงแม้ว่าจะมีการติดต่อผ่านทางข้อมูล เราจะสนใจเฉพาะข้อมูลที่เข้าสู่ระบบ หรือออกจากระบบสู่ภายนอกเท่านั้น

เมื่อเราทราบส่วนประกอบของการเขียนแผนภาพ DFD แล้ว ลองเอาสัญลักษณ์เหล่านี้มาเขียนรวมกันเป็น DFD ของระบบทั้งระบบดังนี้

DFD ระดับสูงสุด (Context Level Data Flow Diagram) เพื่อให้เข้าใจการเขียน DFD ได้ดี เราทดลองเขียนแผนภาพนี้กับระบบบัญชีเจ้าหนี้ จากรูปเป็น DFD ระดับสูงสุดของบัญชี เจ้าหนี้ซึ่งมีชื่อเรียก

อีกอย่างหนึ่งว่า "Context Diagram" ซึ่งระดับนี้จะบอกว่าระบบที่เราสนใจมีอินพุทเป็นอะไรผลลัพธ์ที่ได้จากระบบ และข้อมูลที่เกี่ยวข้องกับระบบ ภายนอก

แผนภาพระดับสูงสุดของ DFD แสดงถึงขอบเขตของระบบ ข้อมูล และผลลัพธ์ของระบบต่อไป จะกล่าวถึงวิธีการสร้าง DFD

สิ่งที่อยู่ภายนอกระบบกับ DFD ระดับสูงสุด DFD ระดับสูงสุดเท่านั้นที่จะแสดงส่วนที่อยู่ภายนอกระบบ ส่วนนี้มีความสำคัญเพราะว่าเป็นส่วนที่บอกว่าระบบนั้นๆ ได้รับข้อมูล มาจากที่ใด และผลลัพธ์ต่างๆ ถูกส่งไปที่ใดบ้าง DFD ในระดับลึกลงไปจะไม่แสดงสิ่งที่อยู่นอกระบบคือ ไม่มีสิ่งนี้เป็นส่วนประกอบ

โดยปกติหรือถ้าเป็นไปได้ เราจะวางแหล่งที่มาของข้อมูลไว้ทางซ้ายมือของ DFD และส่วนภายนอกที่รับผลลัพธ์ของระบบจะอยู่ทางขวามือ ทั้งนี้เพื่อให้อยู่ในรูปแบบของกระแสข้อมูล จากซ้ายไปขวา แต่ในหลายๆ กรณีไม่อาจเขียนตามแนวทางนี้ได้ เนื่องจากบางครั้งอินพุทและผลลัพธ์อาจเป็นสิ่งเดียวกัน ในกรณีนี้เราจะวางข้อมูลและผลลัพธ์ในที่ที่เหมาะสม ซึ่งอาจจะจะอยู่เหนือโพรเซสหรือใต้โพรเซสก็ได้

**สัปดาห์ที่** 13
**ปฏิบัติ เรื่อง การสร้าง** Data Flow Diagram

**สัปดาห์ที่** 14

- **การทดสอบระบบ**
- **การจัดทำคู่มือระบบ**
- **การฝึกอบรม**
- **การบำรุงรักษาระบบ**
- **การประเมินผล การวิเคราะห์และออกแบบระบบ**

.....

**การพัฒนาโปรแกรมและการบำรุงรักษา**

หลักการวิเคราะห์และออกแบบอย่างเดียวยังมิได้รับประกันความสำเร็จของระบบ เมื่อเสร็จสิ้นขั้นตอนการวิเคราะห์ออกแบบแล้ว เราต้องเริ่มพัฒนาโปรแกรมสำหรับระบบใหม่นี้ การพัฒนาโปรแกรมในขั้นนี้จะรวมถึงการเขียนโปรแกรม ทดสอบและปรับปรุง เพื่อให้ได้ระบบที่มีประสิทธิภาพสูงสุด ในขณะเดียวกันเราจะเริ่มอบรมผู้ใช้ และเตรียมสถานที่ ให้พร้อมสำหรับคอมพิวเตอร์ (ในกรณีที่ซื้อใหม่หรือโยกย้าย) ขึ้นถัดมาเมื่อเริ่มนำโปรแกรมที่เขียนได้มาใช้งาน จะต้องถ่าย ข้อมูลเดิมเข้าสู่ระบบใหม่นี้ แล้วจึงเริ่มต้นใช้งานระบบ ใหม่ การบำรุงรักษาในขั้นตอนการพัฒนาระบบจะรวมถึงการบำรุงรักษาประจำวัน คือ ทดสอบว่าระบบทำงานปกติ ถ้าหากพบว่ายังมีข้อบกพร่องที่จุดใด ระบบจะต้องได้รับการแก้ไข

นักวิเคราะห์ระบบจะต้องคอยเป็นพี่เลี้ยงอยู่ตลอดเวลาในขั้นตอนนี้ โดยการช่วยดูแลการเขียนโปรแกรม ทดสอบ และเตรียมสถานที่ รวมถึงมีส่วนช่วยในการอบรมผู้ใช้ และเตรียมคู่มือสำหรับผู้ใช้ด้วย ระหว่างการบำรุงรักษานักวิเคราะห์ระบบอายจะต้องรับฟังความคิดเห็นจากผู้ใช้และพร้อมที่จะแก้ไขให้เป็นไปตามความต้องการของผู้ใช้ถ้าได้รับ การอนุมัติจากผู้บริหาร รายละเอียดของกิจกรรมในขั้นตอนนี้มีดังต่อไปนี้

**การสร้างโปรแกรมและการประกันคุณภาพ**

โปรแกรมเมอร์จะทำหน้าที่เขียนโปรแกรมสำหรับระบบใหม่ทั้งหมด หรือแก้ไขโปรแกรมสำเร็จรูปถ้าซื้อโปรแกรมมา ตัวนักวิเคราะห์ระบบจะต้องกำหนดมาตรฐานของโปรแกรม โดยเขียนเป็น "คู่มือสำหรับโปรแกรมเมอร์" ซึ่งจะกำหนด มาตรฐานของโปรแกรมและเอกสารไว้ในคู่มือนี้ มาตรฐานของโปรแกรมได้แก่ การเขียนโปรแกรมจะต้องเป็นแบบ โปรแกรมโครงสร้าง การตั้งชื่อข้อมูลก็ควรให้อยู่ในรูปแบบเดียวกันคือ โปรแกรมเมอร์ทุกคนใช้ชื่อเดียวกันทั้งหมด สำหรับชื่อโปรแกรมควรจะตั้งให้มีรูปแบบเหมือนกัน เช่น ใช้ตัวอักษร 6 ตัว โดยที่สามตัวแรกเป็นตัวอักษรและสามตัวหลังเป็นตัวเลข เช่น (APY000) เป็นต้น

**การประกันคุณภาพ**

นักวิเคราะห์ระบบจะต้องรับประกันว่า โปรแกรมที่ได้มานั้นจะต้องมีข้อบกพร่องน้อยที่สุด ระหว่างแต่ละขั้นตอนเราจะต้องหาข้อบกพร่องที่อาจจะเกิดขึ้นได้ และกำจัดออกไปก่อนที่ จะก้าวสู่ขั้นตอนถัดไป

เพราะว่าข้อบกพร่องมีอยู่ในระบบมากเท่าใด ก็จะทำให้ราคาในการแก้ข้อบกพร่องมีมากเท่านั้น และจะขึ้นอยู่กับระยะเวลาของการพัฒนาระบบด้วยว่ามีข้อ บกพร่องเกิดขึ้นนานเท่าไรแล้วด้วย ซึ่งค่าใช้จ่าวในการแก้ไขระบบจะเพิ่มขึ้นด้วยอัตราแบบ "Exponential" ตัวอย่างเช่น พบว่าลืมตรวจสอบอินพุตที่สำคัญตัวหนึ่งถ้าอยู่ในขั้น วิเคราะห์ระบบและจะแก้ไขจุดบกพร่องนี้จะเสียค่าใช้จ่ายประมาณ 10 บาท และอาจจะเพิ่มเป็น 100 บาท ถ้าพบข้อบกพร่องนี้ในการออกแบบ และเพิ่มเป็น 1,000 บาท ถ้าพบในขั้นตอนเขียนโปรแกรมและทดสอบระบบ และอาจจะสูงขึ้นถึง 10,000 บาท ถ้าพบหลังจากนำโปรแกรมไปใช้งานแล้ว สรุปแล้วก็คือ แก้ไขในกระดาษนั้นง่ายและถูกกว่าแก้ไข ในโปรแกรมอย่างแน่นอน

**ทบทวน**

การทบทวนจะช่วยให้หาข้อบกพร่องได้ในขั้นตอนแรกๆ ของการพัฒนาระบบ ปกติการทบทวนจะทำงานกันเป็นทีมซึ่งแต่ละคนจะทำหน้าที่ต่างๆ กัน ทีมทบทวนจะประกอบด้วย ผู้นำเสนอ ผู้ประสานงาน เลขา ผู้ดูแลการบำรุงรักษา ผู้กำหนดมาตรฐาน และตัวแทนจากผู้ใช้ ผู้นำเสนอ มักจะเป็นผู้เริ่มโครงการนั้นๆ ผู้ประสานงานทำหน้าที่จัดการเกี่ยวกับ กิจกรรม ต่างๆ ก่อนการทบทวน เช่น ส่งรายงานให้ทุกคนที่ร่วมทีม และเป็นทีมผู้ประสารงานระหว่างที่ทบทวนระบบด้วย เลขา ทำหน้าที่จดบันทึกและส่งรายงานให้ผู้บริหารหลังจาก การทบทวน ผู้ดูแลการบำรุงรักษาจะตรวจสอบโปรแกรมและคาดคะเนว่าในอนาคตจะมีปัญหาเกิดขึ้นได้บ้าง ผู้กำหนดมาตรฐานจะทำหน้าที่ตรวจสอบว่าโปรแกรมที่เขียนไปเป็น ไปตามมาตรฐานหรือไม่ ผู้แทนจากผู้ใช้จะตรวจสอบว่าโปรแกรมทำงานตามที่ต้องการหรือไม่ การทบทวนควรจะทำในทุกๆ ขั้นตอนของการวิเคราะห์และออกแบบระบบ การเขียน โปรแกรมและเขียนคู่มือผู้ใช้ การทบทวนจะช่วยลดข้อบกพร่องลงจาก 3-5 จุดใน 100 บรรทัดของโปรแกรม เหลือเพียง3-5 จุดใน 1,000 บรรทัดของโปรแกรม

**การทดสอบระบบ**

ถึงแม้ว่าเราจะทบทวนระบบแล้วก็ไม่ได้หมายความว่าระบบจะไม่มีข้อบกพร่องอีกแล้ว ดังนั้นเราจะต้องทดสอบระบบให้ถี่ถ้วนอีกที่หนึ่ง ปกติแล้วนักวิเคราะห์ระบบจะไม่ทดสอบ ระบบด้วยตัวเอง แต่จะเป็นคนวางแผนและควบคุมการทดสอบ

**ทดสอบระบบรวม**

หลังจากระบบเสร็จสมบรูณ์แล้ว เราจะต้องทดสอบระบบรวมทั้งหมดอีกครั้งหนึ่ง ซึ่งจะช่วยให้เราหาข้อผิดพลาดอะไรเกิดขึ้นเราจะต้องทำการแก้ไขแล้วทดสอบระบบรวมใหม่ อีกครั้งหนึ่ง หลังจากนั้นทำการทดสอบระบบเพื่อส่งมอบงาน (Acceptance Test) ให้ลูกค้าได้หรือผู้ใช้ได้เห็นระบบการทำงานตามที่ต้องการ ส่วนการทดสอบลำดับสุดท้ายคือ การทดสอบแบบขนาน (Parallel Operation) ซึ่งหมายความว่าระบบใหม่จะทำงานไปพร้อมๆ ระบบเดิมโดยการใช้ข้อมูลจริงที่เหมือนกัน แล้วเปรียบเทียบผลลัพธ์ที่ได้จากระบบทั้งสอง ถ้าผลลัพธ์แตกต่างกันในเวลาหนึ่งเวลาใดเราจะต้องตรวจสอบว่าระบบใหม่มีปัญหา

อะไรแล้วทำการแก้ไขให้ถูกต้อง และทำการเตรียมสถานที่ (Site Preparation) ที่เหมาะสมกับการ ใช้งานต่อไป

**คู่มือผู้ใช้และการอบรมผู้ใช้**

นักวิเคราะห์ระบบจะต้องมีส่วนเกี่ยวข้องในการเตรียมคู่มือผู้ใช้และจัดเตรียมการอบรมผู้ใช้ในบางครั้ง นักวิเคราะห์ระบบจะเป็นผู้ดูแลงานเหล่านี้เอง หรือบางครั้งก็ทำให้ทำเสีย เองก็มี

**คู่มือผู้ใช้**

ผู้ใช้จะมีอยู่ 2 กลุ่มคือ ผู้ที่ไม่มีความรู้เกี่ยวกับระบบเลย และผู้ที่มีประสบการณ์แต่ต้องใช้คู่มือเมื่อต้องการหาอะไรบางอย่างเป็นบางครั้ง ดังนั้นคู่มือผู้ใช้ที่มีรายละเอียดทุกอย่าง ตั่งแต่เริ่มต้นจะเหมาะสมกับผู้ใช้คนแรก และคู่มือแบบสั้นๆ ที่รวบรวมรายละเอียดไว้ทั้งหมดจะเหมาะสำหรับผู้ใช้กลุ่มหลังดังนั้นคู่มือผู้ใช้ควรจะเขียนขึ้นในลักษณะที่ช่วยให้ผู้ใช้ ทำความเข้าใจได้ง่ายทีละขั้นๆ แล้วลงท้ายด้วยการรวบรวมรายละเอียดอย่างสั้นๆ ที่เรียกใช้ได้อย่างรวดเร็ว

**การอบรมผู้ใช้**

ในกรณีที่เป็นระบบนั้นเป็นระบบใหม่ทั้งหมด ดังนั้นคู่มือการใช้อย่างเดียวนั้นจะไม่เพียงพอให้ผู้ใช้เริ่มใช้งานระบบได้ด้วยตัวเอง เนื่องจากระบบนั้นเป็นระบบใหม่ เราจะต้อง อบรมผู้ใช้หลายคนที่เดียวในระยะเวลาสั้นๆ การอบรมที่ละคนจะเป็นวิธีที่ดีที่สุดเพราะว่าจะชี้แจงรายละเอียดและการทำความเข้าใจจะเป็นอย่างสะดวก ในขณะที่การอบรมกลุ่ม จะเสียเวลาน้อยกว่า

**นำมาใช้งานจริง**

การนำระบบใหม่มาใช้งานจริงจะต้องถ่ายข้อมูลจริงเข้าสู่ระบบใหม่ให้หมด แล้วจึงเริ่มให้ระบบใหม่ปฏิบัติงาน นักวิเคราะห์ระบบจะต้องคอยดูแลอยู่ตลอดเวลาเพื่อป้องกันไม่ให้ ระบบที่ใช้อยู่ปัจจุบันกระทบกระเทือน

**การถ่ายเทข้อมูล**

วิธีการถ่ายเทข้อมูลจากไฟล์ที่มีอยู่แล้วจากระบบเก่าเข้าสู่ระบบใหม่วิธีหนึ่งคือ ใช้พนักงานป้อนข้อมูลเข้าไปในระบบใหม่ วิธีนี้แน่นอนที่สุดจะต้องเสียเวลามากและเปลืองค่าใช้จ่าย แต่ถ้าเป็นการเปลี่ยนจากระบบที่ไม่ได้ใช้คอมพิวเตอร์ จะเป็นแบบที่ใช้คอมพิวเตอร์วิธีนี้เป็นวิธีที่หลีกเลี่ยงไม่ได้

ระหว่างการถ่ายเทข้อมูล ควรจะต้องมีการควบคุมการถ่ายเทนี้ด้วย เช่น จำนวนข้อมูลเดิมมีอยู่เท่าไร เมื่อถ่ายเทเข้าระบบใหม่จำนวนข้อมูลควรจะมีเท่ากันเป็นต้น ถ้าไม่ตรงกันจะต้องพิมพ์ออกมาแล้วหาด้วยมืออีกครั้งหนึ่ง

**เริ่มใช้ระบบใหม่**

วิธีที่ถูกที่สุดในการนำระบบใหม่มาใช้งานจริงก็คือ ยกเลิกระบบเก่าในทันทีแล้วใช้ระบบใหม่เข้าแทนที่

วิธีนี้เสี่ยงในกรณีที่ระบบใหม่มีปัญหา เราก็จะไม่มีระบบเก่ามา เปรียบเทียบเพื่อหาข้อผิดพลาด

อีกวิธีหนึ่งในการนำมาใช้งานจริงก็คือ การทำงานแบบขนาน (Parallel Operation) ซึ่งได้กล่าวมาแล้ว วิธีนี้เป็นวิธีที่ปลอดภัยที่สุด แต่ก็เสียค่าใช้จ่ายมากที่สุดด้วย เมื่อทำงานขนาน ไปสักพักจนครบวงจรธุรกิจ เราก็ตัดระบบเก่าออกไปได้

**การบำรุงรักษาและเปลี่ยนแปลงการควบคุม**

เมื่อนำระบบใหม่มาใช้งานจริงแล้ว เราต้องเผชิญหน้ากับการเปลี่ยนแปลงถึงแม้ว่าเราต้องการจะให้ระบบเป็นอยู่อย่างที่เห็น แต่ก็เป็นไปไม่ได้ ทั้งนี้เพราะโลกนี้ไม่เคยหยุดนิ่ง เมื่อใช้ไปสักพัก ผู้ใช้จะต้องขอให้มีส่วนนั้นส่วนนี้เพิ่มเติม หน่วยงานจริงๆ เปลี่ยนแปลงนโยบายหรือต้องการรายงานบางอย่างเพิ่มเติม คอมพิวเตอร์ฮาร์ดแวร์และซอฟแวร์ใหม่ๆ ออกสู่ตลาดไม่ว่าจะเป็นด้วยสาเหตุอะไร ระบบต้องได้รับการเปลี่ยนแปลง ซึ่งนักวิเคราะห์ระบบจะต้องเตรียมระบบที่จะต้องรับการเปลี่ยนแปลงที่เพิ่มขึ้นได้ เมื่อรวมโปรแกรมที่เพิ่มเติมเข้ามาในระบบแล้ว ส่วนอื่นๆ ที่เกี่ยวข้องจะต้องแก้ไขตามไปด้วย ตัวอย่างเช่น ข้อมูลเฉพาะของปัญหาและการออกแบบจะต้องได้รับการแก้ไข ให้รวมของใหม่ไปด้วย

เมื่อรวมโปรแกรมแล้วเราจะต้องทดสอบว่าระบบทำงานต่อไปนี้ได้อย่างถูกต้องหรือไม่

1. ทดสอบการทำงานตามหน้าที่ (Functional Testing) เป็นการทดสอบว่าโมดูลทำงานตามที่ออกแบบไว้หรือไม่ เช่น โมดูลแยกสถานะใบทวงหนี้ก็จะต้องแยกสถานะได้ถูกต้อง ตามเงื่อนไขในโปรแกรมเป็นต้น

2. ทดสอบการกู้ข้อมูล (Recovery Testing) เป็นการทดสอบว่าระบบสามารถดึงข้อมูลทั้งเก่า และใหม่กลับคืนมาได้ เมื่อมีปัญหาเกิดขึ้น เช่น ไฟล์ระหว่างที่ใช้โปรแกรมอยู่ การทดสอบนี้สำคัญมากสำหรับระบบ On-line ใหญ่ๆ เช่น ธนาคาร เป็นต้น

3. ทดสอบสมรรถภาพ (Performance Testing) เป็นการทดสอบว่าระบบจะทำงานและให้คำตอลในเวลาที่ รวดเร็วตามที่ออกแบบไว้

**การวางแผนและควบคุมโครงการ**

การทำงานด้านการวิเคราะห์และออกแบบระบบมีขั้นตอนการทำงานหลายขั้นตอนด้วยกัน ซึ่งจะต้องดำเนินไปอย่างมีระบบ นักวิเคราะห์ระบบควรจะต้องทราบว่าจะจัดการโครงการ ให้ดำเนินการไปอย่างมีระบบได้อย่างไร เมื่อนักวิเคราะห์ระบบเริ่มต้นทำงาน การวางแผนโครงการ และการควบคุมโครงการให้ดำเนินไปตามแผนต้องเริ่มต้นในทันทีเช่นเดียวกัน เรามาพูดเรื่องนี้ในหัวข้อสุดท้ายก็เนื่องจากความจริงที่ว่าเราวางแผนงานไม่ได้ ถ้ายังไม่ทราบว่าการวิเคราะห์ระบบนั้นต้องทำอะไรบ้าง ดังนั้นเมื่อเราเรียนรู้การวิเคราะห์และออก แบบระบบในตอนต้นแล้ว มาถึงหัวข้อนี้เราจึงพร้อมที่จะกล่าวถึงการ วางแผนและควบคุมโครงการ

**การวางแผนโครงการ**

การวางแผนโครงการคือ ความพยายามที่จะคาดคะเนเวลาและค่าใช้จ่ายที่จะใช้ในการดำเนินงานโครงการใด โครงการหนึ่งรวมทั้งผลประโยชน์ที่จะได้รับจากโครงการ ด้วยแผนงานของโครงการจะรวมถึงขั้นตอนการทำงาน กิจกรรมที่จะต้องทำ เวลาที่ใช้ในแต่ละกิจกรรม รวมทั้งบุคคลากรที่เหมาะสมในแต่ละกิจกรรมด้วย แต่ละโครงการควรจะ วางแผนในรายละเอียดให้มากก่อนที่จะเริ่มทำงานจริง และเมื่อดำเนินงานจริงๆ แล้วควรจะติดตามและควบคุมให้เป็นไปตามแผนที่ วางไว้ด้วย

แผนงานของโครงการวิเคราะห์และออกแบบระบบจะประกอบแผนงานย่อยของกิจกรรมต่อไปนี้คือ การวิเคราะห์ การออกแบบ การพัฒนาโปรแกรม เตรียมเอกสาร ฝึกอบรม และการนำระบบมาใช้งานจริง แต่ละกิจกรรมก็จะประกอบด้วยงานย่อยๆ แยกไปอีก และที่จะกล่าวต่อไป ในหัวข้อนี้ได้แก่ การคาดคะเนเวลา และเตรียมตารางการทำงาน คาดคะเน ค่าใช้จ่าย ผลประโยชน์ที่ได้รับ และการวิเคราะห์ค่าใช้จ่ายและผลประโยชน์

**การควบคุมโครงการ**

การควบคุมโครงการจะตรงข้ามกับการวางแผนโครงการ การควบคุมโครงการเป็นการติดตามการทำงานเมื่อ โครงการเริ่มต้นแล้ว ซึ่งจะทำให้ทราบว่าการทำงานเป็นไปตาม จุดประสงค์ของแผนงานหรือไม่ เราจะต้องติดตามการทำงานอย่างต่อเนื่องเพื่อควบคุมการทำงานให้เป็นไปตามตารางแผนงานและค่าใช้จ่ายให้อยู่ในงบประมาณที่ตั้งไว้ การติดตาม การทำงานอย่างต่อเนื่องช่วยให้เราค้นพบปัญหาหนึ่งได้ทันที (ถ้ามี) และจะทำให้การแก้ไขเป็นไปได้ง่ายและประหยัดค่าใช้จ่ายด้วย

**ระบบรักษาความปลอดภัยภายในระบบงาน** (System Security And Integrity)

ในปัจจุบันระบบงานคอมพิวเตอร์แบบเครือข่ายได้ทำให้การใช้ข้อมูลต่างๆ ของหน่วยงานเป็นไปอย่างมีประสิทธิภาพอย่างไม่เคยมีมาก่อน การกระจายอำนาจการใช้ข้อมูลออกไป (Distribution System) ของระบบงาน ทำให้ระบบจำเป็นที่จะต้องมีระบบการรักษาความปลอดภัยภายในระบบงานอย่างดีพอด้วย เพื่อป้องกันไม่ให้ผู้ที่ไม่มีสิทธิ์ใช้ข้อมูล สามารถเข้าถึงข้อมูลได้ นอกจากนี้นักวิเคราะห์ระบบยังต้องให้ความสนใจต่อความถูกต้อง (Integrity) ของระบบ เช่น ระบบงานต่างๆ โปรแกรมและฐานข้อมูลอีกด้วย ในที่นี้เราจะกล่าวถึงวิธีที่นิยมทำกันโดยทั่วไป ซึ่งมี 4 วิธี คือ

**การใช้รหัสลับ** (Passwords) เป็นวิธีการที่นิยมใช้กันโดยทั่วไป โดยมีวัตถุประสงค์ที่จะเข้าไปทำงานในระบบนั้นๆ ได้ ระบบก็จะปฏิเสธ การยอมให้เข้าถึงข้อมูลของระบบโดยอัตโนมัติ ในบางระบบนอกจากการปฏิเสธแล้ว ระบบยังทำการบันทึกชื่อและเวลา และเบอร์โทรศัพท์ที่อาจใช้เรียกเข้ามาของผู้ที่ตอบรหัสลับผิดเอาไว้เป็นข้อมูลเพื่อติดตามภายหลังอีกด้วย

**การสำรองข้อมูล** (System Backups) ในทุกระบบงานที่ดี การวางตารางเวลาเพื่อการสำรองข้อมูลเป็นสิ่งสำคัญอย่างยิ่งที่จะป้องกันปัญหาของการสูญเสียข้อมูลในกรณีที่ไม่คาดฝันเกิดขึ้น การสำรองข้อมูลอาจเลือกใช้เทปหรือ Removable Disk ก็ได้ ซึ่งแล้วแต่ความเหมาะสม

การสำรองข้อมูลมีด้วยกัน 2 แบบ คือ แบบเต็ม (Full) หรือแบบเฉพาะส่วนเพิ่ม (Incremental) โดยการสำรองข้อมูลชนิดเต็ม หมายถึง การสำรองข้อมูลจะทำการสำรองใหม่หมดทุกครั้ง แม้ว่าข้อมูลนั้นจะได้เคยทำการสำรองไว้แล้วก็ตาม ส่วนแบบการสำรองเฉพาะส่วนเพิ่มนั้น เราจะเลือกสำรองข้อมูลในส่วนที่แตกต่างหรือเพิ่มเติมจากส่วนที่เราได้ทำการสำรองไว้ในครั้งก่อนเท่านั้น ซึ่งวิธีนี้จะทำให้เราประหยัดเวลาในการสำรองข้อมูลลงได้

ไม่ว่าการสำรองข้อมูลจะเป็นแบบใด หรือจะใช้อะไรเพื่อการสำรองก็ตาม ระบบงานที่ดีจะต้องมีการสำรองข้อมูลไว้อย่างสม่ำเสมอ และเพื่อเป็นการค้ำประกันต่อความปลอดภัยของข้อมูลเอง การสำรองข้อมูลควรจะทำไว้อย่างน้อย 2 ชุดคือ ชุดที่ 1 ควรจะเก็บเอาไว้ในที่ที่ระบบงานอยู่ ส่วนชุดที่ 2 ก็ควรจะเก็บเอาไว้นอกเขตที่ระบบงานอยู่ เช่น ในเซฟธนาคาร หรือที่สาขาของสำนักงานในเขตอื่น เพื่อในกรณีที่เกิดเหตุการณ์ที่ไม่คาดหมาย เช่น ไฟไหม้ น้ำท่วม ข้อมูลชุดที่ 2 ก็ยังคงปลอดภัย และสามารถนำกลับมาใช้ได้อีก

**การตรวจสอบได้ของระบบ** (Audit Trail) ระบบงานที่ดีควรได้รับการดีไซน์ให้ข้อมูลต่างๆ ที่เกี่ยวข้องในระบบสามารถที่จะตรวจสอบย้อนกลับได้ ว่าเกิดได้อย่างไร มาจากไหน วิธีที่นิยมใช้กันก็คือ การออกรายงานหรือ Check List ต่างๆ ที่แสดงถึงเหตุการณ์หรือข้อมูลต่างๆ ที่ถูกเรียกขึ้นมาเพื่อแก้ไขหรืออินพุตเข้ามาในระบบ เอกสารต่างๆ เหล่านี้จะมีความจำเป็นอย่างมากต่อการตรวจสอบความถูกต้องของการทำงานของระบบงานคอมพิวเตอร์

**การเรียกคืนข้อมูลและเริ่มต้นใหม่ของระบบ** (Recovery And Restart Needs) ในระบบงานคอมพิวเตอร์ ไฟฟ้าเป็นสิ่งสำคัญอย่างยิ่ง นักวิเคราะห์ระบบหรือโปรแกรมเมอร์ทุกคนรู้ดีว่าหากไฟฟ้าดับหรือเกิดการลัดวงจรหรือฟ้าผ่าเข้ามาในสายไฟฟ้า จะส่งผลทำให้ระบบคอมพิวเตอร์ทั้งระบบเกิดความเสียหายอย่างใดอย่างหนึ่ง เมื่อระบบงานเกิดความเสียหาย (Crash) ขึ้น การนำเอาข้อมูลที่ได้สำรองเอาไว้ และการเรียกคืนข้อมูล (Restore Data) เพื่อจะทำให้ระบบฟื้นคืนชีพกลับมาใหม่อาจต้องเกิดขึ้น

เหล่านี้จึงเป็นหน้าที่ของนักวิเคราะห์ที่จะต้องวางแผนงานของการเรียกคืนข้อมูลและโปรแกรมต่างๆ ให้ดีพอที่จะทำให้ระบบงานสามารถฟื้นคืนชีพในเวลาอันสั้นด้วย ในช่วงของการดีไซน์ระบบรักษาความปลอดภัยนี้

**สัปดาห์ที่** 15 – 17
**ปฏิบัติงานกลุ่ม และนำเสนองานกลุ่ม**