**การวิเคราะห์และออกแบบระบบ**

ความรู้ในการวิเคราะห์และออกแบบระบบมีความสำคัญ เพราะเป็นปัจจัยในการสร้างและพัฒนาระบบสารสนเทศ การวิเคราะห์ระบบเป็นเรื่องที่น่าสนใจ เพราะนักวิเคราะห์ ระบบต้องติดต่อกับคนหลายคน ได้รู้ถึงการจัดการและการทำงานในองค์การทำให้เรามีความรู้เกี่ยวกับระบบคอมพิวเตอร์หลายแบบมากขึ้น ผู้ที่สามารถวิเคราะห์ระบบได้ดีควร มีประสบการณ์ ในการเขียนโปรแกรม มีความรู้ทางด้านธุรกิจ ความรู้เกี่ยวกับระบบเครือข่ายและฐานข้อมูล ซึ่งใช้เป็นความรู้ในการออกแบบระบบที่มีความแตกต่างกันออกไปตาม สภาพงาน ดังนั้นหน้าที่ของนักวิเคราะห์ก็คือการศึกษาระบบ แล้วให้คำแนะนำในการปรับปรุงและพัฒนาระบบนั้นจนเสร็จสมบูรณ์ ซึ่งการทำงานทั้งหมดต้องมีลำดับขั้นตอน และการ ศึกษาวิธีการวิเคราะห์และการออกแบบระบบในแต่ละขั้นตอนทำให้เราเข้าใจระวิเคราะห์ระบบนั้นๆ ดียิ่ง และสามารถออกแบบระบบใหม่โดยไม่ยากเย็นนัก โดยสามารถตัดสินใจว่า ระบบใหม่ควรใช้คอมพิวเตอร์ประเภทไหน ใช้โปรแกรมอะไร ออกแบบInput/Output อย่างไรเป็นต้น

ระบบคือกลุ่มขององค์ประกอบต่างๆ ที่ทำงานร่วมกันเพื่อจุดประสงค์อันเดียวกันระบบอาจจะประกอบด้วย บุคคลากร เครื่องมือ เครื่องใช้ พัสดุ วิธีการ ซึ่งทั้งหมดนี้จะต้องมีระบบ จัดการอันหนึ่งเพื่อให้บรรลุจุดประสงค์อันเดียวกัน เช่น ระบบการเรียนการสอน มีจุดประสงค์เพื่อให้นักเรียนได้รับความรู้ในเนื้อหาวิชาที่สอน

6.1 ตัวอย่างระบบการเรียนการสอน

การวิเคราะห์ระบบและการออกแบบ (System Analysis and Design) การวิเคราะห์และออกแบบระบบคือ วิธีการที่ใช้ในการสร้างระบบสารสนเทศขึ้นมาใหม่ในธุรกิจใดธุรกิจหนึ่ง หรือระบบย่อยของธุรกิจ นอกจากการสร้างระบบ สารสนเทศใหม่แล้ว การวิเคราะห์ระบบช่วยในการแก้ไขระบบสารสนเทศเดิมที่มีอยู่แล้วให้ดีขึ้นด้วยก็ได้ การวิเคราะห์ระบบก็คือ การหาความต้องการ (Requirements) ของระบบสารสนเทศว่าคืออะไร หรือต้องการเพิ่มเติมอะไรเข้ามาในระบบ และการออกแบบก็คือ การนำเอาความต้องการของระบบมาเป็น แบบแผน หรือเรียกว่าพิมพ์เขียวในการสร้างระบบสารสนเทศนั้นให้ใช้งานได้จริง ตัวอย่างระบบสารสนเทศ เช่น ระบบการขาย ความต้องการของระบบก็คือ สามารถติดตามยอดขาย ได้เป็นระยะ เพื่อฝ่ายบริหารสามารถปรับปรุงการขายได้ทันท่วงที ตัวอย่างรายงานการขายที่กล่าวมาแล้วจะชี้ให้เห็นว่าเราสามารถติดตามการขายได้อย่างไร

นักวิเคราะห์ระบบ (System Analyst หรือ SA) นักวิเคราะห์ระบบคือ บุคคลที่มีหน้าที่วิเคราะห์และออกแบบระบบ ซึ่งปกติแล้วนักวิเคราะห์ระบบควรจะอยู่ในทีมระบบสารสนเทศขององค์กร หรือธุรกิจนั้นๆ การที่มีนักวิเคราะห์ระบบในองค์กรนั้นเป็นการได้เปรียบเพราะจะรู้โดยละเอียดว่า การทำงานในระบบนั้นๆ เป็นอย่างไร และอะไรคือความต้องการของระบบ ในกรณีที่นักวิเคราะห์ระบบไม่ได้อยู่ในองค์กรนั้น ก็สามารถวิเคราะห์ระบบได้ เช่นกันโดยการศึกษาสอบถามผู้ใช้ และวิธีการอื่นๆ ซึ่งจะกล่าวในภายหลัง ผู้ใช้ในที่นี้ก็คือเจ้าของ และผู้ที่เกี่ยวข้องในระบบสารสนเทศนั้นเอง ผู้ใช้อาจจะเป็น คนเดียว หรือหลายคนก็ได้ เพื่อให้นักวิเคราะห์ระบบทำงานได้อย่างคล่องตัวมีลำดับขั้นและเป้าหมายที่แน่นอน นักวิเคราะห์ระบบควรทราบถึงว่า ระบบสารสนเทศนั้นพัฒนาขึ้นมาอย่างไร มีขั้นตอนอย่างไรบ้าง

**การจัดการข้อมูล**

วงจรการพัฒนาระบบ (System Development Lift Cycle) ระบบสารสนเทศทั้งหลายมีวงจรชีวิตที่เหมือนกัน ตั้งแต่เกิดจนตาย วงจรนี้จะเป็นขั้นตอนที่เป็นลำดับตั้งแต่ต้นจนเสร็จ เรียบร้อย เป็นระบบที่ใช้งานได้ ซึ่งนักวิเคราะห์ระบบต้องทำความเข้าใจให้ดีว่าในแต่ละขั้นตอนจะต้องทำอะไร และทำอย่างไร ขั้นตอนการพัฒนาระบบมีอยู่ด้วยกัน 7 ขั้นตอนคือ
1. เข้าใจปัญหา (Problem Recognition)
2. ศึกษาความเป็นไปได้ (Feasibility Study)
3. วิเคราะห์ (Analysis)
4. ออกแบบ (Design)
5. สร้าง หรือพัฒนาระบบ (Construction)
6. การปรับเปลี่ยน (Conversion)
7. บำรุงรักษา (Maintenance)

**การวิเคราะห์**

การวิเคราะห์ระบบในวงจรการพัฒนาระบบนั้นเริ่มต้นจากการศึกษาระบบเดิม แล้วนำข้อมูลที่ได้จากการศึกษา มาหาความต้องการ (Requirements) หรือสิ่งที่จะต้องปรับปรุง ในระบบหรืออีกอย่างหนึ่งคือ วิธีแก้ปัญหาของระบบ การวิเคราะห์จะเริ่มหลังจากที่ทราบปัญหา และผ่านขั้นตอนการศึกษาความเป็นไปได้แล้ว *รวบรวมข้อมูล* การศึกษาระบบเดิมนั้น นักวิเคราะห์ระบบเริ่มต้นจากการศึกษาเอกสารต่างๆ เช่น คู่มือต่างๆ หลังจากนั้นเป็นการรวบรวมแบบฟอร์มและรายงานต่างๆ เช่น ในระบบบัญชีเจ้าหนี้จะมีแบบฟอร์มใบบรรจุ ผลิตภัณฑ์ ใบทวงหนี้ รายงานเพื่อเตรียมเงินสนเป็นต้น นอกจากนั้นจะต้องคอยสังเกตดูการทำงานของผู้ที่เกี่ยวข้องในระบบที่ศึกษา ท้ายที่สุดอาจจะต้องมีการสัมภาษณ์ผู้ที่มีหน้าที่ รับผิดชอบงานที่เกี่ยวข้องในระบบ หรือบางกรณีอาจจะต้องใช้แบบสอบถามมาช่วยเก็บข้อมูลด้วยก็ได้ วิธีการทั้งหมดเรียกว่า เทคนิคการเก็บ

ติดตาม การเปลี่ยนแปลงของข้อมูลเรียกว่า แผนภาพแสดงกระแสข้อมูล (Data Flow Diagram : DFD)

**การจัดการข้อมูล**

*คำอธิบายการประมวลผล* (Process Description) ต้องมีเพราะถึงแม้ว่าแผนภาพแสดงกระแสข้อมูล (DFD) เป็นเครื่องมือพื้นฐานที่ใช้ในการวิเคราะห์ว่าจะต้องทำอะไรบ้าง หรือจะต้องประมวลผลอะไรบ้างแต่ในแต่ละขั้นตอนถึงแม้จะ แยกย่อยลงมา แล้วก็ยังมีรายละเอียดที่ลึกลงไปอีก การประมวลผลที่ลึกลงไปนั้นใน DFD อธิบายรายละเอียดด้วย "คำอธิบายการประมวลผล" (Process Description) คำอธิบายนี้บอกอย่างแน่ชัดว่าอินพุตถูกเปลี่ยนเป็นเอาต์พุตอย่างไร ตัวอย่างการเปลี่ยนยางรถยนต์ในบทที่แล้วในขั้นตอน ขยับรถยนต์ขึ้นเราจะเขียนเป็นคำอธิบายการประมวลผลเป็นประโยคโครงสร้างได้ดังนี้

*การสร้างแบบข้อมูล* (Data Modeling) คือ การออกแบบฐานข้อมูลนั่นเอง นักวิเคราะห์ระบบต้องออกแบบว่าจะเก็บข้อมูลอย่างไร และการดึงข้อมูลมาใช้จะใช้วิธีอะไร การจะออก แบบฐานข้อมูลได้นักวิเคราะห์ระบบต้องรู้แน่ชัดแล้วว่า ข้อมูลที่ใช้ทั้งหมดมีอะไรบ้าง ตัวอย่างฐานข้อมูลอาจจะเป็นแบบตารางธรรมดา (Relational Database) และการดึงข้อมูลมาใช้ โดยมีคีย์เป็นตัวใช้ค้นหาเป็นแบบอินเด็กซ์ไฟล์ (Index File) เป็นต้นซึ่งส่วนใหญ่ของฐานข้อมูลในปัจจุบันนี้ใช้แบบ ตารางเพราะว่าง่ายที่จะทำความเข้าใจ

*การสร้างแบบจำลองระบบ* (System Modeling) คือ นำทุกสิ่งทุกอย่างที่ได้จากพจนานุกรมข้อมูล แผนภาพแสดงกระแสข้อมูล ฐานข้อมูล มารวมกันเป็นระบบใหม่ และที่สำคัญก็คือ ความต้องการใหม่ของระบบจะถูกเพิ่มเติมเข้ามาในระบบใหม่นี้ ซึ่งระบบใหม่นี้จะเป็นระบบที่เราต้องการ นอกจากนั้นต้องประมาณว่า จะต้องใช้บุคลากร อุปกรณ์ และพัสดุอะไรบ้าง และใช้เป็นจำนวนเท่าไร

*ข้อมูลเฉพาะของปัญหา* (Problem Specification) ในท้ายที่สุดนักวิเคราะห์ระบบจะรวบรวมสิ่งที่ทำมาทั้งหมดเขียนเป็นรายงานฉบับหนึ่ง เรียกว่า ข้อมูลเฉพาะของปัญหา ซึ่งประกอบด้วยพจนานุกรมข้อมูล แผนภาพแสดงกระแสข้อมูล ข้อมูลเฉพาะการประมวลผล ฐานข้อมูล และแบบระบบใหม่ รายงานนี้จะถูกใช้อ้างอิงตลอดโครงการพัฒนาระบบ ถ้าเอกสารนี้ถูกต้องและละเอียดเพียงพอ การออกแบบในขั้นต่อไปจะง่ายมากแต่ถ้าตรงกันข้ามเอกสารนี้มีรายละเอียดไม่เพียงพอ เชื่อได้เลยว่าระบบที่เสร็จออกมาจะต้องมีปัญหาแน่นอน

*การจัดการโครงการ* (Project Management) เนื่องจากการทำหน้าที่เป็นนักวิเคราะห์ระบบ จะเห็นว่ามีงานที่จะต้องทำมากพอสมควร ดังนั้นการควบคุมการทำงานเพื่อไม่ให้เกิน เวลาที่วางแผนเอาไว้ตลอดโครงการเป็นเรื่องที่จำเป็นมากเพราะว่าถ้าใช้เวลามากเกินกว่าที่วางแผนไว้ ก็หมายความว่าค่าใช้จ่ายจะต้องบานปลายแน่นอนการวางแผน และควบคุม โครงการได้ดีก็โดยการวางแผนตารางเวลาสำหรับงานย่อยๆ ซึ่งเราทราบอยู่แล้วว่ามีอะไรบ้าง เครื่องมือที่ช่วยในการวางแผนและติดตามควบคุมโครงการได้แก่ แกนต์ชาร์ต (Gantt Chart) แผนภาพแกนต์ชาร์ตจะมีรายละเอียดของงานที่จะทำและเวลา

รูปที่ 6.3 ตัวอย่างแกนต์ชาร์ต

*ผังงานระบบ* (System Flowchart) เป็นการใช้แผนภาพที่ใช้แสดงอินพุท เอาต์พุต และการประมวลผล (Process) ของระบบ ในบางกรณีเราใช้ผังงานระบบแทนแผนภาพ แสดงกระแสข้อมูล ในบางกรณีก็ใช้ด้วยกัน ตัวอย่าง ผังงานระบบสำหรับแก้ไขข้อมูลในจานแม่เหล็ก

*โมเดลทางกายภาพและทางตรรกภาพ* (Physical and Logical Model) เมื่อเราพูดถึงลอจิคัลจะหมายถึง การกระทำอย่างใดอย่างหนึ่งที่เราพูดถึงโดยที่ไม่สนใจว่าจะทำอย่างไร เช่น เราพูดว่า เราพูดว่าเรียงลำดับข้อมูล เราจะ ไม่สนใจว่าจะทำการเรียงลำดับข้อมูลทำได้อย่างไร เราเรียกการกระทำแบบนี้ว่าลอจิคัล อีกนัยหนึ่งก็คือ 'ทำอะไร" ในขณะที่ ฟิสิคัลจะมี ความหายตรงกันข้าม คือ จะต้องทราบว่า การจะทำอะไรนั้นจะต้องทำอย่างไรเช่น การเรียงลำดับข้อมูลต้องทราบว่า จะต้องใช้โปรแกรม Utility ช่วยในการเรียงลำดับสรุปก็คือ ลอจิคัลไม่สนใจว่า "จะทำอย่างไร" แต่ฟิสิคัลต้องคำนึงถึงว่า "จะต้องทำอย่างไร"

รูปที่ 6.4 ตัวอย่างแผนภาพลอจิคัลและฟิสิคัลสำหรับพิมพ์รายงาน

**เทคนิคในการรวบรวมข้อมูลเพื่อหาความจริงของระบบ**

ถ้าต้องการออกแบบระบบใหม่จะต้องเข้าใจว่าระบบเดิมเป็นอย่างไร ทำงานอย่างไร มีขั้นตอนการทำงานอย่างไร ปัญหาก็คือจะเก็บข้อมูลอย่างไร สิ่งที่เราทราบในขณะนี้คือ ถ้าต้องการออกแบบระบบใหม่ จะต้องเข้าใจว่าระบบเดิมเป็น อย่างไร ทำงานอย่างไร ปัญหาก็คือจะเก็บข้อมูลอย่างไรจึงจะทำให้เข้าใจระบบเดิม วิธีเท่านั้น นอกจากนั้นจะมีตัวอย่าง การเก็บรวบรวมข้อมูล

นักวิเคราะห์ระบบต้องแสดงความจริงใจต่อผู้ใช้และแสดงให้เขาเห็นว่า เราจะเข้ามาแก้ปัญหาให้ โดยที่การ แก้ปัญหาจะทำด้วยกันและจะช่วยให้งานของเขาง่ายยิ่งขึ้น ความร่วมมือของผู้ใช้ เป็นสิ่งสำคัญมากในการพัฒนาระบบ ศาสตราจารย์ หลุยส์ เดวิด แห่งมหาวิทยาลัยแคลิฟอร์เนีย เชื่อว่า "ไม่มีผู้ใดที่จะออกแบบระบบการทำงานให้ผู้อื่นได้" ดังนั้นบทบาทของผู้เชี่ยวชาญก็คือช่วยให้เขาเหล่านั้นออกแบบระบบของตัวเอง เพราะฉะนั้นสิ่งที่ถูกต้องก็คือ ให้ผู้ใช้มีส่วนในการออกแบบระบบของเขาเองด้วย

***การเตรียมการสัมภาษณ์*** (Preparation for an Interview) ก่อนการสัมภาษณ์ เราต้องเตรียมตัวพอสมควร ทีเดียว จะต้องเตรียมแบบฟอร์มข้อมูลและรายงานให้พร้อมระหว่าง การสัมภาษณ์ ถ้าต้องการอ้างอิงถึงเอกสารใดจะได้มีพร้อมไม่ต้องเสียเวลาค้นหาเอกสาร รายงาน หรือแบบฟอร์มที่จะต้องใช้ นอกจากนั้นควรดูว่า ผู้ใช้ที่จะไปสัมภาษณ์มีตำแหน่ง อะไรในองค์การก่อนการสัมภาษณ์จะต้องขออนุญาตจากผู้บังคับบัญชาและนัดล่วงหน้าด้วย

เราต้องกำหนดจุดประสงค์ของการสัมภาษณ์ เช่น ข้อมูลอะไรที่เราต้องการ อาจจะเตรียมคำถามไว้ล่วงหน้าแล้ว ส่งไปให้ผู้ใช้ก่อนวันสัมภาษณ์ 3-4 วัน เพื่อให้ผู้ใช้เตรียมตัวตอบ คำถามเหล่านั้น

ในบางกรณี อาจจะขอให้ผู้ใช้เก็บสถิติการทำงานของเขาทุกวันเป็นเวลาสัก 1 เดือนเพื่อที่จะได้ข้อมูลที่แม่นยำยิ่งขึ้นและบางครั้งเขาอาจจะไม่ทราบว่าที่จริงแล้วปริมาณการทำงาน ของเขาในแต่ละวันเป็นเท่าไร การเก็บสถิติจะทำให้ตัวเขาเอง ได้รับข้อมูลที่ถูกต้องด้วย

เมื่อเริ่มต้นการสัมภาษณ์ถ้าเป็นไปได้เราควรจะสัมภาษณ์ลูกค้าในที่ทำงานของเขาเพราะว่าเอกสารต่างๆ มีพร้อมการสัมภาษณ์ควรจะสัมภาษณ์ทีละคนแบบตัวต่อตัว ถ้ามีบุคคลที่ สามอาจจะทำให้ผู้ใช้สัมภาษณ์อึดอัดได้ การเริ่มสัมภาษณ์ควรจะทำตัวให้ง่าย ทำบรรยากาศให้ดีขึ้นด้วยการพูดเรื่องทั่วๆ ไป หรือเรื่องเกี่ยวกับผู้ให้สัมภาษณ์โดยดูจากสิ่งต่างๆ ที่อยู่ใน ห้องทำงานเขา อาจจะเริ่มต้นด้วยการถามเกี่ยวกับรางวัลที่เขาได้รับ เรื่องครอบครัว (ถ้ามีรูปติดอยู่ในห้องทำงาน) เรื่องกีฬา เป็นต้น

หลังจากนั้นให้อธิบายถึงจุดประสงค์ของการสัมภาษณ์ นั่นก็คือผู้ให้สัมภาษณ์มีประสบการณ์ความเชี่ยวชาญในธุรกิจที่ทำอยู่ แต่นักวิเคราะห์ระบบไม่มีความรู้ในเรื่องนั้น จะต้องให้ เขารู้ว่าระบบใหม่นั้นก็คือ ระบบของเขาเอง เป็นระบบที่เขาสร้างขึ้น เราเป็นเพียงแต่เป็นผู้ช่วยเท่านั้นและในเวลาเดียวกันเราจะต้องทำให้ผู้ใช้แน่ใจว่า สิ่งที่เขาต้องการเป็น ความลับจะถูกปกปิดเป็นอย่างดีต้องทำให้เขาเชื่อมั่นและเชื่อถือในตัวเรา

ระหว่างการสัมภาษณ์ ตัวคำถาม-คำตอบเป็นหัวใจของการสัมภาษณ์ คำถามที่ใช้อาจจะแตกต่างกันขึ้นอยู่กับหน้าที่และความรับผิดชอบของผู้ให้สัมภาษณ์ (ผู้ใช้) ถ้าเป็นฝ่ายบริหาร ผู้จัดการ เราจะถามเกี่ยวกับนโยบายขององค์กร รายละเอียดวิธีการทำงานจะถามได้จากเสมียน ตัวอย่างเช่น ถามประธานบริษัทว่ามีนโยบายเกี่ยวกับหนี้สินของบริษัท อย่างไร แต่ จะถามเสมียนว่าคำนวณหาหนี้สินของบริษัทได้อย่างไร

ระหว่างการสัมภาษณ์อาจจะมีการจดบันทึกหรือใช้เทปอัดไว้ด้วยก็ได้ แต่ควรจะขออนุญาตผู้ให้สัมภาษณ์เสียก่อน คำถามแรกที่ใช้ควรจะพิถีพิถันหน่อย ควรจะเป็นเรื่องกว้างๆ อย่างเช่น คำถามเกี่ยวกับหน้าที่และความรับผิดชอบ ซึ่งจะเห็นได้จากตัวอย่างคำถามในการสัมภาษณ์ที่กล่าวมาแล้ว คำถามนี้เป็นเรื่องที่ผู้ให้สัมภาษณ์คุ้นเคยอยู่แล้ว เพราะฉะนั้น เขาจะให้ความร่วมมืออย่างดี

เมื่อการสัมภาษณ์ดำเนินไปเรื่อยๆ คำถามที่ใช้ก็ควรจะเฉพาะเจาะจงมากขึ้น ตัวอย่างเช่น คำถามว่า "จากรายงานทั้งหมดที่คุณได้รับ มีข้อมูลใดที่คุณไม่เคยใช้เลย" หรือ "จากรายงานเตรียมเงินสดที่คุณได้รับมีข้อมูลใดที่ไม่เคยใช้เลย" คำถามแรกอาจจะกว้างเกินไป ซึ่งอาจจะทำให้ผู้ให้สัมภาษณ์ต้องเสียเวลารวบรวมจากรายงานทั้งหมด ถึงแม้ว่า การตั้งคำถามที่ "เฉพาะเจาะจง" เราต้องตั้งหลายคำถาม แต่ก็จะได้รับรายละเอียดมากยิ่งขึ้นด้วย

คำถามแบบไหนที่ควรจำใช้นั้นอยู่กับรายละเอียด คำถามที่เหมาะสมควรจะเกี่ยวกับหน้าที่ความรับผิดชอบในปัจจุบัน ข้อมูลที่มีอยู่ วิเคราะห์ปริมาณงาน และการควบคุมระบบ เป็นต้น นอกจากนั้นควรเปิดโอกาสให้ผู้ให้สัมภาษณ์วิจารณ์ระบบในปัจจุบันว่าเป็นอย่างไร และในความคิดเห็นของเขาควรจะปรับปรุงระบบได้อย่างไร ข้อมูลอะไรที่ได้มาแล้ว ไม่เคยใช้ หรือข้อมูลอะไรบ้างที่ต้องการในการทำงาน วิธีการอะไรบ้างที่ควรจะปรับปรุงให้มีประสิทธิภาพมากขึ้น หรือควรจะยกเลิกไปเลย เราควรเก็บข้อมูล สถิติ แบบฟอร์มต่างๆ เพื่อสนับสนุนความคิดของผู้ให้สัมภาษณ์ หรือเพื่อพิสูจน์ว่าความคิดอันนั้นไม่ถูกต้อง นอกจากนั้นเขาอาจจะให้ความเห็นเกี่ยวกับการเมืองในองค์กรเพิ่มมากขึ้นก็ได้

หลังจากคำถามแล้วอย่างลืมที่จะ "ฟังคำตอบ" และแทนที่จะฟังคำตอบอย่างเดียวเราควรจะพยายามดึงส่วนสำคัญของคำตอบเพื่อใช้เป็นคำถามต่อไป คงมีหลายครั้งที่เราได้รับการ แนะนำให้รู้จักคนใดคนหนึ่ง แล้วก็ลืมชื่อของเขา ทั้งนี้เพราะว่ามัวแต่กังวลว่าจะพูดอะไรต่อไปดี

ข้อผิดพลาดอื่นๆ ที่อาจจะเกิดได้ระหว่างการสัมภาษณ์ได้แก่ นักวิเคราะห์ระบบพูดมากกว่าฟัง ถามคำถามนำเช่น "นี่เป็นวิธีการสั่งซื้อที่แย่มาก คุณไม่คิดว่า" หรือระหว่างผู้ให้ สัมภาษณ์กำลังพูดขัดจังหวะเขา หรือถามคำถามที่ให้คำตอบอยู่แล้ว เช่น "ดิฉันคิดว่าคุณคงจะใช้วิธี FIFO ดังนั้น" เราควรจะหยุดรับฟังคำตอบถึงแม้ว่าทราบคำตอบอยู่แล้ว สิ่งที่ นักวิเคราะห์ระบบจะต้องเรียนรู้ก็คือ "เมื่อไรที่ควรจะเปิดใจให้กว้าง และเมื่อไรที่ควรจะปิดปาก"

ในกรณีที่ผู้ให้สัมภาษณ์ไม่ให้ความร่วมมือ เราควรยุติการสัมภาษณ์ ถ้าพยายามจะดำเนินการสัมภาษณ์ต่อไป ก็รังจะทำให้เสียเวลาโดยเปล่าประโยชน์ และสิ่งที่ไม่ควรทำก็คือบอก ผู้บังคับบัญชาเขาเกี่ยวกับเรื่องนี้ เพราะสิ่งที่ต้องการ ก็คือ ความร่วมมือ ซึ่งจะทำให้เราได้ข้อมูลที่ถูกต้อง เราควรพยายามนัดครั้งที่สองเพราะว่าเขาอาจจะมีอารมณ์ไม่ดีในวันนั้นถ้า วันที่สองยังเหมือนเดิมก็ควรจะเปลี่ยนแหล่งข้อมูลใหม่ได้แล้ว

สรุปการให้สัมภาษณ์ เมื่อจบการสัมภาษณ์ นักวิเคราะห์ระบบควรจะสรุปปากเปล่าข้อมูลทั้งหมด รวมทั้งประเด็นสำคัญต่างๆ เพื่อให้แน่ใจว่าไม่มี่การเข้าใจผิดเกิดขึ้น และบอกผู้ ให้สัมภาษณ์ทราบว่าจะส่งรายงานสรุปการสัมภาษณ์มาให้ภายหลัง เพื่อตรวจสอบให้แน่ใจอีกครั้งอีกครั้งหนึ่ง และท้ายที่สุดอย่าลืมขอบคุณผู้ใช้สัมภาษณ์ที่ได้สละเวลาอันมีค่าของเขา ในการสัมภาษณ์ครั้งนี้

หลังการให้สัมภาษณ์เมื่อเสร็จสิ้นต้องถอดคำพูดคำต่อคำ เราควรส่งสำเนาสรุปการสัมภาษณ์พร้อมด้วยจดหมายขอบคุณไปให้ผู้ที่เราสัมภาษณ์เพื่อให้เขาตรวจสอบอีกครั้งหนึ่ง ถ้า หากยังมีสิ่งที่ขาดตกบกพร่องหรือข้อมูลไม่เพียงพอ เราทราบได้ทันทีจากสรุปรายงานนี้ ซึ่งอาจจะส่งใบนัดเพื่อขอสัมภาษณ์อีกครั้งหนึ่งก็ได้

ขั้นสุดท้ายของการสัมภาษณ์คือ วิเคราะห์ข้อมูลที่เราได้มาทั้งหมด แล้ววิเคราะห์ว่าข้อมูลนั้นถูกต้องมากน้อย เพียงใด มีข้อมูลที่ลำเอียงหรือไม่ ซึ่งปกติแล้วกิจการธุรกิจทุกๆ ธุรกิจ มักจะมี "สิ่งเคลือบแฝง" อยู่ ผู้ให้สัมภาษณ์บางคนอาจจะให้ข้อมูลที่ไม่ถูกต้อง ดังนั้นนักวิเคราะห์ระบบที่ไม่มีประสบการณ์อาจจะถูกหลอกได้ สิ่งที่นักวิเคราะห์ระบบที่ดีควรจะทำก็คือ ดึงข้อมูลที่ถูกต้องออกจากการสัมภาษณ์นั้นๆ หรือจากแหล่งอื่นๆ ที่เป็นไปได้ การเปรียบเทียบข้อมูลที่ถูกต้อง หลายๆ แหล่งจะทำให้ได้ข้อมูลที่ถูกต้องมากขึ้น ผลการวิเคราะห์ควร จะเก็บเป็นความลับ เพราะว่าคงไม่มีลูกค้าคนไหนพอใจที่จะถูกเปิดเผยบางสิ่งบางอย่างออกมา

หลักการสัมภาษณ์ทั้งหมดนี้สามารถใช้กับทุกๆ คนในโครงการที่เกี่ยวข้องเหมือนกันหมดไม่ว่าจะเป็นผู้จัดการ หรือพนักงานหยิบของในคลัง ทุกคนควรจะได้รับการปฏิบัติ เหมือนกันหมด

**สรุปผลลัพธ์จากการรวบรวมข้อมูล**
1. สำเนาของรายงานและแบบฟอร์ม อินพุททั้งหมดที่เกี่ยวข้อง
2. ศึกษาเอกสารของระบบทั้งหมด วิธีการทำงาน โปรแกรม ไฟล์ และการเชื่อมโยงของไฟล์
3. สังเกตดูการทำงานจริงของระบบ เพื่อทราบขั้นตอนการทำงานที่แท้จริง

**แผนภาพกระแสข้อมูล**
แผนภาพกระแสข้อมูล (DFD) เป็นเครื่องมือที่ใช้กันอย่างแพร่หลายในการเขียนแบบระบบใหม่ โดยเฉพาะกับระบบที่ "หน้าที่" ของระบบมีความสำคัญและมีความสลับซับซ้อนมากกว่าข้อมูลที่ไหลเข้า

รูปที่ 6.5 DRD ต่าง

**ส่วนประกอบของ DFD**DFD มีองค์ประกอบ 4 อย่าง ซึ่งใช้สัญลักษณ์ต่าง ๆ แทนดังต่อไปนี้
1. สัญลักษณ์แทนการประมวลผล (Process) เป็นวงกลม

2. สัญลักษณ์แทนกระแสข้อมูลเป็นลูกศร

3. สัญลักษณ์แทนแหล่งเก็บข้อมูลเป็นเส้นขนาน 2 เส้น โดยมีชื่อกำกับ

4. สี่เหลี่ยมผืนผ้าเป็นสัญลักษณ์แทนสิ่งที่อยู่นอกระบบ

การประมวลผลโพรเซส (Process) การประมวลผลโพรเซส (Process) คือ งานที่จะต้องทำแทนด้วยวงกลมและมีขื่ออยู่ภายในวงกลม 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 เข่น เลขที่สินค้า หรือกลุ่มของข้อมูล เช่น ข้อมูล พนักงาน ข้อมูลลูกค้า เป็นต้น กลุ่มของข้อมูลควรจะเป็นเรื่องเดียวกัน หรือสัมพันธ์กัน ตัวอย่างเช่น ข้อมูลลูกค้าอาจจะมีรายละเอียดเป็นชื่อลูกค้า เลขที่ ที่อยู่ แต่ไม่ควรรวมจำนวน สินค้าในคลังอยู่ในข้อมูลเดียวกัน ถ้าต้องการอ้างอิงข้อมูลทั้งสองที่ไม่เกี่ยวข้องกันให้เขียนแยกเป็นลูกศร 2 อัน

รูปที่ 6.6 ข้อมูล 2 อันไม่เหมือนกันจะต้องแยกลูกศรออก

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

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

รูปที่ 6.7 การแก้ไขข้อมูล

*สิ่งที่อยู่นอกระบบ* (Terminator) สิ่งที่อยู่นอกระบบแทนด้วยสี่เหลี่ยมผืนผ้า ซึ่งมีชื่อกำกับอยู่ด้วย ส่วนใหญ่จะเป็นตัวบุคคล หรือองค์กรต่างๆ สิ่งที่อยู่นอกระบบอาจจะเป็นที่ส่งข้อมูล เข้าสู่ระบบ หรืออาจจะเป็นที่รับข้อมูลจากระบบก็ได้ เราไม่สนใจการทำงานภายในของสิ่งที่อยู่นอกระบบ ถึงแม้ว่าจะมีการติดต่อผ่านทางข้อมูล เราจะสนใจเฉพาะข้อมูลที่เข้าสู่ระบบ หรือออกจากระบบสู่ภายนอกเท่านั้น
DFD ระดับสูงสุด (Context Level Data Flow Diagram) เพื่อให้เข้าใจการเขียน DFD ได้ดี เราทดลองเขียนแผนภาพนี้กับระบบบัญชีเจ้าหนี้ จากรูปเป็น DFD ระดับสูงสุดของบัญชี เจ้าหนี้ซึ่งมีชื่อเรียกอีกอย่างหนึ่งว่า "Context Diagram" ซึ่งระดับนี้จะบอกว่าระบบที่เราสนใจมีอินพุทเป็นอะไรผลลัพธ์ที่ได้จากระบบ และข้อมูลที่เกี่ยวข้องกับระบบ ภายนอก

แผนภาพระดับสูงสุดของ DFD แสดงถึงขอบเขตของระบบ ข้อมูล และผลลัพธ์ของระบบต่อไปจะกล่าวถึงวิธีการสร้าง DFD
สิ่งที่อยู่ภายนอกระบบกับ DFD ระดับสูงสุด DFD ระดับสูงสุดเท่านั้นที่จะแสดงส่วนที่อยู่ภายนอกระบบ ส่วนนี้มีความสำคัญเพราะว่าเป็นส่วนที่บอกว่าระบบนั้นๆ ได้รับข้อมูล มาจากที่ใด และผลลัพธ์ต่างๆ ถูกส่งไปที่ใดบ้าง DFD ในระดับลึกลงไปจะไม่แสดงสิ่งที่อยู่นอกระบบคือ ไม่มีสิ่งนี้เป็นส่วนประกอบ
การเก็บข้อมูลกับ DFD ระดับสูงสุด จากรูป DFD ระดับสูงสุดจะไม่มีสัญลักษณ์การเก็บข้อมูลปรากฏอยู่ แต่ ไม่ได้หมายความว่าระบบนี้ไม่มีการเก็บข้อมูล แต่การเก็บข้อมูลจะ อยู่ภายในระบบ หรือภายในโพรเซส ดังนั้นจึงไม่ปรากฏอยู่ใน DFD ระดับนี้ การเก็บข้อมูลจะปรากฏอยู่ใน DFD ระดับลึกลงไปหรือในชั้นลูกหลานต่อๆ ไป ดังแสดงดังรูป

รูปที่ 6.9 DFD ระดับสูงสุดและระดับ

โพซสกับ DFD ระดับสูงสุด ในแผนภาพ DFD ระดับแม่จะมีเพียงโพรเซสเดียวคือ การจ่ายใบทวงหนี้ DFD ระดับนี้เป็นการตั้งชื่อการทำงานรวมของทั้งระบบ โดยปกติขื่อ โพรเซสในระดับนี้อาจจะตั้งเป็นคำนามก็ได้ซึ่งต่างจาก กฏเกณฑ์ที่กล่าวมาแล้วว่าโพรเซสควรจะตั้งชื่อเป็นคำกริยา การตั้งขื่อโพรเซสเป็นคำนามมีอยู่เฉพาะใน DFD ระดับแม่ เท่านั้น
กระแสข้อมูลกับ DFD ระดับแม่ กระแสข้อมูลใน DFD ระดับแม่ จะแสดงอินพุทจากระบบภายนอกสู่ระบบของเราและผลิตผลลัพธ์ออกจากระบบของเราออกสู่ภายนอก ตัวอย่างเช่น ใบทวงหนี้จากผู้ขายสู่ระบบการจ่ายเงินใบทวงหนี้ จากระบบเราก็จ่ายเงินให้กับผู้ขาย ซึ่งจะมีข้อมูลเลขที่บัญชีจากฝ่ายบัญชีเข้าสู่ระบบและระบบจะผลิตรายงานเพื่อการ ตรวจสอบ และ ใบสำคัญแสดงบัญชีเจ้าหนี้ ในที่นี้ลูกศรหนึ่งอันใช้แทนข้อมูล 2 อย่าง ซึ่งเราไม่อนุญาตในการเขียน DFD ระดับลูก แต่ในที่นี้เราอนุญาต เพราะว่าการใช้ลูกศรหนึ่งอันแทนข้อมูล หนึ่งอย่างจะทำให้ DFD ดูรกเกินไป ในทำนอง เดียวกันกับข้อมูลสู่ฝ่ายบริหารจะมีรายงานเพื่อเตรียมเงินสดกับรายงานเพื่อการตรวจสอบบัญชีเจ้าหนี้อยู่บนลูกศร อันเดียวกันและมี รายงานเพื่อการตรวจสอบบัญชีวิ่งออกจากระบบเหมือนกัน 2 แห่งคือ เข้าสู่ฝ่ายบัญชี และฝ่ายบริหาร ในที่นี้รายงานนี้ควรจะเป็นฉบับสำเนา (Copy) ซึ่งมีรายละเอียดเหมือนกันนั่นเอง

ภาพรวมของ DFD ระดับลูกตามมาอันแรกจะเป็นภาพการทำงานของระบบทั้งหมดเรียกว่า "ภาพรวมของ DFD (Overview DFD)"

DFD ระดับแม่ไม่เพียงพอทีจะบอกรายละเอียดการทำงานทั้งหมดของระบบได้ ดังนั้นจึงจำเป็นที่ต้องแตกเป็น ลูกหลานต่อไป ซึ่งจะทำให้ได้ DFD ต่างระดับขึ้น DFD ระดับลูกอันแรกติดกับระดับแม่เป็นภาพรวมของการทำงานดังรูปความสัมพันธ์ในที่นี้จะเป็นแบบ "แม่/ลูก" และ DFD ในระดับลูกนี้ ก็อาจจะเป็นแม่ของลูกในระดับถัดไป และสามารถ แตกแยกย่อยลงไปได้เรื่อยๆ ภาพรวมของ DFD นี้จะไม่มีสัญลักษณ์ของระบบภายนอกเข้ามาเกี่ยวข้อง เพราะว่าระบบ ภายนอกปรากฏอยู่เฉพาะใน DFD ระดับสูงสุดอย่างเดียวเท่านั้น

การเก็บข้อมูลกับ DFD ระดับลูก การเก็บข้อมูลอันแรกในภาพนี้คือ ไฟล์ใบสั่งซื้อ ทำไมจึงจำเป็นต้องมีการเก็บข้อมูลอันนี้ทั้งนี้เพราะว่าข้อมูลนี้จะต้องเก็บไว้ระยะเวลาหนึ่งซึ่ง จะต้องนำมาใช้ในภายหลัง เริ่มตั้งแต่มีการสั่งซื้อจนกระทั่งเสร็จสิ้นการจ่ายเงิน นอกจากนั้นในกรณีที่ข้อมูลถูกป้อนเข้ามาแบบหนึ่ง แต่เราต้องเรียกใช้อีกแบบหนึ่ง ลักษณะนี้จำ เป็นต้องมีการเก็บข้อมูลเหมือนกัน เช่น ข้อมูลผู้ขาย เป็นต้น

โดยปกติแล้วเราจะแบ่งโพรเซสตามหน้าที่ของ DFD ที่เห็นในรูปได้จากการเขียนแล้วหลายครั้ง ในที่สุดก็ได้รูปนี้ ขึ้นมา การทดลองเขียนครั้งแรกนั้นรวมโพรเซสที่ 3 และ 4 เข้าด้วยกันและส่วนหนึ่งของโพรเซสที่ 3 ในการตรวจสอบการอนุมัติไปรวมอยู่ในโพรเซสที่ 2 ในกรณีนี้เราจะสามารถใช้ข้อมูลที่เหมือนกันหลายอย่างได้ในโพรเซสที่ 4 แต่การ ทำงานในโพรเซสนี้จะมากมายทีเดียว ดังนั้น DFD ในภาพข้างบนจึงดูดีกว่า

การเขียน DFD รูปหนึ่งๆ ไม่ควรมีจำนวนโพรเซสมากกวา 7 เลข 7 เป็นเลขที่สวยสำหรับสมองคน เช่น เบอร์โทรศัพท์เป็นต้น และเหมาะที่จะเขียนบนกระดาษขนาด A4 (8.5\*11 นิ้ว) แต่ถ้าจำเป็นมากที่จะต้องมีมากกว่า 7 โพรเซสก็ ไม่เป็นไร เราไม่ควรเขียนโพรเซสเดียวกันใน DFD รูปเดียวกัน DFD ในระดับนี้จะไม่รวมโพรเซสจำพวก "Edit/แก้ไข" หรือ "Format/จัดรูปแบบ" การทำงานทำนองนี้จะเขียนอยู่ใน DFD ระดับลึกกว่านี้อีก จากตัวอย่างจะเห็นว่าทุกโพรเซสตั้งชื่อโดยใช้คำกริยาเป็นคำเริ่มต้น ซึ่งเราปฏิบัติตามกฎทุกประการ

กระแสข้อมูลกับภาพรวมของ DFD ก่อนที่จะพูดถึงข้อมูลที่เคลื่อนไหวใน DFD เราลองมาพิจารณาพจนานุกรมข้อมูลคือ

ใบอนุมัติจ่ายเงิน = เลขที่ใบสั่งซื้อ

รายละเอียดการจ่ายเงิน = เลขที่เช็ค + เลขที่ใบทวงหนี้ + เลขทีใบสั่งซื้อ

ใบทวงหนี้ค้างจ่าย = {เรคอร์ดใบทวงหนี้ค้างจ่าย}

เรคอร์ดใบทวงหนี้ค้างจ่าย =เลขที่ใบทวงหนี้ + เลขที่ใบสั่งซื้อ + จำนวนเงินในใบทวงหนี้ + เลขที่ผู้ขาย + ส่วนลด + กำหนดวันชำระเงิน

ชื่อของข้อมูลจะต้องไม่เหมือนกัน ลูกศรของข้อมูลมีเพียงข้อมูลเดียวเท่านั้นกำกับอยู่หัวลูกศรบอกทิศทางการเคลื่อนที่ ถ้ามีการเคลื่อนที่ของข้อมูลออกจากไฟล์ หรือเข้าสู่ไฟล์โดย ที่ข้อมูลทั้งหมดหรือข้อมูลส่วนใหญ่ของเรคอร์ดถูกใช้ ก็ไม่จำเป็นจะต้องมีขื่อข้อมูลกำกับลูกศรหมายความว่าข้อมูลที่เคลื่อนที่นั้นคือ ข้อมูลในไฟล์นั่นเอง ถ้าใช้เพียงบางส่วนของ เรคอร์ดเท่านั้นก็ให้ใส่ชื่อข้อมูลกำกับไว้ด้วย ตัวอย่างในรูป DFD ลูกศรจากไฟล์สั่งซื้อและไฟล์ใบทวงหนี้ค้างจ่ายไม่มีชื่อ ข้อมูลกำกับแสดงว่าข้อมูลทั้งเรคอร์ดจะต้องถูกใช้ แต่ข้อจากไฟล์ผู้ขายอันหนึ่งมีชื่อกำกับว่า "ชื่อผู้ขาย" เป็นการแสดงว่าฟิลด์ชื่อผู้ขายเท่านั้นที่ต้องการ

**โพรเซสกับภาพรวมของ DFD**
ภาพรวมของ DFD โดยทั่วไปมักจะมีโพรเซสทั้งหมดด้วยกัน 5 โพรเซส โดยมีเลขที่กำกับด้วย แต่ละโพรเซสทำงานของตัวเองแยกกัน ปัญหาของการเขียนโพรเซสคือ ทำอย่างไร จึงจะ "แบ่ง" งานออกจากกันได้ ในตัวอย่างอาจจะลบโพรเซสที่ 3 ออกแล้วรวมเอาไว้ในโพรเซสที่ 4 ก็ได้ หรือจะดึงงานบางส่วนในโพรเซสที่ 1 ไปรวมกับโพรเซสที่ 2 ก็ได้อีก เช่นกัน การแบ่งจำนวนงานนั้นไม่มีคำตอบว่า "ถูกหรือผิด" ที่แน่นอนตายตัว แต่คำตอบหนึ่งอาจจะดีกว่าคำตอบหนึ่งก็ได้ เราอาจจะแบ่งการทำงานใหม่ซึ่งจะทำให้ระบบนั้นดีขึ้นหรือเลวลง

การแบ่งจำนวนโพเซสใน DFD ไม่มีกฎเกณฑ์ตายตัว การแบ่งจำนวนนี้ขึ้นอยู่กับ "ความชำนาญหลังจากที่มี ประสบการณ์มาพอสมควร" ถ้าเปรียบเทียบการเขียนโปรแกรมก็ เหมือนกับการแยกเขียนโปรแกรมย่อยนั่นเอง ซึ่งจะต้องอาศัยประสบการณ์ในการเขียนโปรแกรมมาช่วยมากทีเดียว ปัญหาการแบ่งงานก็คือ ขอบเขตของงานนั่นเอง

ถึงจุดนี้จะขอแนะนำหลักการใหม่อีกอันหนึ่งที่ควรทราบนั่นก็คือ ความสมดุล DFD ในระดับแม่จะต้องสมดุลกับ DFD ในระดับลูก ซึ่งหมายความว่าข้อมูลขาเข้ากับผลลัพธ์ในระดับ ลูกจะต้องเหมือนกันในระดับแม่ เช่น ใบทวงหนี้ซึ่งเป็นอินพุทใน DFD ระดับสูงสุด จะต้องเป็นอินพุทในโพรเซสใดโพรเซสหนึ่งใน DFD ระดับลูกคือ ระดับภาพรวมการทำงาน ความสมดุลนี้ไม่รวมข้อมูลที่วิ่งอยู่ในภายใน DFD หนึ่งๆ จะเป็นข้อมูลภายในของแผนภาพนั้นๆ ไม่เกี่ยวกับ DFD ในระดับแม่

เราลองมาพิจารณา DFD ระดับแม่และระดับลูกที่กล่าวมาแล้ว จะเห็นว่าใบทวงหนี้รายงานอนุมัติการจ่ายเงิน เลขที่บัญชีแยกประเภท รายงานเพื่อเตรียมเงินสด การจ่ายเงิน และใบสำคัญบัญชีเจ้าหนี้ทั้งหมดนี้สมดุลกันหมด ยกเว้นรายงานตรวจสอบบัญชีเจ้าหนี้ซึ่งปรากฏในระดับแม่สองครั้ง แต่ในระดับลูกปรากฏเพียงครั้งเดียว ทั้งนี้เป็นเพราะว่า ในระดับแม่นั้นรายงานที่เห็นเป็น 2 ฉบับนั้นเป็นสำเนาของรายงานฉบับเดียวกัน ซึ่งไม่จำเป็นจะต้องสมดุลกันระหว่างระดับแม่และลูก

คราวนี้มาลองสังเกตดูทิศทางกระแสข้อมูลระหว่าง DFD ทั้งสองระดับ จะเห็นว่าทิศทางของลูกศรจะไม่ตรงกันคือวิ่งไปกันคนละทิศทาง แต่ถึงอย่างไรก็ตามความหมายของกระแส ข้อมูลยังคงถูกต้องทุกประการ การเขียนทิศทางกระแสของข้อมูลมีกฎว่า เขียนอย่างตรงไปตรงมา ตราบใดที่กระแสข้อมูลยังถูกต้อง แต่ถ้าจะให้ดีก็คือพยายามให้อินพุทไหลเช้ามาทาง ซ้ายมือด้านบนและผลลัพธ์ไหลออกมาทางมุมล่างขวาสุดของหน้ากระดาษ และให้เขียนโยงไปให้ถึงริมกระดาษ เพื่อจะได้สังเกตความสมดุลได้ง่าย สำหรับข้อมูลที่วิ่งระหว่างโพรเซส หรือวิ่งเข้าออกจากไฟล์ที่สร้างขึ้นมาใหม่ DFD ระดับล่างนั้น จำเป็น เพราะว่าเป็นอินพุทของโพรเซสต่างๆ ตัวอย่างเช่น มีข้อมูล 3 ข้อมูล วิ่งเข้าสู่โพรเซส 4 ใน DFD ระดับลูก และ โพเซสนี้จะสร้างผลลัพธ์เป็นรายงานการตรวจสอบบัญชีเจ้าหนี้ และใบสำคัญบัญชีเจ้าหนี้ ข้อมูลอันดับหนึ่งที่เข้าสู่โพรเซสที่ 4 คือ เลขที่เช็ค ซึ่งเป็นข้อมูลอันหนึ่งในรายงานการตรวจ สอบบัญชีเจ้าหนี้ด้วย สำหรับเลขบัญชีแยกประเภท จะใช้เป็นข้อมูลสำหรับการเตรียมรายงานใบสำคัญบัญชีเจ้าหนี้ จะเห็นได้ว่าโพรเซสที่ 4 นี้มีข้อมูลที่จำเป็นทั้งหมดในการผลิตรายงาน ทั้งหมดของโพรเซสนี้

โพรเซสต่างๆ จะต้องเตรียมข้อมูลที่จำเป็นในการผลิตผลลัพธ์ รายงาน หรือข้อมูลสำหรับโพรเซสอื่นๆ พูดง่ายๆ ก็คือ จะต้องมีข้อมูลที่เพียงพอจึงจะผลิตผลลัพธ์ที่ต้องการได้

ในรูป DFD ระดับลูกนั้นข้อมูลรายงานการจ่ายเงินจะวิ่งมาตัดกับข้อมูลที่วิ่งขนานกัน 2 เส้นเข้าสู่โพรเซสที่ 3 เส้นขนาน 2 เส้นที่เขียนแบบนี้ ( ) หมายถึงข้อมูลชุดเดียวกัน ที่มาจากแหล่งข้อมูลที่ต่างกัน รายงานการจ่ายเงินในที่นี้คือ รายงานฉบับที่สร้างขึ้นในโพรเซสที่ 2 และที่มาจากการอนุมัติจากฝ่ายบริหารนั่นเอง ข้อมูลที่มีลักษณะแตกต่างออกไป อีกแบบหนึ่งก็คือ ข้อมูลอันหนึ่งกระจายเข้าสู่หลายโพรเซสดังในรูปข้างล่างนี้ เราควรจะพยายามใช้หลักการเขียนแบบนี้เพื่อให้ DFD ง่ายขึ้น

รูปที่ 6.12 ข้อมูลหนึ่งแหล่งที่กระจายสู่โปรเซสเซอร์

การทำความเข้าใจ DFD ในระดับแม่และลูกมีความสำคัญมากก่อนที่เราจะแตกลูกหลานต่อไปอีก ซึ่งก็จะให้ หลักการอันเดียวกันกับที่กล่าวมาแล้วในตัวอย่างทั้งหมด ถ้าเข้าใจ การแยกย่อยในระดับแม่ลงมาหาลูก และการทำความ สมดุลกัน ก็จะสามารถเขียน DFD อันอื่นๆ ต่อไปได้ง่าย คราวนี้เราลองมาดู DFD ในระดับหลาน โดยมี DFD ภาพรวมเป็นแม่

**วิธีการสร้าง DFD**
ทั้งหมดนี้เป็นขั้นตอนในการสร้าง DFD ที่มีระบบมากขึ้น
1. กำหนดสิ่งที่อยู่ภายนอกระบบทั้งหมด และหาว่าข้อมูลอะไรบ้างที่เข้าสู่ระบบหรือออกจากระบบที่เราสนใจสู่ระบบที่อย่าภายนอก ขั้นตอนนี้สำคัญมากทั้งนี้เพราะจะทำให้ทราบ ว่าขอบเขตของระบบนั้นมีอะไรบ้าง
2. ใช้ข้อมูลที่ได้จากขั้นตอนที่ 1 นำมาสร้าง DFD ต่างระดับ
3. ขั้นตอนถัดมาอีก 4 ขั้นตอนโดยให้ทำทั้ง4 ขั้นตอนนี้ซ้ำๆ หลายๆ ครั้ง จนกระทั่งได้ DFD ระดับต่ำสุด
3.1 เขียน DFD ฉบับแรก กำหนดโพรเซสและข้อมูลที่ไหลออกจากโพรเซส
3.2 เขียน DFD อื่นๆ ที่เป็นไปได้จนกระทั่งได้ DFD ที่ถูกที่สุด ถ้ามีส่วนหนึ่งส่วนใด ที่รู้สึกว่าไม่ง่ายนักก็ให้พยายามเขียนใหม่อีกครั้งหนึ่ง แต่ไม่ควรเสียเวลาเขียน จนกระทั่งได้ DFD ที่สมบรูณ์แบบ เลือก DFD ที่เห็นว่าดีที่สุดในสายตาของเรา
3.3 พยายามหาว่ามีข้อผิดพลาดอะไรหรือไม่ ซึ่งมีรายละเอียดในหัวข้อ "ข้อผิดพลาดใน DFD"
3.4 เขียนแผนภาพแต่ละภาพอย่างดี ซึ่ง DFD ฉบับนี้จะใช้ต่อไปในการออกแบบ และใช้ด้วยกันกับบุคคล อื่นๆ ที่เกี่ยวข้องในโครงการด้วย
4. นำแผนภาพทั้งหมดที่เขียนแล้วมาเรียงลำดับ ทำสำเนา และพร้อมที่จะนำไปตรวจสอบข้อผิดพลาดจากผู้ร่วมทีมงาน ถ้ามีแผนภาพใดที่มีจุดอ่อนให้กลับไปเริ่มต้นที่ขั้นตอนที่ 3 อีกครั้งหนึ่ง
5. นำ DFD ที่ได้ไปตรวจสอบข้อผิดพลาดกับผู้ใช้ระบบเพื่อหาว่ามีแผนภาพใดไม่ถูกต้องหรือไม่
6. ผลิตแผนภาพฉบับสุดท้ายทั้งหมด

จะเห็นว่าการเขียน DFD นั้นต้องมีการเปลี่ยนแปลงแก้ไขอยู่เกือบตลอดเวลา การเขียน DFD ด้วยมืออาจจะไม่สะดวกนัก ดังนั้นการเขียนด้วยคอมพิวเตอร์จะง่ายในการแก้ไข ซึ่งมีโปรแกรมสำเร็จรูปหลายโปรแกรมใช้ในการเขียน DFD ที่ใช้กันมากอย่างแพร่หลาย ระหว่างเขียน DFD ต้องสร้างพจนานุกรมข้อมูล และเขียนรายละเอียดข้อมูลเฉพาะ ของโพรเซส ขั้นตอนการเขียนทั้งหมดนี้จะช่วยให้เราหาข้อผิดพลาดต่างๆ ที่อาจเกิดขึ้นได้ด้วย

ข้อผิดพลาดใน DFD

การเขียน DFD อาจะเขียนได้หลายแบบ ผลลัพธ์ฉบับสุดท้ายอาจจะไม่เหมือนกันถ้าเขียนโดยนักวิเคราะห์ระบบคนละคน ถึงอย่างไรแนวทางการเขียน DFD ซึ่งจะช่วยให้ เราเขียน DFD ได้ถูกต้องมากขึ้นก็มีอยู่บ้าง ซึ่งพอสรุปได้ดังนี้
1. ถ้า DFD ซับซ้อนมาก ทุกๆ นิ้วในกระดาษถูกใช้งานทั้งหมด แสดงว่า DFD นั้นควรจะแตกย่อยไปอีกระดับหนึ่งหรือมากกว่าหนึ่ง
2. ข้อมูลที่ออกจากโพรเซส หรือผลลัพธ์มีข้อมูลขาเข้าไม่เพียงพอ เราจะต้องพิจารณาแผนภาพต่อไปอีก แต่ที่สำคัญไม่ควรใส่ข้อมูลที่ไม่เคยใช้เข้ามาในโพรเซสเป็นอันขาด
3. การตั้งชื่อโพรเซสนั้นไม่ง่ายนัก อาจจะมีปัญหา 2 อย่างคือ โพรเซสนั้นควรจะแยกออกเป็น 2 ส่วน หรือเรา ไม่ทราบว่ามีอะไรเกิดขึ้นบ้างในโพรเซสนั้นๆ ในกรณีนี้เรา ต้องศึกษาระบบให้ละเอียดยิ่งขึ้น
4. จำนวนระดับในแต่ละแผนภาพแตกต่างกันมาก เช่นโพรเซสที่ 1 มีลูก 2 ชั้น แต่โพรเซสที่ 2 มีลูก 10 ชั้นแสดงว่าการแบ่งจำนวนโพรเซสไม่ดีนัก จำนวนลูกของโพรเซส ไม่จำเป็นต้องเท่ากัน แต่ไม่ควรจะแตกต่างกันมากนัก
5. มีการแตกแยกย่อยข้อมูล รวมตัวของข้อมูล หรือมีการตัดสินใจในโพรเซส แสดงว่าโพรเซสนั้นไม่ถูกต้องการแยกข้อมูล หรือรวมตัวของข้อมูลเป็นหน้าที่ของพจนานุกรม ข้อมูล การตัดสินใจเป็นรายละเอียดอยู่ใน คำอธิบายโพรเซส

การสร้าง DFD ที่ดีเป็นงานที่ยากที่สุดสำหรับนักวิเคราะห์ระบบมือใหม่ หรือแม้แต่ผู้ที่มีประสบการณ์มาแล้วก็ตาม DFD ที่ไม่ดีจะทำให้ผลลัพธ์สุดท้ายของระบบออกมาไม่ดีเช่นเดียว กันทั้งนี้เนื่องจาก DFD เป็นรากฐานสำหรับการออกแบบและพัฒนาโปรแกรม

**คำอธิบายการประมวลผล**

เราสามารถกำหนอข้อมูลของระบบไว้ในพจนานุกรมข้อมูล และแบ่งการทำงานเป็นหน้าที่ต่างๆ ย่อยลงได้ด้วย แผนภาพกระแสข้อมูล แต่เมื่อแตกลูกหลานของโพรเซส ถึงระดับต่ำที่สุดก็ยังไม่ทราบแน่ชัดว่าภายในโพรเซสแต่ละอย่างนั้น ทำงานจริงๆ อย่างไรคำอธิบายการประมวลผลหรือคำอธิบายการทำงานของโพรเซสจึงเข้ามามีบทบาทตอนนี้

คำอธิบายการประมวลผล "Process Description" หรือเรียกอีกอย่างหนึ่งว่า "Minispecs" จะอธิบาย รายละเอียดการทำงานภายในโพรเซสหนึ่งๆ หรืออีกนัยหนึ่งก็คือ โพรเซสนี้เปลี่ยนอินพุตเป็นเอาต์พุตอย่างไร โพรเซสระดับล่างสุดใน DFD เราจะต้องเขียนคำอธิบายว่ามันทำงานอย่างไร

วิธีการที่ใช้อธิบายการประมวลผลที่จะกล่าวในที่นี้มีอยู่ด้วยกัน 2 วิธี คือ
1. ประโยคโครงสร้าง (Structure Sentences)
2. การตัดสินใจแบบตาราง (Decision Tables)
เราจะเลือกใช้วิธีการอันใดอันหนึ่งหรือใช้ปนกันก็ได้ขึ้นอยู่กับความเหมะสม แต่ไม่ว่าจะเขียนด้วยวิธีใดๆ เมื่อเขียนแล้วควรจะมีคุณสมบัติดังนี้

- เขียนแล้วคำอธิบายนั้นสามารถนำมาตรวจสอบความถูกต้องกับผู้ใช้ได้ง่าย การเขียนเป็นประโยคโครงสร้างอาจจะไม่เหมาะสมถ้าต้องนำมาตรวจสอบกับผู้ใช้เพราะว่า คำอธิบายนั้นจะยาวและ คำอธิบายเกี่ยวกับเงื่อนไข หรือการทำงานซ้ำก็เขียนไม่สะดวก ตัวอย่างเช่น เงื่อนไขที่มี AND, OR หรือ NOT เป็นต้น
- เขียนแล้วคำอธิบายนั้นควรจะใช้สื่อสารกับผู้อื่นที่เกี่ยวข้องในระบบได้ง่ายผู้อื่นที่เกี่ยวข้องอาจจะเป็นผู้ใช้ ผู้จัดการ ผู้ตรวจสอบ เป็นต้น การเขียนคำอธิบายเป็นประโยค โครงสร้าง หรือเขียนเป็นการ ตัดสินใจแบบตารางน่าจะเหมาะสมกับบุคคลเหล่านั้นเพราะว่าทั้ง 2 วิธีนั้นง่ายต่อการทำความเช้าใจ ถึงแม้ว่าอาจจะยาวไปหน่อยก็ตาม

โดยทั่วไปแล้ววิธีการเขียนแบบประโยคโครงสร้างเป็นที่นิยมใช้กันมากที่สุด และในโครงการเดียวกันควรจะเลือกใช้วิธีเดียวกันเพื่อให้ง่ายต่อการสื่อสาร การจะเลือกใช้ วิธีการมากกว่าหนึ่งวิธีก็อาจจะเป็นไปได้ทั้งนี้ขึ้นอยู่กับ 1. ความชอบของผู้ใช้ 2. ความชอบของผู้เขียน (นักวิเคราะห์ระบบ) และ 3 ลักษณะการทำงานของโพรเซส

**ประโยคโครงสร้าง**
วิธีนี้ใช้การอธิบายเป็นประโยคเขียนให้มีลักษณะเป็นโครงสร้าง คล้ายๆ การเขียนโปรแกรมโครงสร้างดังตัวอย่างข้างต้น การเขียนประโยคโครงสร้างเราใช้คำศัพท์ต่างๆ กัน ซึ่งอาจจะเลือกใช้คำต่างๆ กันได้ดังนี้

- ใช้คำกริยาที่เมื่อทำแล้วมีความหมายว่าได้ผลลัพธ์บางอย่างออกมา เช่น "คำนวณ" สิ่งนั้นสิ่งนี้ หรือ "เปรียบเทียบ" สิ่งนั้นกับสิ่งนี้เป็นต้น คำกริยาที่อาจจะเลือกใช้ได้ เช่น
GET COMPUTE
PUT DELETE
FIND VALIDATE
ADD MOVE
SUBTRACT REPLACE
MULTIPLR SET
DIVIDE SORT เป็นต้น
- ใช้ชื่อข้อมูลเป็นคำนามในประโยค ตัวอย่างเช่น วันชำระใบทวงหนี้ รายงานเพื่อเตรียมเงินสด เป็นต้น
- ใช้คำศัพท์ที่แสดงความสัมพันธ์ระหว่างข้อมูลเช่น "และ" "หรือ" "เท่ากับ" "ไม่เท่ากับ" "มากกว่า" และ "น้อยกว่า" เป็นต้น
- ใช้คำที่บอกการเคลื่อนที่ของข้อมูลคล้ายกับคำที่ใช้ในการเขียนโปรแกรมได้แก่
1. ถ้า..…มิฉะนั้น (if…..else…..)
2. กรณี….. (case)
3. ทำซ้ำ (Do3…..loop)
4. ทำตามลำดับ (Sequence)

**วิธีการตัดสินใจแบบตาราง**
การตัดสินใจแบบตารางเป็นตาราง 2 มิติ โดยที่แถวตั้งด้านซ้ายเป็นเงื่อนไข และแถวนอนเป็นรายละเอียดของ เงื่อนไขและผลการตัดสิน เงื่อนไขก็คือ สิ่งที่มีค่าเปลี่ยนแปลงได้

**สรุปประโยคโครงสร้าง**
ประโยคโครงสร้างและตารางตัดสินใจเป็นเครื่องมือที่ใช้อธิบายการทำงานภายในของโพรเซส เราควรจะเลือกวิธีที่เขียนอธิบายการทำงานที่ดีและเหมาะสมกับงานนั้นๆ

**พจนานุกรมข้อมูล**
ทุกๆ ระบบแม้แต่ระบบที่ง่ายที่สุดก็ต้องมีข้อมูล ตัวอย่างเช่น ระบบบัญชีเจ้าหนี้จะต้องมีข้อมูลที่จำเป็นในการจ่ายเงินแก่เจ้าหนี้ได้แก่ ชื่อ และที่อยู่ของเจ้าหนี้ จำนวนเงินที่ต้องชำระ เป็นต้น ระบบที่พัฒนาเสร็จต้องติดตามข้อมูลเหล่านี้ได้ แต่สำหรับนักวิเคราะห์ระบบจะเก็บข้อมูล รูปแบบ (format) หรือการใช้งานข้อมูลเหล่านี้อย่างไร คำตอบก็คือ พจนานุกรม ข้อมูลซึ่งเป็นที่เก็บรวบรวมข้อมูลทั้งหมดจะเป็นที่ซึ่งเราจะหาข้อมูลที่ต้องการเกี่ยวกับข้อมูลของระบบทั้งหมดได้ พจนานุกรมข้อมูลควรสร้างไว้ใช้ตั้งแต่เริ่มโครงการโดยเริ่มต้นจาก จำนวนข้อมูลน้อยๆ และเริ่มเพิ่มขึ้นเรื่อยๆ โครงการเล็กๆ จะมีข้อมูลประมาณ 1,000 ข้อมูล สำหรับโครงการใหญ่ อาจจะมีมากถึง 50,000 ข้อมูลด้วยกัน

**การจัดพจนานุกรมข้อมูล**
พจนานุกรมข้อมูลถูกเชียนเรียบเรียงสำดับตามตัวอักษรโดยไม่สนใจว่าข้อมูลนั้นจะเป็นแบบไหนคือเขียนรวมปะปนกันทั้งข้อมูลเดี่ยว ข้อมูลที่ไหลใน DFD หรือเก็บในไฟล์ แล้วนำมาเรียงลำดับ (sorted) ตามตัวอักษรแบบเรียงจากน้อยไปหามาก ปกติเราจะเขียนพจนานุกรมด้วยมือ แต่ปัจจุบันก็มีคอมพิวเตอร์ซอฟแวร์ที่ช่วยเราทำงานนี้ได้ด้วย ปกติแล้ว พจนานุกรมาข้อมูลมีขนาดใหญ่มาก จึงเป็นการสะดวกมากกว่าถ้าใช้คอมพิวเตอร์ช่วย

**การสร้างแบบสำหรับระบบใหม่**
ในบทนี้เราจะใช้เครื่องมือต่างๆ ที่ได้กล่าวมาแล้วมาช่วยในการสร้างระบบใหม่ เครื่องมือเหล่านี้ได้แก่ พจนานุกรม แผนภาพกระแสข้อมูล (Data Flow Diagram : DFD) คำอธิบายโพรเซสและฐานข้อมูล นักวิเคราะห์ระบบจะใช้แบบของระบบใหม่นี้ เพื่อทดสอบความคิดและให้เห็นภาพลักษณ์ของระบบใหม่ด้วย แบบของระบบใหม่นี้คล้ายๆ กับแบบแปลนบ้านของสถาปนิกนั่นเอง แบบแปลนนี้จะทำให้เรานี้ว่าบ้านที่เราจะปลูกมีน่าตาเป็นอย่างไรก่อนการลงมือก่อสร้างจริง กระบวนการสร้างแบบของระบบใหม่ช่วยให้เราปรับปรุงระบบใหม่ให้ดีขึ้นกว่าระบบปัจจุบันที่กำลังใช้งานอยู่ด้วย

ความคิดหลายๆ ความคิดที่ใช้ในที่นี้มาจาก DeMarco (1979) และ McMenamin and Palmar (1985) ความคิดทั้งหมดมีความสัมพันธ์กับความคิดเดิมของ Yourdon Inc. ซึ่งเป็นบริษัทชั้นแนวหน้าในเรื่องการวิเคราะห์ระบบแบบโครงสร้าง

ก่อนที่จะเริ่มการสร้างระบบ เราลองมาทำความเข้าใจคำว่าลอจิคัลและฟิสิคัล เราสมมุติเราพูดถึง "จ่ายเงินใบ ทวงหนี้" จะเพียงว่าทำอะไร แต่ถ้าพูดว่าการจ่ายเงินอาจจะทำ ได้หลายวิธี เช่น เขียนเช็คด้วยมือ หรือเขียนเช็คด้วยเครื่องคอมพิวเตอร์อัตโนมัติ หรืออาจจะใช้วิธีโอนเงินระหว่างธนาคารก็ได้ ดังนั้นที่พูดว่า "จ่ายเงินใบทวงหนี้" จะเป็น วิธีการ จ่ายเงินทั้ง 3 วิธีจะเป็นฟิสิคัล ลอจิคัล เป็นกิจกรรมสำคัญของระบบ ซึ่งจะคงอยู่ในระบบใหม่ที่พัฒนาขึ้นมาด้วย โดยไม่ขึ้นอยู่กับเทคโนโลยีในทางปฏิบัติ

ปกตินักวิเคราะห์ระบบใช้แบบระบบด้วยกัน 4 แบบในการทำงาน จากระบบปัจจุบันสู่ระบบใหม่โดยเริ่มต้นจาก 1. ฟิสิคัลของระบบปัจจุบันแก้ไขเป็น 2 ลอจิคัลของระบบปัจจุบัน แล้วจึงเขียนเป็น 3. ลอจิคัลบของระบบใหม่ และสุดท้าย ที่เป็น 4. . ฟิสิคัลของระบบใหม่ แบบของระบบทั้ง 4 มีความสัมพันธ์กันดังแสดงในรูป

รูปที่ 6.13 การสร้างแบบระบบ

**การออกแบบระดับกายภาพ**
การออกแบบใระดับกายภาพแตกต่างจากระดับตรรกะในแง่ของการแสดงขั้นตอนของระบบ โดยจะให้ความสำคัญเกี่ยวกับรายละเอียดของข้อมูล ผลลัพธ์ และการประมวลผล รวมถึงชนิดของสื่อที่ใช้ในการบรรจุข้อมูลด้วย

มาดูตัวอย่างการออกแบบที่ดีกันสักตัวอย่างหนึ่ง ได้แก่ โปรแกรมในการคำนวณบัญชี เงินเดือนของบริษัทแห่งหนึ่ง ซึ่งผลที่ได้จากการคำนวณนี้จะถูกนำไปปรับค่าข้อมูล ในไฟล์หลักของข้อมูลพนักงานด้วย ดังแสดงไว้ด้วยผังงานระบบ(System Flowchart) การออกแบบระบบในระดับนี้จะระบุถึงข้อมูล และผลลัพธ์ที่ได้จากการทำงาน รวมถึงขั้นตอน ในการทำงานในกรณีที่เกิดข้อผิดพลาดขึ้นความสำคัญของผังงานระบบนี้เปรียบเสมือนเป็น "พิมพ์เขียว" ที่จะใช้ในการพัฒนาระบบต่อไป

ข้อมูลของระบบนี้ จากการวิเคราะห์สรุปว่า ข้อมูลได้แก่ ไฟล์รายการเปลี่ยนแปลงซึ่งบรรจุอยู่ในเทปเรคอร์ดในไฟล์ที่ว่านี้ระบุจำนวนชั่วโมงทำงาของพนักงานแต่ละคน และเพื่อให้ การทำบัญชีเงินเดือนเป็นไปโดยีประสิทธิภาพ เราจะนำข้อมูลดังกล่าวมาเรียงลำดับตามรหัสประจำตัวพนักงานกันก่อน

ส่วนข้อมูลอีกส่วนหนึ่งได้แก่ ไฟล์ข้อมูลหลักที่เก็บข้อมูลของพนักงานเอาไว้ข้อมูลที่ว่านี้ได้แก่ รหัสพนักงาน ชื่อ ที่อยู่ และอัตราค่าแรงของพนักงานแต่ละคน เงินเดือนที่ได้รับ ตั้งแต่ต้นปี และเงินเดือนหลังหักภาษีแล้ว เป็นต้น และก็ เช่นเดียวกันคือเรคอร์ดในไฟล์หลักนี้จะเรียงลำดับไว้ตามรหัสประจำตัวพนักงาน เพื่อให้สะดวกในการทำงาน

ผลลัพธ์ที่ต้องการจากโปรแกรมบัญชีเงินเดือน ย่อมได้แก่การพิมพ์เช็คเงินเดือนที่จะจ่ายให้พนักงานและการ ปรับปรุงค่าเงินเดือนในไฟล์หลักโดยการเพิ่มค่าเงินเดือนที่ได้รับ ในเดือนนั้นเข้าไป นอกเหนือจากนั้นก็ได้แก่รายงานเงินเดือนที่จ่ายไปทั้งหมด พร้อมด้วยรายงานข้อผิดพลาดต่างๆ เช่น มีรหัสพนักงานปรากฏในไฟล์รายการเปลี่ยนแปลงที่ ไม่มีตัวตนในไฟล์หลัก เป็นต้น

รูปที่ 6.16 ผังงานระบบสำหรับโปรแกรมบัญชีเงินเดือน

การประมวลผลเกิดขึ้นในส่วนของโปรแกรมบัญชีเงินเดือน ได้แก่ ขั้นตอนของการอ่านไฟล์รายการเปลี่ยนแปลง และค้นหาเรคอร์ดของพนักงานที่ว่าในไฟล์หลักเพื่อคำนวณ เงินเดือนและเปลี่ยนแปลงค่านำไฟล์หลัก ขอให้สังเกตว่า ในการออกแบบระบบเราจะยังไม่แสดงรายละเอียดของวิธีการทำงานของโปรแกรม

นอกเหนือจากการประมวลผลแล้ว การดำเนินการที่สำคัญที่การออกแบบระบบที่ดีจะมองข้ามไปเสียไม่ได้ ได้แก่ การควบคุมการเกิดข้อผิดพลาดต่างๆ ส่วนของการควบคุมได้แก่ การตรวจเช็คข้อมูลในรายการเปลี่ยนแปลง เช่น การตรวจการสะกดคำผิด การตรวจดูการใส่ข้อมูลว่ามีรูปแบบถูกต้องหรือไม่ และการตรวจสอบค่าตัวเลข เช่น อาจจะนำตัวเลข ทั้งหมดที่คีย์เข้าไปแล้วมารวมกันเช็คดูกับค่ารวมที่หาเอาไว้ก่อนเป็นต้น

หลังจากการออกแบบระบบได้เสร็จสิ้นเรียบร้อยแล้ว นักวิเคราะห์ระบบจะต้องเสนอการออกแบบนี้เพื่อให้เจ้าของงานรับรองว่าเป็นไปตามความต้องการ จากนั้นจึงเริ่มเข้าสู่ขั้น ตอนต่อไปคือการสร้างระบบนั้นเอง

รายละเอียดที่ต้องการในนการออกแบบเฉพาะอย่างมักจะขึ้นอยู่กับการประยุกต์ใช้ ตัวอย่าง เช่นในรายงานความเชื่อถือด้านการเงินการออกแบบอาจต้องรวมถึงรายละเอียดของ ผู้กู้เกี่ยวกับ ที่อยู่ หมายเลขโทรศัพท์ การอ้างอิง ความ เชื่อถือด้านการเงิน สินทรัพย์และหนี้สิน เงินเดือน เป็นต้น ผู้พัฒนาระบบต้องมีความรอบคอบคือไม่ใส่รายละเอียดมากหรือ น้อยจนเกินไป การใส่รายละเอียดน้อยไปอาจเป็นผลให้การออกแบบล้มเหลวในตอนสุดท้าย แนวคิดการออกแบบถ้าทำอย่างหยาบๆ จะทำให้ผู้ใช้ไม่ประทับใจในระบบ หรือผู้ใช้อาจ จะไม่อยากพัฒนาการใช้ระบบเพิ่มเติม และอาจจะใช้งานไม่ได้ การใส่รายละเอียดมากเกินไป อาจทำให้ต้องเสียเวลามาก และเป็นสาเหตุให้ผู้ใช้กลัวเพราะหาข้อมูลยาก หาไม่พบเมื่อเป็นเช่นนี้ผู้ใช้จำเป็นต้องเรียกผู้ออกแบบมาทบทวนการทำงานให้

การออกแบบส่วนแสดงผลของระบบ (Designing systems outputs) มีปัจจัย 6 ประการที่ควรนำมาพิจารณาในการออกแบบส่วนแสดงผล ได้แก่

เนื้อหา(Content)
รูปร่าง(Form)
ปริมาณ(Volume)
ทันเวลา(Timeliness)
สื่อ(Media)
รูปแบบ(Format)
เราได้เรียนรู้การสร้างผังงานโครงสร้างมาแล้วต่อไปควรสามารถที่จะตรวจสอบและประเมินได้ว่าผังงานโครงสร้าง ที่ได้มานั้นดีหรือไม่ เพื่อใช้เป็นหลักเกณฑ์ในการตัดสิน ใจเลือกแนวทางการออกแบบต่อไป

การตรวจสอบการยึดเกราะเป็นสิ่งที่ใช้วัดว่างานในแต่ละโมดูลเป็นอันหนึ่งอันเดียวกันหรือไม่ เราจะต้องตัดสินใจว่าโมดูลใดมีการยึดเกาะอย่างเหนียวแน่นมาก ถ้าโมดูลนั้น ไม่สามารถแบ่งย่อยได้อีกแล้ว ตัวอย่างโมดูลที่มีการยึดเกาะสูงได้แก่ โมดูลเขียนรายงานเป็นต้น สิ่งที่เราต้องการในการออกแบบคือโมดูลที่มีการยึดเกาะสูง ในทางตรงกันข้ามกัน ถ้าโมดูลที่ทำงานหลายอย่างที่ไม่เกี่ยวข้องกันเช่นโมดูลเขียนรายงานน และคำนวณจะเป็นโมดูลที่มีการเกาะต่ำซึ่งไม่ใช่โมดูลที่เราต้องการนำไปใช้ในการออกแบบ

ความเข้าใจเกี่ยวกับการยึดเกาะคงจะชัดเจนขึ้นบ้างแล้ว ในข่วงเวลาการวิเคราะห์เราพยายามแยกโพรเซสย่อยลงไปเรื่อยๆ เพื่อให้ได้โพรเซสต่ำลงสุดทำงานเพียงงานเดียว หลักการอันเดียวกันนี้ก็เป็นหลักการที่คล้ายคลึงกับการสร้างผังงานโครงสร้างที่ดีเช่นเดียวกัน

**การออกแบบระดับกายภาพ**
การออกแบบใระดับกายภาพแตกต่างจากระดับตรรกะในแง่ของการแสดงขั้นตอนของระบบ โดยจะให้ความสำคัญเกี่ยวกับรายละเอียดของข้อมูล ผลลัพธ์ และการประมวลผล รวมถึงชนิดของสื่อที่ใช้ในการบรรจุข้อมูลด้วย

มาดูตัวอย่างการออกแบบที่ดีกันสักตัวอย่างหนึ่ง ได้แก่ โปรแกรมในการคำนวณบัญชี เงินเดือนของบริษัทแห่งหนึ่ง ซึ่งผลที่ได้จากการคำนวณนี้จะถูกนำไปปรับค่าข้อมูล ในไฟล์หลักของข้อมูลพนักงานด้วย ดังแสดงไว้ด้วยผังงานระบบ(System Flowchart) การออกแบบระบบในระดับนี้จะระบุถึงข้อมูล และผลลัพธ์ที่ได้จากการทำงาน รวมถึงขั้นตอน ในการทำงานในกรณีที่เกิดข้อผิดพลาดขึ้นความสำคัญของผังงานระบบนี้เปรียบเสมือนเป็น "พิมพ์เขียว" ที่จะใช้ในการพัฒนาระบบต่อไป

ข้อมูลของระบบนี้ จากการวิเคราะห์สรุปว่า ข้อมูลได้แก่ ไฟล์รายการเปลี่ยนแปลงซึ่งบรรจุอยู่ในเทปเรคอร์ดในไฟล์ที่ว่านี้ระบุจำนวนชั่วโมงทำงาของพนักงานแต่ละคน และเพื่อให้ การทำบัญชีเงินเดือนเป็นไปโดยีประสิทธิภาพ เราจะนำข้อมูลดังกล่าวมาเรียงลำดับตามรหัสประจำตัวพนักงานกันก่อน

ส่วนข้อมูลอีกส่วนหนึ่งได้แก่ ไฟล์ข้อมูลหลักที่เก็บข้อมูลของพนักงานเอาไว้ข้อมูลที่ว่านี้ได้แก่ รหัสพนักงาน ชื่อ ที่อยู่ และอัตราค่าแรงของพนักงานแต่ละคน เงินเดือนที่ได้รับ ตั้งแต่ต้นปี และเงินเดือนหลังหักภาษีแล้ว เป็นต้น และก็ เช่นเดียวกันคือเรคอร์ดในไฟล์หลักนี้จะเรียงลำดับไว้ตามรหัสประจำตัวพนักงาน เพื่อให้สะดวกในการทำงาน

ผลลัพธ์ที่ต้องการจากโปรแกรมบัญชีเงินเดือน ย่อมได้แก่การพิมพ์เช็คเงินเดือนที่จะจ่ายให้พนักงานและการ ปรับปรุงค่าเงินเดือนในไฟล์หลักโดยการเพิ่มค่าเงินเดือนที่ได้รับ ในเดือนนั้นเข้าไป นอกเหนือจากนั้นก็ได้แก่รายงานเงินเดือนที่จ่ายไปทั้งหมด พร้อมด้วยรายงานข้อผิดพลาดต่างๆ เช่น มีรหัสพนักงานปรากฏในไฟล์รายการเปลี่ยนแปลงที่ ไม่มีตัวตนในไฟล์หลัก เป็นต้น

รูปที่ 6.16 ผังงานระบบสำหรับโปรแกรมบัญชีเงินเดือน

การประมวลผลเกิดขึ้นในส่วนของโปรแกรมบัญชีเงินเดือน ได้แก่ ขั้นตอนของการอ่านไฟล์รายการเปลี่ยนแปลง และค้นหาเรคอร์ดของพนักงานที่ว่าในไฟล์หลักเพื่อคำนวณ เงินเดือนและเปลี่ยนแปลงค่านำไฟล์หลัก ขอให้สังเกตว่า ในการออกแบบระบบเราจะยังไม่แสดงรายละเอียดของวิธีการทำงานของโปรแกรม

นอกเหนือจากการประมวลผลแล้ว การดำเนินการที่สำคัญที่การออกแบบระบบที่ดีจะมองข้ามไปเสียไม่ได้ ได้แก่ การควบคุมการเกิดข้อผิดพลาดต่างๆ ส่วนของการควบคุมได้แก่ การตรวจเช็คข้อมูลในรายการเปลี่ยนแปลง เช่น การตรวจการสะกดคำผิด การตรวจดูการใส่ข้อมูลว่ามีรูปแบบถูกต้องหรือไม่ และการตรวจสอบค่าตัวเลข เช่น อาจจะนำตัวเลข ทั้งหมดที่คีย์เข้าไปแล้วมารวมกันเช็คดูกับค่ารวมที่หาเอาไว้ก่อนเป็นต้น

หลังจากการออกแบบระบบได้เสร็จสิ้นเรียบร้อยแล้ว นักวิเคราะห์ระบบจะต้องเสนอการออกแบบนี้เพื่อให้เจ้าของงานรับรองว่าเป็นไปตามความต้องการ จากนั้นจึงเริ่มเข้าสู่ขั้น ตอนต่อไปคือการสร้างระบบนั้นเอง

รายละเอียดที่ต้องการในนการออกแบบเฉพาะอย่างมักจะขึ้นอยู่กับการประยุกต์ใช้ ตัวอย่าง เช่นในรายงานความเชื่อถือด้านการเงินการออกแบบอาจต้องรวมถึงรายละเอียดของ ผู้กู้เกี่ยวกับ ที่อยู่ หมายเลขโทรศัพท์ การอ้างอิง ความ เชื่อถือด้านการเงิน สินทรัพย์และหนี้สิน เงินเดือน เป็นต้น ผู้พัฒนาระบบต้องมีความรอบคอบคือไม่ใส่รายละเอียดมากหรือ น้อยจนเกินไป การใส่รายละเอียดน้อยไปอาจเป็นผลให้การออกแบบล้มเหลวในตอนสุดท้าย แนวคิดการออกแบบถ้าทำอย่างหยาบๆ จะทำให้ผู้ใช้ไม่ประทับใจในระบบ หรือผู้ใช้อาจ จะไม่อยากพัฒนาการใช้ระบบเพิ่มเติม และอาจจะใช้งานไม่ได้ การใส่รายละเอียดมากเกินไป อาจทำให้ต้องเสียเวลามาก และเป็นสาเหตุให้ผู้ใช้กลัวเพราะหาข้อมูลยาก หาไม่พบเมื่อเป็นเช่นนี้ผู้ใช้จำเป็นต้องเรียกผู้ออกแบบมาทบทวนการทำงานให้

การออกแบบส่วนแสดงผลของระบบ (Designing systems outputs) มีปัจจัย 6 ประการที่ควรนำมาพิจารณาในการออกแบบส่วนแสดงผล ได้แก่

เนื้อหา(Content)
รูปร่าง(Form)
ปริมาณ(Volume)
ทันเวลา(Timeliness)
สื่อ(Media)
รูปแบบ(Format)
เราได้เรียนรู้การสร้างผังงานโครงสร้างมาแล้วต่อไปควรสามารถที่จะตรวจสอบและประเมินได้ว่าผังงานโครงสร้าง ที่ได้มานั้นดีหรือไม่ เพื่อใช้เป็นหลักเกณฑ์ในการตัดสิน ใจเลือกแนวทางการออกแบบต่อไป

การตรวจสอบการยึดเกราะเป็นสิ่งที่ใช้วัดว่างานในแต่ละโมดูลเป็นอันหนึ่งอันเดียวกันหรือไม่ เราจะต้องตัดสินใจว่าโมดูลใดมีการยึดเกาะอย่างเหนียวแน่นมาก ถ้าโมดูลนั้น ไม่สามารถแบ่งย่อยได้อีกแล้ว ตัวอย่างโมดูลที่มีการยึดเกาะสูงได้แก่ โมดูลเขียนรายงานเป็นต้น สิ่งที่เราต้องการในการออกแบบคือโมดูลที่มีการยึดเกาะสูง ในทางตรงกันข้ามกัน ถ้าโมดูลที่ทำงานหลายอย่างที่ไม่เกี่ยวข้องกันเช่นโมดูลเขียนรายงานน และคำนวณจะเป็นโมดูลที่มีการเกาะต่ำซึ่งไม่ใช่โมดูลที่เราต้องการนำไปใช้ในการออกแบบ

ความเข้าใจเกี่ยวกับการยึดเกาะคงจะชัดเจนขึ้นบ้างแล้ว ในข่วงเวลาการวิเคราะห์เราพยายามแยกโพรเซสย่อยลงไปเรื่อยๆ เพื่อให้ได้โพรเซสต่ำลงสุดทำงานเพียงงานเดียว หลักการอันเดียวกันนี้ก็เป็นหลักการที่คล้ายคลึงกับการสร้างผังงานโครงสร้างที่ดีเช่นเดียวกัน

**แบบของการยึดเกาะ**
ข้างล่างนี้จะแบ่งการยึดเกาะออกเป็น 7 แบบด้วยกันคือ

|  |  |  |
| --- | --- | --- |
| **แบบ** | **ระดับความต้องการ** | **คำอธิบาย** |
| 1. หน้าที่ | ดี-การยึดเกาะสูง | คนเดียว |
| 2. ลำดับ | ดี | ข้อมูลร่วมกัน การเรียงลำดับมีความสำคัญ |
| 3. การติดต่อ | ดี | ข้อมูลร่วมกัน การเรียงลำดับไม่มีความสำคัญ |
| 4. วิธีการ | ดี | ไม่มีข้อมูลร่วมกัน การเรียงลำดับมีความสำคัญ |
| 5. เวลา | ดี | ไม่มีข้อมูลร่วมกัน ทำงานเวลาเดียวกัน |
| 6. เหตุผล | ดี | ทำงานขนานกัน แต่ไม่ขึ้นซึ่งกันและกัน |
| 7. บังเอิญ | เลว-การยึดเกาะต่ำ | กิจกรรมไม่สัมพันธ์กัน |

**แบบของข้อมูล**
มีอยู่ด้วยกัน 5 แบบในผังงานโครงสร้าง ดังแสดงไว้ในตารางที่ 10.2 โดยที่จะเริ่มจากแบบที่ดีในตอนต้นสู่แบบที่เลวในตอนท้ายตาราง ข้อมูล 2 แบบแรกนั้นยอมรับได้ ที่เหลือ 3 แบบนั้นยอมรับไม่ได้ ดังนั้นควรหลีกเลี่ยงไม่ควรใช้แบบนี้ เวลาเราจะประเมินแบบของข้อมูลเราจะแยกประเภทข้อมูลทั้งจากบนลงล่างหรือล่างขึ้นบน

|  |  |  |
| --- | --- | --- |
| **แบบ** | **ระดับความต้องการ** | **คำอธิบาย** |
| 1. ข้อมูลเดี่ยว | ดี-หลวม | ข้อมูลตัวเดียว |
| 2. ข้อมูลกลุ่ม | ดี | ข้อมูลกลุ่มในโครงสร้างเดียวกัน |
| 3. การควบคุม | ดี | แฟล็กควบคุม |
| 4. ร่วมกัน | ดี | เนื้อที่ร่วมกัน |
| 5. ข้อความ | เลว-แน่น  | การเปลี่ยนแปลงภายในโมดูล |

**การออกแบบฟอร์ม**
การออกแบบรูปแบบรายงาน รูปแบบข้อมูลขาเข้า และรูปแบบหน้าจอ เป็นเรื่องสำคัญทีเดียว เพราะว่าทั้งหมดคือ สิ่งที่ผู้ใช้เห็นได้ง่ายที่สุด และเป็นสิ่งที่ติดต่อระหว่างผู้ใช้กับ ระบบทั้งหมด และผู้ใช้จะใช้สิ่งที่เห็นเป็นตัวช่วยตัดสินว่าระบบดีหรือไม่ ถ้าสิ่งที่ผู้ใช้เห็นไม่ว่าเป็นรายงานหรือหน้าจอคอมพิวเตอร์ ถ้าดูไม่สวยงามดูวุ่นวายทำให้ผู้ใช้ไม่พอใจ ถึงแม้ว่าในระบบจริงๆ จะทำงานได้ดีมากก็ตาม พูดอีกนัยหนึ่งก็คือ ระบบของเราควรจะมีลักษณะที่ว่า "สวยทั้งรูป จูบก็หอม" นั่นเอง

เราจะต้องเริ่มต้นการออกแบบฟอร์มต่างๆ จากสิ่งที่ผู้ใช้ต้องการซึ่งเราควรจะได้ข้อมูลมาตั่งแต่การสัมภาษณ์ในช่วงการวิเคราะห์แล้ว เราจะเริ่มต้นออกแบบรายงานก่อน แล้วจึงออกแบบแบบฟอร์มของข้อมูลขาเข้า หลังจากนั้นจึงมาถึง ขั้นตอนการออกแบบหน้าจอคอมพิวเตอร์ เพราะว่าส่วนนี้จะเป็นทั้งแบบฟอร์มสำหรับข้อมูลขาเข้าและส่วนแสดง ผลลัพธ์ด้วย

รายงานอาจจะแบ่งออกได้เป็นหลายแบบขึ้นอยู่กับว่าใครเป็นผู้ใช้ข้อมูลที่พิมพ์ในรายงานนั้นและจำนวนพิมพ์ บ่อยครั้งเท่าไร รายงานอาจจะแบ่งออกเป็นชนิดต่างๆ ได้ดังนี้คือ

รายงานภายใน (Internal Report) เป็นรายงานที่ใช้ภายในองค์กรและใช้กับพนักงานในองค์กรเป็นส่วนใหญ่ ดังนั้นการออกแบบก็ควรคำนึงถึงการใช้งานและประโยชน์ ที่จะได้ในสายตาของผู้ใช้ รวมทั้งเรื่องราคาด้วย

รายงานภายนอก (External Report) เป็นรายงานที่เราส่งให้หน่วยงานภายนอก ตัวอย่างเข่น ใบทวงหนี้ลูกค้า รายงานสำหรับผู้ถือหุ้นและรายงานการปันผล รายงานแบบฟอร์ม ภาษีสำหรับรัฐบาลเป็นต้น สิ่งที่ควรคำนึงถึงนอกเหนือประโยชน์การใช้งานแล้ว จะต้องคำนึงถึงความถูกต้องตามกฎหมายและภาพลักษณ์ขององค์กร ตัวอย่างเข่น รายงาน ประจำปี ของบริษัท อาจจะต้องมีภาพสีสวยงามแสดงตัวเลขต่างๆ เพื่อให้คนภายนอกมีทัศนคติที่ดีต่อองค์กร สำหรับรายงานแบบฟอร์มภาษีก็ต้องผลิตให้ตรงกับที่หน่วยงานราชการต้องการเป็นต้น

รายงานยกเว้น (Exception Report) จะเป็นรายงานที่แสดงข้อมูลผิดพลาดแตกต่างจากข้อมูลธรรมดา ตัวอย่างเช่น รายงานเลขที่สินค้าผิดพลาดเป็นต้น

รายงานสรุป (Summery Report) เป็นรายงานแสดงของตัวเลขบางตัว และอาจจะเป็นรายงานสรุปจากรายงานรายละเอียดบางฉบับ ตัวอย่างเช่น รายงานสรุปยอดขายประจำปีก็คือ รายงานแสดงผลรวมยอดขายประจำเดือนเป็นต้น รายงานนี้จะมีประโยชน์ต่อฝ่ายบริหารมาก เพราะผู้บริหารไม่มีเวลาดูตัวเลขแต่ละวันแต่ต้องการดูยอดสรุปของแต่ละเดือน และของทั้งปี โดยเฉพาะตัวเลขที่แสดงจำนวนเงิน

รายงานตารางเวลาการทำงาน (Scheduled Reports) รายงานนี้จะผลิตตามคาบเวลา เช่น รายวัน รายสัปดาห์ รายเดือน และรายปี ซึ่งจะกำหนดแผนการทำงานและกิจกรรม ที่จะต้องทำตามคาบเวลานั้นๆ รายงานนั้นจะถูกส่งต่อไปยัง ผู้ที่มีหน้าที่เกี่ยวข้องกับกิจกรรมตามตารางการทำงานนั้นๆ

รายงานตามคำขอ (On-Demand Reports) เป็นรายงานที่จะพิมพ์เมื่อมีคำขอเท่านั้น ตัวอย่างเช่น รายงานลูกค้าของบริษัท เป็นต้น แต่การพิมพ์รายงานนี้อาจจะไม่ได้ในทันทีที่ขอ เนื่องจากเวลาคอมพิวเตอร์อาจจะไม่พอ เราอาจจะกำหนดไว้ว่ารายงานประเภทนี้จะพิมพ์เพียงวันละครั้งตอนเย็น เป็นต้น

**แบบของกระดาษ**
กระดาษที่ใช้พิมพ์รายงานมีอยู่ 2 ประเภทคือ กระดาษธรรมดา และกระดาษที่พิมพ์ข้อความไว้แล้ว (Preprinted forms) กระดาษธรรมดาก็คือ กระดาษเปล่าว่างๆ และเป็นชนิด ต่อเนื่องซึ่งมีอยู่ด้วยกันหลายขนาด และอาจจะทำเป็นหลายก๊อปปี้ด้วยก็ได้ สำหรับกระดาษที่พิมพ์ข้อความไว้แล้วนั้นจะมีข้อความบางข้อความที่พิมพ์ไว้ก่อนแล้ว ซึ่งข้อความเหล่านี้ ปกติจะไม่เปลี่ยนแปลง ตัวอย่างเช่น เช็ค่ซึ่งจะมีชื่อธนาคารพิมพ์อยู่ในทุกๆ ใบ ซึ่งเป็นข้อความที่ไม่เปลี่ยนแปลง และจะมีข้อความบางส่วนที่จะพิมพ์เพิ่มเติมลงไป เช่น ชื่อผู้รับเงิน เป็นต้น

**การออกแบบรายงาน**
ปกติเวลาเราออกแบบรายงานเราจะใช้แบบฟอร์มที่มีตาราง (Spacing Chart) ซึ่งมีลักษณะเป็นช่องๆ นำมากรอกข้อความที่ต้องการจะพิมพ์ ตัวอย่างแบบฟอร์มตารางที่ใช้ในการ ออกแบบรายงาน เราจะใส่ข้อความที่ไม่มีการเปลี่ยนแปลง เช่น ชื่อรายงานในแบบฟอร์มลงในตำแหน่งที่เราต้องการให้ข้อความปรากฏในรายงาน ในกรณีที่ข้อมูลที่เปลี่ยนแปลงได้ เราจะใช้ตัวอักษร "X" แทนตัวหนังสือ และ "9" แทนตัวเลข แล้วจึงเติมจำนวนบรรทัดที่ต้องการพิมพ์ในหน้านั้นๆ นอกจากนั้นอาจจะใส่ตัวอักษรพิเศษต่างๆ เช่น $, I, - หรือ . ด้วยก็ ได้ เราจะกรอกข้อมูลเฉพาะบรรทัดแรกและบรรทัด สุดท้ายเท่านั้น ระหว่าง 2 บรรทัดนั้นจะร่างเส้นโค้งเพื่อเชื่อมโยง ซึ่งจะหมายความว่า รูปแบบข้อมูลตรงกลางจะเหมือนกัน กับบรรทัดแรกและบรรทัดสุดท้ายนั่น รูปแบบข้อมูลตรงกลางจะเหมือนกันกับบรรทัดแรกและบรรทัดสุดท้ายนั่นเอง หัวเรื่องของรายงานอาจจะประกอบด้วยชื่อรายงาน วันที่ที่พิมพ์ เลขที่หน้าของ รายงาน ข้อมูลในแต่ละแถงควรจะมีช่องว่างแทรกเพื่อให้ดูรายละเอียดได้สบายตามากขึ้นและดูไม่แน่นจนเกินไป

**แบบฟอร์มข้อมูลขาเข้า**
ข้อมูลที่เราจะป้อนเข้าสู่ระบบก็ควรจะออกแบบให้อยู่ในรูปแบบที่ง่ายต่อพนักงานป้อนข้อมูลที่จะพิมพ์เข้าสู่คอมพิวเตอร์ ตัวอย่างเช่นใบทวงหนี้จะมีข้อความหลายอย่าง ดังนั้นแบบ ฟอร์มนี้จะต้องออกแบบให้ง่ายต่อการแปลงจาก ใบทวงหนี้เดิมให้อยู่ในรูปที่ง่ายต่อการกรอกข้อมูลและป้อนเข้าเครื่องคอมพิวเตอร์ ปกติเราจะออกแบบให้พนักงานป้อน ข้อมูลอ่าน แบบฟอร์มจากซ้ายไปขวา จากบนลงล่าง ข้อมูลที่สัมพันธ์กันควรอยู่ด้วยกัน ตัวอย่างเช่น ข้อมูลที่อยู่จะมี รายละเอียด บ้านเลขที่ ซอย ถนน ตำบล อำเภอ และจังหวัด เราควรจะแยก ข้อความให้อยู่ในหลายบรรทัด เช่น ประมาณ 3 บรรทัด เพื่อผู้กรอกจะได้กรอกข้อความได้ครบถ้วน สำหรับข้อมูลประเภทตัวอักษรและตัวเลข ถ้าเป็นไปได้เราควรจะแยกเป็น กลุ่มออกจากัน เพื่อผู้ป้อนข้อมูลเป็นชุดๆ เช่น ตัวเลขทั้งหมดและจึงป้อนตัวอักษรทั้งหมดซึ่งจะทำให้การเคลื่อนไหวของมือทำได้สะดวกและไม่เสียเวลาด้วย

ในกรณีข้อมูลที่จะกรอกเป็นข้อมูลที่ให้เลือก เราควรจะออกแบบเป็นกรอบสี่เหลี่ยมให้เช็ค เช่น ตัวอย่างถามว่าอาชีพอะไร เราอาจออกแบบดังนี้

**การออกแบบจอภาพ**
จอภาพของคอมพิวเตอร์เป็นไปได้ทั้งตัวรับข้อมูลและแสดงผลลัพธ์ จอภาพมีประโยชน์สำหรับแสดงผลลัพธ์ ในกรณีที่เราไม่ต้องการพิมพ์รายงานบนกระดาษ แต่ต้องการดูผลอะไร บางอย่าง เช่น ดูสถานะเครดิตลูกค้าเฉพาะรายเป็นต้น นอกจากนั้นปัจจุบันเราก็นิยมพิมพ์ข้อมูลเข้าหรืออินพุตผ่านทางหน้าจอ เช่น ป้อนข้อมูลการเคลื่อนไหวของสินค้าเป็นต้น
ปกติหน้าจอคอมพิวเตอร์มีขนาด 80\*25 (25 บรรทัด บรรทัดละ 80 ตัวอักษร) ดังนั้นเราจะมีจำนวนจำกัดในการแสดงข้อความบนจอ แบบฟอร์มจอภาพ (Screen Layout) (คล้าย ๆ Spacing Chart ของการออกแบบรายงาน) จะช่วยในการออกแบบจอภาพ โดยที่เราจะกรอกข้อความที่จะให้ปรากฏจอคอมพิวเตอร์ในแบบฟอร์มนี้

**เมนู**
เนื่องจากแต่ละระบบจะมีโปรแกรมอยู่มากมาย ดังนั้นเราจะใช้เมนูเพื่อช่วยเลือกงานที่เราต้องการจะทำ คล้ายกับการเลือกอาหารจากเมนูในร้านอาหาร โดยที่เมนูบนจอภาพ จะมีรายการต่างๆ ที่เราจะเลือกทำได้ บรรทัดสุดท้ายจะเป็นที่ที่ รอรับคำสั่งจากผู้ใช้ ถ้าเราเลือกเลข 5 งานระบบบัญชีก็จะถูกเรียกใช้และวิ่งไปสู่เมนูระดับถัดไป เราเลือกเลข 3 เมนูใหม่ก็จะปรากฏขึ้นมา ดังนั้นเมนูก็จะมีระดับแยกย่อยลึกลงไปเรื่อย ๆ

**แบบของข้อมูล**
มีอยู่ด้วยกัน 5 แบบในผังงานโครงสร้าง ดังแสดงไว้ในตารางที่ 10.2 โดยที่จะเริ่มจากแบบที่ดีในตอนต้นสู่แบบที่เลวในตอนท้ายตาราง ข้อมูล 2 แบบแรกนั้นยอมรับได้ ที่เหลือ 3 แบบนั้นยอมรับไม่ได้ ดังนั้นควรหลีกเลี่ยงไม่ควรใช้แบบนี้ เวลาเราจะประเมินแบบของข้อมูลเราจะแยกประเภทข้อมูลทั้งจากบนลงล่างหรือล่างขึ้นบน

|  |  |  |
| --- | --- | --- |
| **แบบ** | **ระดับความต้องการ** | **คำอธิบาย** |
| 1. ข้อมูลเดี่ยว | ดี-หลวม | ข้อมูลตัวเดียว |
| 2. ข้อมูลกลุ่ม | ดี | ข้อมูลกลุ่มในโครงสร้างเดียวกัน |
| 3. การควบคุม | ดี | แฟล็กควบคุม |
| 4. ร่วมกัน | ดี | เนื้อที่ร่วมกัน |
| 5. ข้อความ | เลว-แน่น  | การเปลี่ยนแปลงภายในโมดูล |

รูปที่ 6.17 จอภาพของเมนู

จำนวนบรรทัดในเมนูไม่ควรจะมากเกินไป ประมาณ 7 บรรทัดดูเหมือนจะมากที่สุดและเวลาที่เลือกเมนูระดับล่างแล้วก็ควรจะวิ่งกลับมาเมนูแม่ของมันได้ เช่น โดย การกด "ESC" คีย์ เป็นต้น

**การพัฒนาโปรแกรมและการบำรุงรักษา**หลักการวิเคราะห์และออกแบบอย่างเดียวยังมิได้รับประกันความสำเร็จของระบบ เมื่อเสร็จสิ้นขั้นตอนการวิเคราะห์ออกแบบแล้ว เราต้องเริ่มพัฒนาโปรแกรมสำหรับระบบใหม่นี้ การพัฒนาโปรแกรมในขั้นนี้จะรวมถึงการเขียนโปรแกรม ทดสอบและปรับปรุง เพื่อให้ได้ระบบที่มีประสิทธิภาพสูงสุด ในขณะเดียวกันเราจะเริ่มอบรมผู้ใช้ และเตรียมสถานที่ ให้พร้อมสำหรับคอมพิวเตอร์ (ในกรณีที่ซื้อใหม่หรือโยกย้าย) ขึ้นถัดมาเมื่อเริ่มนำโปรแกรมที่เขียนได้มาใช้งาน จะต้องถ่าย ข้อมูลเดิมเข้าสู่ระบบใหม่นี้ แล้วจึงเริ่มต้นใช้งานระบบ ใหม่ การบำรุงรักษาในขั้นตอนการพัฒนาระบบจะรวมถึงการบำรุงรักษาประจำวันคือ 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 ถ้าพบในขั้น ตอนเขียนโปรแกรมและทดสอบระบบ 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 ถ้าผลลัพธ์แตกต่างกันในเวลาหนึ่งเวลาใดเราจะต้องตรวจสอบว่าระบบใหม่มีปัญหาอะไรแล้วทำการแก้ไขให้ถูกต้อง และทำการเตรียมสถานที่ (Site Preparation) ที่เหมาะสมกับการ ใช้งานต่อไป

**คู่มือผู้ใช้และการอบรมผู้ใช้**
นักวิเคราะห์ระบบจะต้องมีส่วนเกี่ยวข้องในการเตรียมคู่มือผู้ใช้และจัดเตรียมการอบรมผู้ใช้ในบางครั้งนักวิเคราะห์ระบบจะเป็นผู้ดูแลงานเหล่านี้เอง 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 แต่ถ้าเป็นการเปลี่ยนจากระบบที่ไม่ได้ใช้คอมพิวเตอร์เป็นแบบที่ใช้คอมพิวเตอร์วิธีนี้เป็นวิธีที่หลีกเลี่ยงไม่ได้
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

เมื่อรวมโปรแกรมแล้ว เราจะต้องทดสอบว่าระบบทำงานต่อไปนี้ได้อย่างถูกต้องหรือไม่
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 และการวิเคราะห์ค่าใช้จ่าย และผลประโยชน์

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

**เทคนิคในการคาดคะเน**
การวางแผนงานมีประโยชน์มาก ถ้าการคาดคะเนถูกต้อง การคาดคะเนที่ถูกต้องทั้งเวลาค่าใช้จ่าย และ ผลประโยชน์ จะเป็นหัวเลี้ยวหัวต่อสำคัญของการส่งมอบระบบให้ตรงเวลา และค่าใช้จ่ายไม่บานปลาย แต่เป็นที่น่าเสียดายที่ไม่มีเทคนิคที่แน่นอนตายตัว ปกติแล้วเทคนิคที่ช่วยในการคาดคะเนนั้นจะมาจาก "ความหยั่งรู้" หรือ "ประสบการณ์" ที่มีอยู่แล้ว จากระบบที่คล้ายคลึงกัน หรือมาจากความรู้ที่กว้างขวางในระบบทั่วๆ ไป
วิธีการที่ดีกว่าที่จะช่วยในการคาดคะเนได้แก่ การนับลักษณะพิเศษของระบบ ตัวอย่างเช่น จำนวนระบบงานที่จะต้องทำ จำนวนระบบที่จะเชื่อมต่อกัน รูปแบบของโครงสร้างข้อมูล แล้วพยายามหาสูตรสำหรับคาดคะเนโดยยึดมาตรฐานที่วัดได้ ในที่นี้เราจะพยายามคาดคะเนการวางแผน และควบคุมโดยใช้สูตรแทนที่จะใช้วิธี "หยั่งรู้"

**สูตรในการคาดคะเน**
สูตรที่เฉพาะเจาะจงจะช่วยให้เราคาดคะเนสิ่งต่างๆ ในโครงการได้ แต่สูตรนี้จะต้องกว้างพอที่จะใช้ได้กับทุกๆ ระบบ ถ้าจะต้องมีการแก้ไขให้เข้ากับระบบใดระบบหนึ่งก็ควร จะแก้ไขแต่เพียงเล็กน้อยเท่านั้น DeMacro แนะนำวิธีที่ เรียกว่า "แมทริกซ์" (Matrix) มาช่วยในการกำหนดหน้าที่ต่างๆ หรือส่วนประกอบในระบบ และให้น้ำหนักกับหน้าที่เหล่านี้ น้ำหนักในที่นี้อาจจะเป็นค่าใช้จ่าย เวลา หรือผลประโยชน์ที่จะได้รับจากระบบ วิธีการใช้แมทริกซ์นี้จะช่วยให้เราคาดคะเนจำนวนเงินและเวลาที่จะใช้ในการสร้างระบบใดระบบหนึ่ง ได้ สำหรับในที่นี้เราจะไม่กล่าวถึงความซับซ้อนของวิธีการแมทริกซ์ แต่จะกล่าวเฉพาะส่วนที่เกี่ยวข้องเท่านั้น

แมทริกซ์อันแรกสุดที่จะคาดคะเนในการวิเคราะห์และออกแบบก็คือ ส่วนประกอบของระบบในโมดูลระดับล่างสุดของผังงานโครงสร้าง ทั้งนี้เพราะว่าโมดูลระดับล่างสุดจะเป็นตัวที่บอกถึงขนาดของระบบ โมดูลระดับล่างอาจจะทำงานไม่เหมือนกัน เช่น โมดูลบางอันทำหน้าที่คำนาม แต่บางโมดูลทำหน้าที่พิมพ์รายงานเท่านั้น ดังนั้นเราจะให้น้ำหนัก หรือเวลากับโมดูลที่ทำหน้าที่คำนวณมากกว่าโมดูลที่ทำหน้าที่พิมพ์รายงานเป็นต้น

แมทริกซ์อันที่สองก็ได้แก่ สูตรที่เกี่ยวข้องกับสิ่งต่อไปนี้คือ การตัดสินใจ ไฟล์ ข้อมูลเดี่ยว ข้อมูลซ้ำ ข้อมูลให้เลือก รวมทั้งตัวชี้ระหว่างกลุ่มข้อมูล นอกจากนั้นจะต้องพิจารณา ถึงชนิดของระบบด้วยว่าเป็นแบบไหน เช่น ระบบ On-line จะต้องให้น้ำหนักมากกว่าระบบ Batch เป็นต้น

น้ำหนัก หรือเวลาที่จะให้ในแต่ละแมตริกซ์ในตอนเริ่มต้นนี้ อาจจะยังไม่ถูกต้องนัก เราจะต้องพยายามทำและ ปรับปรุงให้ดีขึ้นเรื่อยๆ จนกว่าจะพอใจ ไม่ว่าน้ำหนักที่เราให้จะเป็น เท่าไรก็ตาม เราจะต้องใส่แฟกเตอร์ตัวหนึ่งเรียกว่า แฟกเตอร์ปลอดภัย แฟกเตอร์ตัวนี้จะช่วยแก้ไขปัญหา สิ่งที่ไม่คาดคิดว่าจะเกิดชึ้นในโครงการ ซึ่งเป็นผลให้โครงการเสร็จ ล่าช้ากว่ากำหนด เราจะต้องใส่แฟกเตอร์ปลอดภัยนี้ในสูตรมาตรฐานในการจัดการธุรกิจมาช่วย สูตรนี้จะมีตัวแปร 3 ตัวคือ 1. เวลาที่คาดว่างานจะเสร็จอย่างรวดเร็วที่สุด 2. เวลาที่คิดว่าการทำงานเสร็จอย่างแน่นอน และ 3 . เวลาที่มากที่สุดที่งานนั้นจะเสร็จสมบรูณ์
ราจะคำนวณเวลาเฉลี่ยที่จะใช้ในการทำงานนั้นจากสูตรต่อไปนี้
เวลาเฉลี่ย = {(เวลาที่น้อยที่สุด) + 4(เวลาที่งานควรจะเสร็จ) + (เวลาที่มากที่สุด)} / 6
เวลาเฉลี่ยนี้จะช่วยแก้ปัญหาให้งานเสร็จลุล่วงแม้ว่าจะมีสิ่งที่ไม่คาดคิดเกิดขึ้นในโครงการและไม่ทำให้โครงการเสร็จล่าช้าไปมากนัก

**คาดคะเนเวลาที่จะใช้ในโครงการ**
เครื่องมือที่ใช้ในการเขียนตาราง-เวลาการทำงานได้แก่ แผนภาพ "PERT" (Performance Evaluation and Review Technique) การทำงานในโครงการจะเขียนเป็นแผนภาพ เครือข่าย (Network) ซึ่งจะบอกถึงลำดับการทำงานแต่ละขั้นตอนในโครงการ

**แผนภาพเครือข่ายการทำงานทั่วๆ ไป**
เครือข่าย (Network) คือ แผนภาพทีใช้แสดงกิจกรรมต่างๆ ที่จะต้องทำเพื่อให้โครงการสำเร็จลงอย่างมีระบบ ส่วนประกอบของเครือข่ายจะมีโหนด (Node) ซึ่งใช้แสดงเหตุการณ์ ต่างๆ โดยที่แต่ละโหนดจะเชื่อมโยงด้วยเส้นตรงแสดงถึงความสัมพันธ์ของกิจกรรมต่างๆ รวมทั้งเวลาที่จะต้องใช้ในการทำกิจกรรมนั้นๆ การเขียนตารางเวลาการทำงานโดยใช้เครือข่ายนี้มีอยู่ 2 วิธี คือ CPM (Critical Path Method) และ PERT (Program Evaluation and Review Technique)
CPM คือ เครื่องมือทางด้านการจัดการเพื่อใช้ในการกำหนด การรวม และวิเคราะห์ กิจกรรมต่างๆ ที่จะต้องทำในโครงการอย่างประหยัดที่สุดและให้เสร็จทันเวลา

**การคาดคะเนค่าใช้จ่าย**
การประมาณค่าใช้จ่ายจะคำนวณจากทรัพยากรต่างๆ ที่จะต้องใช้ในโครงการทรัพยากรที่จะกล่าวถึงได้แก่ บุคลากร เครื่องคอมพิวเตอร์ และเครื่องมือเครื่องใช้อื่นๆ
ค่าใช้จ่ายในการพัฒนาระบบ (Development Cost) คือ ค่าใช้จ่ายสำหรับทรัพยากรต่างๆ เมื่อเริ่มต้นสร้างระบบ ค่าใช้จ่ายในการพัฒนาช่วงเริ่มต้นได้
ค่าใช้จ่ายระหว่างการปฏิบัติงาน (Operating Cost) คือค่าใช้จ่ายที่จะเกิดขึ้นในแต่ละวันระหว่างที่ใช้งานระบบใหม่
เพื่อให้การประเมินค่าใช้จ่ายระบบใหม่ และดูมีประสิทธิภาพมากขึ้น ควรจะนำค่าใช้จ่ายของระบบเก่ามา เปรียบเทียบด้วย การคำนวณค่าใช้จ่ายของระบบเก่าจะง่ายกว่าของ ระบบใหม่ แต่ถึงอย่างไรก็ตามค่าใช้จ่ายบางอย่างของระบบเก่าก็คำนวณเป็นตัวเงินได้ยาก เช่น ค่าใช้จ่ายจากการเสียโอกาสในกรณีที่ผู้บริหารไม่สามารถติดตามสถานะทางการ เงินของบริษัทได้ เนื่องจากไม่มีรายงานด้านการเงินเป็นต้น

**การคาดคะเนผลประโยชน์**
ผลประโยชน์ที่แน่นอนจะได้จากระบบที่เสร็จสมบรูณ์แล้วจะแบ่งเป็นผลประโยชน์ที่มองเห็น คือผลประโยชน์ที่คำนวณเป็นตัวเงินได้ เช่น การลดค่าใช้จ่ายในการทำงานล่างเวลา และผลประโยชน์ ที่มองไม่เห็น คือประโยชน์ที่ไม่สามารถคำนวณเป็นตัวเงินได้ แต่จะให้ภาพลักษณ์ในด้านบวกของบริษัท

**การวิเคราะห์ค่าใช้จ่าย-ผลประโยชน์**
การวิเคราะห์ค่าใช้จ่าย-ผลประโยชน์จะช่วยในการประเมินทางเลือกต่างๆ และการประเมินจะเป็นหน้าที่ของฝ่ายบริหาร หน้าที่ของนักวิเคราะห์ระบบคือ เตรียมตัวเลขต่างๆ ไว้ให้ฝ่ายบริหารตัดสินใจ ถึงอย่างไรก็ตามนักวิเคราะห์ระบบจะต้องเข้าใจถึงวิธีการคำนวณตัวเลขเหล่านั้น และจะต้องตระหนักถึงข้อเท็จจริงว่าตัวเลขที่เขาเตรียมไว้นั้นมีผลต่อการลงทุนขององค์กร

**สรุป**

บทเรียนนี้คงช่วยให้นักศึกษาและผู้สนใจแนวทางในการเริ่มต้นการวิเคราะห์ระบบและออกแบบระบบสามารถมองภาพโดยรวมของระบบที่เราต้องการพัฒนา และสามารถใช้เครื่องมือในการ วิเคราะห์ระบบ เช่นเทคนิคการรวบรวมข้อมูล คำต้องการของผู้ใช้ระบบ การเขียนแผนภาพกระแสข้อมูล คำอธิบายการประมวลผล เพื่อเป็นพิมพ์เขียวหรือต้นแบบในการสร้างระบบใหม่ พร้อมทั้ง ได้แนวทางในการตรวจสอบแบบที่ออกมาดีหรือไม่โดยใช้หลักการตรวจสอบผังโครงสร้าง แผนภาพลำดับขั้น รูปแบบอินพุต/เอาต์พุต วิธีการออกแบบฟอร์ม แล้วจึงเริ่มการดำเนินการ พัฒนาโปรแกรม ติดตั้งระบบ โดยมีการวางแผนและควบคุมโครงการ ให้ระบบงานที่ออกแบบใหม่สำเร็จตามวัตถุประสงค์