



บทที่ 2

ลักษณะของต้นแบบซอฟต์แวร์และแฟ้มข้อมูล

ความหมายของต้นแบบซอฟต์แวร์

ต้นแบบ (Prototype) หมายถึง ต้นฉบับหรือรูปแบบเบื้องต้นของสิ่งประดิษฐ์ต่างๆ ซึ่งเป็นเพียงโครงร่างที่สะท้อนให้เห็นถึงลักษณะคร่าวๆของสิ่งประดิษฐ์นั้น ตามแนวความคิดของผู้ประดิษฐ์หรือผู้ที่ต้องการใช้สิ่งประดิษฐ์นั้น ต้นแบบที่สร้างขึ้นมาอาจต้องมีการปรับปรุงแก้ไขหลายครั้ง เพื่อให้สามารถถ่ายทอดความคิดของนักประดิษฐ์กับผู้ใช้ได้มองเห็นภาพพจน์ที่เป็นรูปธรรมร่วมกันได้ เช่น การก่อสร้างอาคารที่อยู่อาศัย วิศวกรจำเป็นต้องสร้างต้นแบบของอาคารตามความต้องการของเจ้าของ เพื่อเป็นการแสดงแนวความคิดของตนเองว่า อาคารที่จะสร้างขึ้นนี้มีรูปร่างลักษณะ ตรงตามที่ตกลงกันหรือไม่ และลักษณะที่เป็นอยู่จะมีปัญหาทางปฏิบัติในด้านใดบ้าง ถ้าเจ้าของอาคารเห็นว่าสมควรเพิ่มเติมส่วนใดเข้าไป หรือมีส่วนใดที่ไม่เหมาะสม จะได้ปรับปรุงแก้ไขก่อนที่จะลงมือก่อสร้างจริง เป็นต้น

ความหมายของการทำต้นแบบ อาจคล้ายกับการทำแบบจำลอง เนื่องจากเป็นการสร้างรูปแบบเบื้องต้นก่อนการประดิษฐ์เช่นเดียวกัน แต่แบบจำลองหมายถึง การย่อส่วนหรือการสร้างเลียนแบบคุณลักษณะทางกายภาพ ของสิ่งประดิษฐ์นั้น และสามารถทำได้ใน 2 ลักษณะคือ

1. สร้างแบบจำลองก่อนการประดิษฐ์ เพื่อให้ประดิษฐ์ตามแบบจำลอง
2. สร้างแบบจำลองหลังการประดิษฐ์ เพื่อเป็นการลอกเลียนแบบ

ต้นแบบต้องสร้างก่อนการประดิษฐ์เสมอ โดยเน้นถึงสาระสำคัญและขอบเขตที่เป็นจุดประสงค์หลักและเป้าหมายของการประดิษฐ์เท่านั้น รายละเอียดต่างๆที่ไม่มีผลโดย

ตรงต่อการประดิษฐ์ ก็ยังไม่จำเป็นต้องบรรจุลงไปในต้นแบบ เพราะอาจต้องมีการแก้ไขอีกหลายอย่าง ถ้ามีรายละเอียดมากเกินไป ความคล่องตัวในการปรับแต่งแก้ไขจะน้อยลง ทั้งยังผิดความมุ่งหมายของการทำต้นแบบอีกด้วย

การทำต้นแบบไม่ใช่การทำแบบจำลองคือ ไม่จำเป็นต้องย่อส่วนขนาด, รูปร่างลักษณะ, น้ำหนัก, หรือองค์ประกอบอื่นๆของสิ่งประดิษฐ์ที่จะสร้างขึ้นจริงในภายหลัง เพียงแต่มีส่วนที่คล้ายกันในหลักการคือ ทั้งต้นแบบและแบบจำลองไม่ใช่สิ่งประดิษฐ์ที่ต้องการ ไม่สามารถนำมาใช้งานได้ แต่มีไว้เพียงเพื่อให้ผู้ประดิษฐ์สามารถเข้าใจเป้าหมาย และความต้องการของผู้ใช้สิ่งประดิษฐ์นั้นได้อย่างถูกต้อง

ต้นแบบซอฟต์แวร์จึงหมายถึง การสร้างระบบงานรุ่นแรกซึ่งไม่จำเป็นต้องสะท้อนภาพโครงงานทั้งหมดของระบบจริง แต่ทำเฉพาะที่น่าสนใจและมีนัยสำคัญต่อการพัฒนาระบบงาน โดยมีจุดประสงค์ขั้นต้นเพื่อเป็นการทดลองใช้งาน (เสมือนเป็นระบบงานที่ทำเสร็จสมบูรณ์แล้ว) และศึกษาหาแนวทางของการพัฒนาระบบงานต่อไป (2) ต้นแบบซอฟต์แวร์ที่สร้างขึ้นมานี้สมควรเป็นระบบที่ผู้วิเคราะห์กับผู้ใช้ ร่วมกันแสดงความคิดเห็นและปรับปรุงแก้ไขให้ตรงกับวัตถุประสงค์ของงาน ซึ่งจะช่วยให้เห็นข้อเท็จจริง หรือสาระสำคัญของงานที่ยังอาจบกพร่องอยู่ได้ชัดเจนมากขึ้น การทำต้นแบบซอฟต์แวร์ไม่จำเป็นต้องใช้ภาษาระดับสูงเสมอไป อาจใช้ภาษารุ่นที่ 4 เช่น เครื่องมือสร้างรายงาน (Report Generator), ระบบจัดการฐานข้อมูล (Data Base System) เป็นต้น

คุณสมบัติของต้นแบบซอฟต์แวร์

คุณสมบัติที่สำคัญของต้นแบบซอฟต์แวร์ คือ สามารถสร้างต้นแบบได้อย่างรวดเร็ว และเสียค่าใช้จ่ายน้อย ในขณะที่เดียวกันทั้งผู้วิเคราะห์และผู้ใช้ สามารถที่จะปรับปรุงแก้ไข ได้อย่างสะดวกและรวดเร็วด้วย โดยพยายามเน้นให้เห็นรูปแบบต่างๆที่สมควรทำความเข้าใจ ตกลงร่วมกัน ดังนี้คือ (3)

1. ต้นแบบของบทบรรยาย (Dialogue Prototype)

หมายถึง ออกแบบข้อความคำอธิบายหรือบทบรรยายต่างๆบนจอภาพ เพื่อให้ผู้ใช้มีความเข้าใจรายละเอียดและวิธีการใช้โปรแกรมได้อย่างถูกต้อง เนื่องจากภาษาคำศัพท์ต่างๆที่ปรากฏบนจอภาพ ถ้าข้อความไม่ชัดเจนหรือไม่ตรงกับความหมายที่ผู้ใช้มีความคุ้นเคยอยู่ อาจทำให้ผู้ใช้เกิดความสับสนในระหว่างการทำงาน และอาจเลือกรายการคำสั่งผิดๆ ซึ่งเป็นผลให้ข้อมูลเกิดความเสียหายได้ แต่ถ้าใช้ภาษาหรือคำศัพท์ที่ผู้ใช้เข้าใจจะเป็นการตัดปัญหาการใช้งานในเบื้องต้นได้ เพราะระบบโปรแกรมบางอย่างอาจไม่จำเป็นต้องทำ และผู้ใช้เองก็ไม่ต้องเสียเวลาจำขั้นตอนการใช้โปรแกรมมากนัก

2. ระบบการเตรียมข้อมูล (Data Entry)

หมายถึง ส่วนที่ใช้สำหรับกรอกข้อมูลของผู้ใช้ เนื่องจากงานบางชนิดมีปริมาณข้อมูลที่ต้องประมวลผลเป็นจำนวนมาก ผู้ใช้บางกลุ่มมีหน้าที่กรอกข้อมูลเพียงอย่างเดียวจึงอาจต้องการทดสอบความสะดวก และรูปแบบวิธีการกรอกข้อมูลว่าต้องเตรียมการเบื้องต้นอย่างไรบ้าง เช่น เอกสารชนิดใดต้องใช้หรือไม่ใช้ในการกรอกข้อมูล ต้องลงรหัสในแบบฟอร์มก่อนหรือไม่ เป็นต้น ที่สำคัญคือ ถ้ากรอกข้อมูลผิดจะมีวิธีแก้ไขได้อย่างไร โปรแกรมจะช่วยเตือนหรือตรวจสอบเบื้องต้นให้ได้หรือไม่ ซึ่งรวมถึงการกรอกข้อมูลไม่ครบและกรอกข้อมูลเกินด้วย

3. ระบบการทำรายงาน (Reporting System)

หมายถึง การพิมพ์รายงานเพื่อแสดงให้เห็นว่าผลลัพธ์ของงานที่ต้องการนั้นมีความถูกต้องมากน้อยเพียงใด รวมทั้งเป็นเอกสารสำคัญที่ผู้ใช้ต้องการอีกด้วย และบางครั้งยังมีผลต่อเนื่องอีกคือ ต้องนำไปใช้เป็นข้อมูลสำหรับการทำงานขั้นต่อไป ผู้ใช้ส่วนมากจะสนใจว่ารายงานที่พิมพ์ออกมานั้นมีข้อแตกต่างจากรายงานที่ผู้ใช้ทำเองอย่างไร รายละเอียดของรายงานครบตามความต้องการ หรือใช้ประโยชน์ได้มากน้อยเพียงใด รูปแบบการพิมพ์อ่านได้ง่ายหรือยาก มีการเน้นในส่วนของสาระสำคัญหรือไม่

4. ระบบฐานข้อมูล (Data Base System)

หมายถึง การสร้างตัวอย่างฐานข้อมูลเพียงเล็กน้อย เพื่อนำไปใช้ทดสอบระบบอื่นๆ เช่น การพิมพ์รายงาน เป็นต้น ฐานข้อมูลนี้จะให้ประโยชน์ทั้งในด้านการทดสอบระบบและช่วยให้ผู้วิเคราะห์กับผู้ใช้สามารถสำรวจได้ว่า ฐานข้อมูลที่เหมาะสมกับระบบงานนั้นจะต้องประกอบด้วยอะไรบ้าง แหล่งที่มาของข้อมูลมีความเที่ยงตรงมากน้อยเพียงใด ถ้าต้องการทำระบบตรวจสอบหรือสำรองข้อมูล สมควรให้โปรแกรมทำในช่วงเวลาใด ผู้ใช้มีหน้าที่อย่างไรบ้างกับระบบดังกล่าว

5. การคำนวณและทำงานเชิงตรรกวิทยา (Calculation and Logic)

หมายถึง การกำหนดเงื่อนไขและคำนวณหาค่าตัวเลขต่างๆจากข้อมูลที่มีอยู่ในฐานข้อมูล (ซึ่งผู้ใช้กรอกไว้เอง) เช่น การคำนวณภาษีเงินได้บุคคลธรรมดา สามารถคำนวณได้จากอัตราเงินเดือนกับสถานภาพด้านต่างๆของพนักงาน เป็นต้น สิ่งที่โปรแกรมจะสามารถคำนวณให้ได้นี้ บางครั้งผู้ใช้อาจนึกไม่ถึงว่าจะทำได้ และผลลัพธ์ที่ได้มาซึ่งสามารถนำไปใช้เป็นข้อมูลในขั้นต่อไปได้ด้วย ฉะนั้นถ้าสามารถแสดงตัวอย่างการคำนวณให้ผู้ใช้เห็นผลลัพธ์ได้ การวิเคราะห์หรือวางแผนในส่วนอื่นๆอาจง่ายขึ้น ข้อมูลที่ต้องกรอกอาจน้อยลง การพัฒนาโปรแกรมจะได้ลดขั้นตอนและจำนวนโปรแกรมลงไปด้วย

6. ชุดคำสั่งสำเร็จประยุกต์ (Application Package)

หมายถึง โปรแกรมสำเร็จรูปเบื้องต้นที่สร้างให้กับกลุ่มผู้ใช้ ซึ่งมีหน้าที่ใช้งานโดยตรงก่อน เช่น ผู้ที่มีหน้าที่กรอกข้อมูลอย่างเดี๋ยวนั้น เป็นต้น บุคคลเหล่านี้ต้องสัมผัสกับข้อมูลและการทำงานของโปรแกรมจริงไม่ใช่ผู้นำผลลัพธ์ไปใช้โดยตรง เช่น เจ้าของกิจการ เป็นต้น เพื่อให้เกิดความเข้าใจ, ความสะดวกในการใช้โปรแกรม และภาระหน้าที่ในการบำรุงรักษาข้อมูลให้ถูกต้องตามข้อเท็จจริง ผู้วิเคราะห์จะได้ทราบแนวทางและหาผลสรุปที่เหมาะสม ในการตกแต่งแก้ไขโปรแกรมให้ได้ก่อนที่จะนำไปรวมกับระบบงานส่วนอื่นๆ

7. หลักการและแนวความคิด (Concept)

หมายถึง จุดประสงค์ที่แท้จริงของการสร้างระบบงานทั้งหมด ซึ่งจำเป็นอย่างยิ่งที่ต้องมีการวิเคราะห์ให้ละเอียดถี่ถ้วนก่อนที่จะสร้างระบบงานขึ้นมา มิฉะนั้นอาจต้องเสียค่าใช้จ่ายและเวลาในการปรับปรุงแก้ไขตลอดเวลา ทำให้บางครั้งการสร้างระบบใหม่ทั้งหมดอาจง่ายกว่า แต่ปัญหานี้สามารถป้องกันได้ ถ้ามีการวิเคราะห์หาจุดประสงค์ที่แท้จริงของระบบงานตั้งแต่ยังไม่ได้ลงมือสร้าง โดยอาศัยการทดสอบระบบฐานข้อมูล กรอกข้อมูลทางจอภาพ แสดงรูปแบบของรายงาน เป็นต้น สิ่งเหล่านี้สามารถทำได้อย่างรวดเร็วหรืออาจเสียเวลานับบ้าง แต่ก็ยังการสะดวกกว่าที่จะมาตามแก้ไขโปรแกรมในภายหลัง

ชนิดของต้นแบบซอฟต์แวร์

ต้นแบบซอฟต์แวร์ สามารถแบ่งออกได้เป็น 3 ประเภท ดังนี้คือ (2)

1. ต้นแบบชนิดใช้แล้วทิ้ง (Throw-it-away Prototyping)

ต้นแบบชนิดนี้มีจุดประสงค์สำหรับวิเคราะห์ความต้องการและข้อกำหนดของผู้ใช้ หรืออาจเรียกว่าเป็นต้นแบบชนิดกำหนดขอบเขต (Specification Prototype) ที่สำคัญคือ ต้องสามารถทำงานได้ในตัวเอง แต่ไม่ใช่ผลผลิตที่จะนำไปใช้งานจริง จึงมีประโยชน์ในแง่ของการออกแบบ สํารวจแนวทางการออกแบบและประเมินความเป็นไปได้ของแนวความคิดแบบใหม่ๆ รวมทั้งสามารถใช้เป็นตัวเปรียบเทียบผลงานที่ได้มาจากระบบงานจริง ตัวต้นแบบอาจสร้างขึ้นบนเครื่องคอมพิวเตอร์อื่นที่ไม่ใช่เครื่องเดียวกับเครื่องที่ใช้ทำงานจริงก็ได้ เนื่องจากรายละเอียดของระบบที่ต้องการนั้นยังอาจสับสนอยู่จึงไม่อาจทราบได้ว่าควรใช้เครื่องรุ่นใดแบบใดมาประยุกต์ใช้งาน การทำต้นแบบชนิดนี้เพียงเพื่อให้เห็นความเป็นไปได้ของการออกแบบระบบ ตามที่วิเคราะห์กันในเบื้องต้นเท่านั้น ภาษาที่สมควรนำมาสร้างต้นแบบชนิดนี้คือ ภาษาระดับสูง (Very High Level)

2. ต้นแบบชนิดวิวัฒนาการ (Evolutionary Prototyping)

จุดประสงค์ของการทำต้นแบบชนิดนี้เพื่อเป็นการแนะนำระบบงานที่จะพัฒนาให้กับองค์กรใดองค์กรหนึ่ง ในลักษณะพัฒนาและขยายระบบเพิ่มขึ้นทีละน้อยๆ จนกลายเป็นระบบงานทั้งหมดที่จะใช้ในองค์กรนั้น การสร้างต้นแบบจะแยกระบบออกเป็นส่วนๆ และนำกลับมารวมกันในภายหลัง ดังนั้นประเด็นหลักของการทำต้นแบบชนิดนี้คือ ต้องมีการเพิ่มเติมและปรับปรุงแก้ไขอยู่ตลอดเวลา โดยคิดตั้งให้ผู้ใช้ทดลองปฏิบัติงานได้ เพื่อศึกษาสภาวะแวดล้อมของการใช้งานที่คล้ายระบบจริง แต่มีผลกระทบต่อความเข้าใจในการวิเคราะห์ระบบงานทั้งหมด เมื่อออกแบบและวิเคราะห์ได้ครบทั้งระบบแล้วอาจใช้ต้นแบบที่สร้างไว้นี้เป็นโปรแกรมหลักของการทำงานจริง หรืออาจเขียนโปรแกรมขึ้นมาใหม่ให้ทำงานได้เหมือนกับต้นแบบก็ได้ ขึ้นอยู่กับว่าต้นแบบที่สร้างไว้ มีขอบเขตความสามารถค่อนข้างจำกัดจนนำมา

ใช้งานจริงไม่ได้เช่น ประมวลผลข้อมูลได้น้อย การคำนวณมีความลาดเคลื่อน เป็นต้น

3. ต้นแบบชนิดส่วนเพิ่ม (Incremental Prototyping)

ความหมายของต้นแบบชนิดนี้คล้ายกับแบบชนิดวิวัฒนาการคือ เป็นการทำต้นแบบทุกๆส่วนของระบบงานเหมือนกัน แต่ต่างกันตรงที่ว่าต้นแบบชนิดนี้ จะเน้นการออกแบบระบบงานทั้งหมดในคราวเดียว ไม่ใช่ใช้ออกแบบทีละส่วนไปเรื่อยๆจนกลายเป็นระบบใหม่ ดังนั้นต้นแบบชนิดส่วนเพิ่มจึงเหมาะกับระบบงานที่มีการตกลงทำความเข้าใจกัน ระหว่างผู้ใช้กับผู้วิเคราะห์เรียบร้อยแล้ว เพียงแต่ช่วยให้เข้าใจรายละเอียดการใช้งานและการควบคุมต่างๆได้ง่ายขึ้นเท่านั้น

แฟ้มฐานข้อมูล (Data Base)

แฟ้มฐานข้อมูล หมายถึง กลุ่มของแฟ้มข้อมูลที่ใช้งานร่วมกันในการประมวลผลของโปรแกรม และมีความเกี่ยวข้องกันในจุดประสงค์ของการใช้งาน แฟ้มข้อมูลที่ประกอบกันเป็นฐานข้อมูลนี้ จะประกอบด้วยรายการข้อมูลต่างๆ ซึ่งบันทึกไว้หรือนำกลับมาใช้งานตลอดเวลา (4)

ความสัมพันธ์ของแฟ้มฐานข้อมูล (Relational Data Base)

การวางแผนสร้างแฟ้มฐานข้อมูล สามารถกำหนดให้ข้อมูลมีความสัมพันธ์กันได้ 3 แบบ ดังนี้คือ

1. ตัวต่อตัว (One-to-one)

หมายถึง ข้อมูล 1 ระเบียบ (Record) จากแฟ้มข้อมูลหนึ่งมีความสัมพันธ์กับข้อมูล 1 ระเบียบที่อยู่ในแฟ้มข้อมูลอื่น

2. หนึ่งตัวต่อหลายตัว (One-to-many)

หมายถึง ข้อมูล 1 ระเบียบจากแฟ้มข้อมูลหนึ่งมีความสัมพันธ์กับข้อมูลหลายระเบียบที่อยู่ในแฟ้มข้อมูลอื่น

3. หลายตัวต่อหลายตัว (Many-to-many)

หมายถึง ข้อมูลหลายระเบียบจากแฟ้มข้อมูลหนึ่ง มีความสัมพันธ์กับข้อมูลอีกหลายระเบียบที่อยู่ในแฟ้มข้อมูลอื่น