

การพัฒนาระบบให้ประสบความสำเร็จ

ดร. ครรชิต มัลลียงศ์

เมื่อเร็ว ๆ นี้ผมได้มีโอกาสเห็นข้อมูลที่น่าสนใจเรื่องหนึ่งเกี่ยวกับการพัฒนาระบบงานคอมพิวเตอร์ในสหรัฐอเมริกา ข้อมูลนั้นแสดงว่าระบบงานที่พัฒนาแล้วประสบความสำเร็จด้วยดีนั้นมีน้อยกว่ายี่สิบเปอร์เซ็นต์ นอกนั้นเป็นระบบที่พัฒนาแล้วล้มเหลว

ได้เห็นข้อมูลแล้วก็ไม่แปลกใจที่การพัฒนาระบบคอมพิวเตอร์ในเมืองไทยเราเองก็มีข่าวเกี่ยวกับความล้มเหลวให้ได้ยินได้ฟังอยู่บ่อย ๆ

ฟังดูแล้วไม่ใช่เรื่องสนุกหรือครับ เพราะความล้มเหลวนั้นพิสูจน์ว่าเป็นความพ่ายแพ้ขององค์กร คือถ้าหากองค์กรจ้างบริษัทภายนอกมาพัฒนาระบบให้ ก็เป็นความพ่ายแพ้ของทั้งผู้ว่าจ้างและผู้รับจ้าง ผู้รับจ้างเองก็ไม่ได้เงิน หรือ อาจจะต้องเสียเงินเป็นจำนวนมากด้วยซ้ำ ส่วนผู้ว่าจ้างเองก็ไม่ได้งานที่ควรจะได้รับ และถ้างานนั้นสำคัญมาก การไม่ได้ระบบงานที่ควรจะได้ ก็อาจจะทำให้เสียหาย หรือ เสียประโยชน์ที่ควรจะได้รับอีกมากกว่าทีเดียว

เหตุผลที่การพัฒนาระบบทั้งหลายต้องมีอันล้มเหลวบ่อยครั้งนั้น มีผู้วิเคราะห์วิจัยกันมากแล้ว ผมเองก็นำเรื่องนี้มาเขียนหลายครั้งแล้ว อย่างไรก็ตามในช่วงนี้ผมได้รับข้อมูลและประสบการณ์ทางด้านนี้มากขึ้น จึงอยากจะนำมาเขียนให้อ่านในอีกรูปแบบหนึ่ง

นั่นคือมองในแง่ของความสำเร็จยิ่งกว่าความล้มเหลว

ความสำเร็จวัดกันที่ไหน

ก่อนจะกล่าวถึงลักษณะของการพัฒนาระบบให้ประสบความสำเร็จ เห็นจะต้องตกลงกันก่อนว่า ความสำเร็จของการพัฒนาระบบงานคืออะไร

มองในแง่ของผู้พัฒนา บางคนบอกว่าการพัฒนานั้นประสบความสำเร็จเมื่อทำโปรแกรมเสร็จตามที่ได้รับมอบหมาย และผู้ว่าจ้างจ่ายเงินให้ตามสัญญา ส่วนงานที่ล้มเหลวก็คืองานที่จบไม่ลงสักที ทำเท่าไร ๆ ก็ไม่เสร็จสิ้น เปลี่ยนคนทำไปหลายครั้งหลายหนก็ไม่จบ

มองในแง่ของผู้ว่าจ้าง บางคนเห็นว่าการพัฒนาระบบนั้นประสบความสำเร็จก็ต่อเมื่อได้โปรแกรมที่ทำงานได้ตามที่ตนต้องการ แต่ก็มีบางคนมองเห็นว่าการพัฒนาระบบเสร็จสิ้นเมื่อขีดเส้นปิดโครงการได้

แน่ละครับ ต่างคนต่างก็มองในแง่ของตนเองเป็นหลัก

จริง ๆ แล้วมีปัจจัยอยู่หลายอย่างสำหรับใช้วัดความสำเร็จของการพัฒนาระบบงาน

ปัจจัยที่สำคัญมากที่สุดก็คือ ระบบงานนั้นต้องทำงานได้ตามข้อกำหนดและวัตถุประสงค์ที่ผู้ใช้ต้องการ เรื่องนี้ก็เหมือนกับการไปตัดเสื้อนั่นแหละครับ เมื่อลูกค้าไปตัดเสื้อก็ยอม

ต้องการได้เสียที่มีแบบ มีลักษณะตรงกับที่ตนต้องการ หากต้องการเสียแขนยาวมีปีก แต่คนตัด
เสียไปทำเสียแขนสั้นไม่มีปีกมาให้ ลูกค้าจะพอใจไปได้อย่างไร

ปัจจัยที่สอง การพัฒนาระบบงานนั้นจะต้องใช้งบประมาณและเวลาในการพัฒนาพอ
เหมาะสมคือไม่มากไปกว่าที่ได้คาดคะเนไว้ก่อนจะเริ่มงาน หากคาดคะเนงบประมาณเอาไว้
จำนวนหนึ่ง แต่เวลาทำงานจริงกลับใช้เวลาและทรัพยากรมากกว่าที่กะไว้สองเท่าสามเท่า แม้ว่า
ระบบที่พัฒนาได้จะตรงกับความต้องการ ก็ไม่อาจเรียกว่าประสบความสำเร็จ

ปัจจัยที่สาม การพัฒนาระบบงานนั้นควรจะส่งผลให้ได้ผลงานที่มีการทำงานดีกว่าเดิม
มีประสิทธิภาพดีกว่าเดิม เรื่องนี้เป็นเรื่องที่ยังคงมีผู้ไม่เห็นว่า เพราะมันไปติดอยู่ที่จะต้อง
ได้ระบบงานที่ทำงานได้ตามความต้องการ ซึ่งความต้องการนั้นก็อาจจะไม่แตกต่างไปจากเดิม
นัก เคยทำงานมาอย่างไรก็ต้องการให้การใช้คอมพิวเตอร์มีลักษณะการทำงานเหมือนเดิม ไม่ได้
คิดที่จะใช้คอมพิวเตอร์ในรูปแบบที่จะทำให้การทำงานมีประสิทธิภาพดีขึ้น ดังนั้นระบบงานที่
ประสบความสำเร็จจึงควรช่วยให้การทำงานของผู้ใช้มีประสิทธิภาพดียิ่งขึ้นกว่าเดิม และนี่เป็น
ความท้าทายสำหรับนักคอมพิวเตอร์มืออาชีพ

ปัจจัยที่สี่ ระบบงานที่พัฒนาได้ควรจะทำงานได้ถูกต้องไม่ผิดพลาด นอกจากนั้นควรมี
ความยืดหยุ่น คือสามารถปรับเปลี่ยนการทำงานให้รับการเปลี่ยนแปลงในอนาคตได้บ้าง ทั้งนี้
เพราะโลกของเรานั้นเต็มไปด้วยการเปลี่ยนแปลงอยู่ตลอดเวลา วันนี้ทางการมีกฎเกณฑ์ออกมา
อย่างหนึ่ง แต่ต่อมามีอีกสักพักกฎเกณฑ์เหล่านั้นก็อาจจะเปลี่ยนไปได้ ตัวอย่างที่เห็นได้ง่ายก็คือ
เรื่องของภาษีมูลค่าเพิ่ม เมื่อต้นปี 2540 เราเสียภาษีมูลค่าเพิ่มกันเพียง 7% แต่พอเราเกิด
ปัญหาเศรษฐกิจรุนแรงเราจำเป็นต้องไปยืมเงิน IMF เราก็ต้องเปลี่ยนอัตราภาษีเป็น 10% นั่นหมายถึง
จะต้องแก้ไขโปรแกรมในเครื่องบริการ ณ จุดขายกันเป็นการใหญ่ หากเราไม่สามารถปรับปรุงแก้ไข
โปรแกรมได้ก็ต้องประสบกับความเสียหายมากทีเดียว

เมื่อพิจารณาปัจจัยเหล่านี้แล้ว ก็คงจะพอมองเห็นนะครับว่าการพัฒนาระบบให้ประสบ
ความสำเร็จนั้นไม่ใช่เรื่องง่ายเลย โดยเฉพาะในเมื่อระบบงานคอมพิวเตอร์นั้นมีลักษณะเป็น
นามธรรมค่อนข้างมาก ไม่เหมือนกับการสร้างอาคารที่มีลักษณะเป็นรูปธรรมมาก ก่อนสร้าง
อาคารเรามองออกอยู่แล้วว่า อาคารจะเป็นแบบไหน ใช้วัสดุแบบไหน มีทรวดทรง มีการตกแต่ง
เป็นอย่างไร และในทางปฏิบัติเราก็ทราบราคาของวัสดุต่าง ๆ ที่จะนำมาใช้ค่อนข้างชัดเจน แต่
ระบบงานคอมพิวเตอร์นั้นทั้งคนพัฒนาและคนเป็นเจ้าของมองไม่ออกหรือคิดว่า ระบบจะมี
ลักษณะหน้าตาเป็นอย่างไร ยิ่งเรื่องการใช้งานหรือการปฏิบัติงานของระบบยิ่งยากมากขึ้น
เพราะผู้ใช้งานก็อาจจะไม่คุ้นเคยกับระบบคอมพิวเตอร์มาก่อน

รูปแบบของการพัฒนาระบบ

การพัฒนาระบบงานคอมพิวเตอร์นั้นมีรูปแบบแตกต่างกันหลายอย่าง ก่อนหน้าที่เราจะมาคุยกันถึงลักษณะของการพัฒนาระบบที่ประสบความสำเร็จ เห็นจะต้องทำความเข้าใจเรื่องรูปแบบที่หลากหลายนี้ก่อน

รูปแบบแรกคือ การพัฒนาระบบงานแบบเบ็ดเสร็จ การพัฒนาระบบแบบนี้หมายถึงการมอบหมายให้บริษัทผู้รับจ้าง ดำเนินการทุกอย่างเองหมด ตั้งแต่การจัดหาฮาร์ดแวร์ การติดตั้งฮาร์ดแวร์ การจัดหาและจัดทำซอฟต์แวร์ให้ตรงกับความต้องการ การทดสอบระบบ การฝึกอบรมผู้ใช้ จากนั้นก็ส่งมอบระบบทั้งหมดให้ผู้ใช้ การพัฒนาระบบแบบนี้มีความเสี่ยงสูงมาก เพราะเมื่อเริ่มการพัฒนาระบบงานนั้น รายละเอียดต่าง ๆ ของการทำงานอาจจะยังไม่ชัดเจนมากนัก ความต้องการของทางผู้ใช้อีกก็ไม่ชัดเจนเหมือนกัน แต่การพัฒนาไปกำหนดงบประมาณและระยะเวลาไว้แล้ว ดังนั้นโอกาสที่จะพัฒนาระบบให้ดีได้จึงมีน้อย

รูปแบบที่สอง คือ การพัฒนาเฉพาะส่วนที่เป็นซอฟต์แวร์ คือ เจ้าของงานมีฮาร์ดแวร์อยู่แล้วเป็นส่วนใหญ่ หรือกำหนดไว้ชัดเจนว่าจะใช้ฮาร์ดแวร์อะไร หากจำเป็นจะต้องขยายปรับเปลี่ยนฮาร์ดแวร์บ้างก็เป็นงานส่วนน้อยที่ควบคุมได้ การพัฒนาระบบงานจึงมุ่งที่ซอฟต์แวร์เท่านั้น การพัฒนาระบบงานตามรูปแบบนี้ยังอาจขอย่อยออกไปได้อีก (ความจริงในรูปแบบแรกก็อาจขอย่อยได้ตามแนวนี้เหมือนกันแหละครับ)

ก. การพัฒนาระบบงานโดยใช้ภาษามาตรฐาน เช่น ใช้ภาษาโคบอล ปาสกาล หรือ ภาษาซี หรือภาษาอื่นใดที่ต้องเขียนโปรแกรมอย่างละเอียด

ข. การพัฒนาระบบงานโดยใช้เครื่องมือเช่น CASE Tool หรือ ใช้ภาษารุ่นที่สี่

ง. การพัฒนาระบบงานโดยซื้อซอฟต์แวร์สำเร็จมาใช้ หรือนำมาปรับเปลี่ยนให้ตรงกับความต้องการก่อนใช้งาน

รูปแบบที่สาม คือ การพัฒนาระบบงานส่วนที่เป็นซอฟต์แวร์เหมือนรูปแบบที่สอง แต่ผิดกันคือพัฒนาเพิ่มเติมให้กับซอฟต์แวร์ที่มีอยู่ก่อนแล้ว เช่น เดิมทีเคยมีซอฟต์แวร์สำหรับงานขายสินค้า ก็อาจจะจัดทำซอฟต์แวร์เพิ่มเติมสำหรับจัดทำฐานข้อมูลลูกค้า หรือ ทำซอฟต์แวร์สำหรับเชื่อมโยงซอฟต์แวร์หนึ่ง เข้ากับอีกซอฟต์แวร์หนึ่ง การพัฒนาระบบงานในรูปแบบที่สามนี้มีความจำกัดทางด้านรูปแบบของภาษา คือไม่อาจเลือกภาษาได้ตามความต้องการ แต่ต้องเลือกใช้ให้ตรงกับซอฟต์แวร์ที่มีอยู่เดิม มิฉะนั้นการเชื่อมโยง หรือ การพัฒนาส่วนขยายก็อาจจะไม่ได้ผลดีเท่าที่ควร

รูปแบบที่สี่ คือ การปรับเปลี่ยนซอฟต์แวร์ที่มีอยู่เดิม ให้กลายเป็นรูปแบบใหม่ เช่น การเปลี่ยนซอฟต์แวร์ที่เคยใช้ทำงานในเครื่องเดียว ๆ ให้กลายเป็นซอฟต์แวร์สำหรับทำงานในรูปแบบระบบเครือข่าย หรือ การเปลี่ยนจากภาษาหนึ่งไปเป็นอีกภาษาหนึ่งให้เหมาะสมกับยุคสมัย

มากขึ้น การพัฒนาระบบงานในรูปแบบที่สั้นๆ ก็ไม่น่าจะยาก เพราะมีต้นแบบของเก่าให้พิจารณาอยู่แล้ว แต่กระนั้นก็ยังมีความเสี่ยงอยู่บ้างเหมือนกัน

การพัฒนาระบบงานรูปแบบต่าง ๆ ข้างต้นนี้มีให้พบเสมอ ๆ สุดแท้แต่เป็นงานของหน่วยงานใด ถ้าเข้าใจไม่ผิดการพัฒนาระบบงานของกรมสรรพากร และ กรมศุลกากรนั้นตรงกับรูปแบบแรกซึ่งมีความยุ่งยากมากจนในกรณีของกรมสรรพากรนั้นได้เกิดปัญหาจนขีดขยายดไปทั้งผู้ว่าจ้างและผู้รับจ้าง

ระเบียบวิธีในการพัฒนาระบบงาน

การพัฒนาระบบงานคอมพิวเตอร์นั้นมีระเบียบวิธี (methodology) ให้เลือกใช้ได้หลายแบบด้วยกัน วิธีที่ใช้กันมากก็คือการพัฒนาโดยยึดขั้นตอนตามแนวทางวัฏจักรพัฒนาระบบงาน (System Development Life Cycle หรือ SDLC) ส่วนอีกวิธีหนึ่งก็คือการพัฒนาโดยการทำต้นแบบ (prototyping)

การพัฒนาระบบโดยยึดวัฏจักรพัฒนาระบบงานนั้น ยังอาจแบ่งออกไปได้อีกว่าจะยึดแนวทางการพัฒนาแบบเป็นขั้นเป็นตอนเหมือนน้ำตก ซึ่งเรียกกันทั่วไปว่า Waterfall Model (รูปที่ 1) หรือจะยึดแนวทางพัฒนาแบบเป็นวงขดลวด หรือ Spiral Model (รูปที่ 2)

หากยึดแนวทางแบบ Waterfall Model ก็หมายความว่าผู้พัฒนาจะต้องทำงานในแต่ละขั้นตอนให้เสร็จสิ้นก่อนจะเปลี่ยนไปทำงานในขั้นตอนต่อไป แต่ถ้าหากเปลี่ยนแล้วพบว่าทำต่อไปไม่ได้เพราะงานบางส่วนที่ทำในขั้นตอนที่แล้วนั้นยังได้ผลดีไม่มากเท่าที่ควร ผู้พัฒนาระบบก็อาจจะย้อนกลับไปทำในขั้นตอนเดิมได้อีก ดังนั้นตามรูปที่ 1 นั้น แทนที่จะเห็นลูกศรชี้ลงมาทางเดียวเหมือนน้ำตก กลับเห็นลูกศรย้อนกลับขึ้นไปสู่ขั้นตอนที่ผ่านมาได้ด้วย

ในกรณีของการพัฒนาระบบแบบวงขดลวดนั้น ความจริงแล้วขั้นตอนต่าง ๆ ก็มีลักษณะต่อเนื่องกันไปเหมือนกับการพัฒนาเป็นขั้นตอนแบบน้ำตกนั่นเอง แต่ความแตกต่างอยู่ที่การพัฒนาระบบตามแบบขดลวดนั้น มุ่งที่พยายามพัฒนาระบบแยกเป็นส่วนย่อย ๆ หลาย ๆ ส่วน และพัฒนาระบบย่อยให้เสร็จไปที่ละส่วน เมื่อเสร็จแล้วก็เริ่มมุ่งพัฒนาระบบย่อยอื่น ๆ ต่อไปอีกทีละระบบ ทำเช่นนั้นเรื่อย ๆ ไป จนกระทั่งสำเร็จหมดทั้งระบบ ส่วนการพัฒนาแบบน้ำตกนั้น มุ่งที่จะพยายามทำทั้งระบบให้เสร็จทีละขั้น ๆ

ขั้นตอนในการพัฒนาระบบงานคอมพิวเตอร์นั้นมีผู้เขียนไว้แตกต่างกันหลายแบบ แต่ไม่ว่าจะเขียนแตกต่างกันอย่างไร ในหลักการแล้วก็เหมือนกัน คือมีลักษณะดังแสดงในรูปที่ 3

ตามรูปนี้เราเริ่มด้วยการศึกษาความเหมาะสมของโครงการพัฒนาระบบงานคอมพิวเตอร์ก่อน งานขั้นนี้มีความจำเป็นมากเพราะจะช่วยให้ผู้เป็นต้นคิด ผู้เป็นเจ้าของ และ ผู้ที่จะพัฒนา

ระบบได้รับทราบข้อมูลพื้นฐานว่าระบบที่ต้องการนั้นคืออะไร มีวัตถุประสงค์อย่างกว้าง ๆ อะไรบ้าง ต้องการให้ระบบทำอะไร ควรทำแบบไหน และ ระบบที่คิดทำนั้นจะคุ้มค่าหรือไม่

งานขั้นต่อมาเป็นการศึกษาความต้องการของหน่วยงานที่เป็นเจ้าของระบบ และวิเคราะห์ลักษณะการทำงานของระบบในปัจจุบัน งานขั้นนี้เป็นงานที่มีความสำคัญอย่างยิ่งต่อความสำเร็จของการพัฒนาระบบ หากผู้ดำเนินการพัฒนาไม่เข้าใจความต้องการของหน่วยงาน ซึ่งก็คือทั้งผู้บริหารและผู้ปฏิบัติการอย่างแท้จริงแล้ว ก็เป็นการยากที่จะกำหนดวัตถุประสงค์ของระบบงานได้ถูกต้อง ดังนั้นไม่ว่าจะพัฒนาระบบให้ดีอย่างไรก็ไม่มีวันถูกใจเจ้าของระบบไปได้ เรื่องแบบนี้เรามีคำพังเพยที่รู้จักกันดีว่า **เกาไม่ถูกที่คัน** หรือถ้าหากเป็นฝรั่งก็จะใช้วลีว่า **Solving the wrong problem.**

เมื่อวิเคราะห์ระบบจนเข้าใจการทำงานของระบบปัจจุบัน และเข้าใจความต้องการของหน่วยงานชัดเจนแล้ว นักวิเคราะห์ก็จะมองเห็นปัญหาในระบบปัจจุบันได้ชัดเจนขึ้น และ มองเห็นว่าจะปรับปรุงระบบงานนั้นให้ดีขึ้นได้อย่างไร การปรับปรุงนี้ไม่ใช่เป็นการนำคอมพิวเตอร์มาให้ใช้เป็นเครื่องมือในการทำงานเท่านั้น แต่เป็นการปรับเปลี่ยนการทำงานทั้งหมดที่อาจจะรุ่มร่าม ล้าช้า ผิดพลาด ใช้คนมากเกินไป หรือมีขั้นตอนมากเกินไป เพื่อให้ระบบใหม่มีการทำงานที่ดียิ่ง ขึ้นกว่าเดิม มีประสิทธิภาพและประสิทธิผลมากขึ้น และที่สำคัญคือประหยัดเงินค่าดำเนินการ ลงไปได้อีก

จบจากการวิเคราะห์ระบบแล้วจึงถึงขั้นตอนการออกแบบระบบงาน ซึ่งการออกแบบนี้ก็มี อยู่ด้วยกันสองส่วนใหญ่ ๆ

ส่วนแรกก็คือการออกแบบระบบงานในภาพรวม โดยมุ่งเน้นว่าจะให้ระบบทำงานอะไรได้บ้าง กระแสของงานหรือกิจกรรมในระบบจะเป็นอย่างไร ระบบจำเป็นจะต้องมีฟังก์ชันอะไรบ้าง งานขั้นนี้เปรียบเสมือนกับการที่สถาปนิกลงมือร่างแบบอาคารให้เห็นว่ามีกี่ชั้น มีลักษณะอย่างไร จะเอาประตูหน้าต่างไว้ตรงไหน สวยงามหรือไม่ เราเรียกงานออกแบบขั้นนี้ว่าเป็น **Logical design** ของระบบ

ส่วนที่สองก็คือการออกแบบระบบงานในรายละเอียด โดยกำหนดรายละเอียดที่จำเป็นที่จะทำให้อาคารสร้างระบบได้ เช่นกำหนดลักษณะและรายการข้อมูลทั้งหมดอย่างละเอียด กำหนดรูปแบบของรายงาน หน้าจอ แบบฟอร์มข้อมูล แพ้มข้อมูล ไปจนถึงการออกแบบโปรแกรมว่าจะให้ทำงานอย่างไรด้วย การออกแบบขั้นนี้เปรียบเสมือนกับการที่วิศวกรกำหนดรายละเอียดของอาคารว่าจะต้องมีเหล็กเสริมในเสาในคานกี่เส้น จะต้องตอกเสาเข็มกี่ต้น ฯลฯ เราเรียกงานออกแบบขั้นนี้ว่า **Physical design**

หลังจากออกแบบระบบงานเสร็จแล้วจึงมาถึงขั้นการเขียนโปรแกรม โดยการนำรายละเอียดจากงานออกแบบมาดำเนินการ สร้างให้เป็นโปรแกรมที่ทำงานได้ตามที่ผู้ออกแบบคิดฝัน

เอาไว้ งานชิ้นนี้เป็นงานที่จุกจิก มีรายละเอียดมาก เพราะการออกแบบโปรแกรมและเพิ่มข้อมูลที่กำหนดไว้แต่เดิมนั้นก็ยังคงเป็นแค่เพียงขั้นต้นเท่านั้น ยังคงมีรายละเอียดที่จะต้องเพิ่มเติมอีกมาก หลังจากเขียนโปรแกรมเสร็จแล้ว ผู้เขียนโปรแกรมก็ต้องทดสอบโปรแกรมที่เขียนขึ้นว่าทำงานถูกต้องจริงหรือไม่ การทดสอบโปรแกรมนี้อาจเป็นเรื่องใหญ่เหมือนกัน เพราะถ้าหากทดสอบไม่รอบคอบแล้วก็อาจจะเหลือข้อบกพร่องค้างอยู่ในโปรแกรมอีกมาก ทำให้ผลงานที่คิดว่าทำได้เยี่ยมยอดนั้นกลายเป็นขยะไป

ขั้นต่อมาเป็นการทดสอบระบบ งานชิ้นนี้เป็นงานที่ยุ่งยากมากเหมือนกัน แม้ว่าเราจะทดสอบโปรแกรมในขั้นตอนที่แล้วมาอย่างดีแล้ว แต่โปรแกรมที่ได้ก็อาจจะไม่สามารถทำงานร่วมกันได้ดีนัก หรือ อาจจะไม่เข้ากับระบบที่ออกแบบให้คนทำงานร่วมกับเครื่องคอมพิวเตอร์ หรือ ทำมาแล้วผู้ใช้งานไม่สามารถใช้ได้สะดวก ดังนั้นการทดสอบระบบในขั้นนี้จึงเป็นการทดสอบอย่างกว้างขวาง มีการทดสอบการทำงานของโปรแกรมและของผู้ใช้รวมกันไป มีการทดสอบการสื่อสารระหว่างโปรแกรมที่พัฒนาขึ้นกับโปรแกรมอื่น ๆ มีการทดสอบการถ่ายโอนข้อมูลหรือเพิ่มข้อมูลระหว่างเครื่องคอมพิวเตอร์ในแผนกหนึ่งไปยังอีกแผนกหนึ่ง มีการทดสอบความเข้าใจของผู้ใช้ว่าจะสามารถใช้ระบบนั้นได้ดีเพียงใด ฯลฯ

เมื่อทดสอบระบบจบแล้ว ก็ยังมีขั้นตอนที่จะต้องดำเนินการก่อนที่จะเปลี่ยนเข้ามาสู่การใช้ระบบใหม่อีกหลายขั้นด้วยกัน อาทิการจัดทำคู่มือและเอกสารกำกับโปรแกรม การจัดทำแบบฟอร์มข้อมูล การเปลี่ยนข้อมูลและเพิ่มข้อมูลแบบเดิม ให้อยู่ในรูปแบบที่จะใช้กับระบบใหม่ได้ การฝึกอบรมพนักงานที่จะดูแลการติดตั้งและแก้ไขปัญหาการใช้ระบบงาน และการฝึกอบรมผู้ใช้ให้เข้าใจและใช้ระบบงานนั้นเป็น

ต่อจากนั้นจึงเป็นการเปลี่ยนการปฏิบัติงานมาเป็นการใช้ระบบใหม่ ซึ่งก็อาจจะใช้วิธีการเปลี่ยนแบบทันทีทันใด หรือ เปลี่ยนแบบค่อยเป็นค่อยไป หรือ เปลี่ยนแบบขนาน การเปลี่ยนระบบงานที่กล่าวมานี้มีข้อดีข้อเสียต่างกันซึ่งจะไม่กล่าวถึงในที่นี้

การนำเอาวัฏจักรพัฒนาระบบงานมาเล่าสู่กันฟังโดยย่อในที่นี้ก็เพื่อให้เห็นภาพรวมของการพัฒนาระบบก่อน และเพื่อให้เข้าใจประเด็นที่จะต้องเชื่อมโยงกับขั้นตอนต่าง ๆ ได้ตรงกับที่ ต้องการจะอธิบาย

ประเด็นก่อนการพัฒนาระบบ

การพัฒนาระบบงานคอมพิวเตอร์ในเมืองไทยนั้นมีที่มาได้หลายทาง บางครั้งโครงการพัฒนาระบบก็เกิดขึ้นจากผู้บริหารระดับสูงซึ่งอาจจะมีวิสัยทัศน์กว้างไกล หรือ เคยเห็นงานของผู้อื่นที่ใช้ระบบคอมพิวเตอร์ได้อย่างกว้างขวางครอบคลุมทั้งองค์กร

บางครั้งการพัฒนาระบบก็อาจจะเกิดจากผู้บริหารระดับกลางที่ทนดูความไร้ประสิทธิภาพในการทำงานของหน่วยงานของตนไม่ได้

และบางครั้งการพัฒนาระบบก็อาจจะเกิดจากทางฝ่ายบุคลากรคอมพิวเตอร์เอง ที่ต้องการผลักดันให้มีการแก้ไขปรับปรุงการทำงานขององค์กรให้ดีขึ้น

ไม่ว่าจะเป็นโครงการที่เกิดขึ้นจากความคิดหรือการผลักดันของใคร แต่จุดตั้งต้นจุดแรกนั้นมักจะเป็นประเด็นที่ชัดเจนชีวิตของระบบเอาไว้เรียบร้อยแล้วว่าการพัฒนาระบบนั้นจะสำเร็จหรือไม่

ในกรณีที่ผู้บริหารระดับสูงเป็นผู้สั่งการนั้น หากเป็นหน่วยงานราชการแล้ว การพัฒนาระบบให้สำเร็จมักจะทำได้ยาก ทั้งนี้เพราะผู้บริหารระดับสูงมักจะไม่ค่อยมีโอกาสนั่งทำงานต่อเนื่องได้เป็นเวลานาน ๆ หากเป็นนักการเมืองยุคนี้ก็อาจจะมีเวลานั่งทำงานในกระทรวงใดกระทรวงหนึ่งเพียงประมาณหนึ่งปี เมื่อเข้าทำงานรับผิดชอบ กว่าที่จะจับโครงการคอมพิวเตอร์มาคิด กว่าจะสั่งการก็พอเกิดความผันผวน ต้องมีอันพันตำแหน่ง หรือ ต้องโยกย้ายไปทำงานกระทรวงอื่น ๆ เสียแล้ว หากเป็นข้าราชการประจำ เก้าอี้ก็มีโอกาสร้อนได้คล้าย ๆ กัน เพราะเก้าอี้ที่ผู้บริหารแต่ละคนนั่งนั้น มีผู้จ้องอยากจะทำนั่งแทนตลอดเวลาอยู่แล้ว

ข้อน่าสังเกตก็คือเมื่อผู้บริหารเริ่มผลักดันโครงการขึ้นแล้ว ก็อาจจะมีผู้คอยรับสนองอย่างไม่ลืมหูลืมตาหลายคน เมื่อใครได้รับมอบหมายให้ดำเนินการก็จะพยายามเขียนโครงการให้ทันความต้องการของเจ้านายอย่างเร่งด่วน ตรงนี้แหละที่เริ่มพลาด เพราะแทนที่จะมีการศึกษาความเหมาะสมอย่างถี่ถ้วน และดำเนินการวางแผนอย่างรอบคอบ ผู้รับสนองก็ต้องพยายามผลักดันโครงการให้เกิดเร็วที่สุดโดยไม่ได้พิจารณาความเป็นไปได้หรือไม่

ดังนั้นเราจึงได้ยินเรื่องแปลก ๆ เช่นการจัดซื้อคอมพิวเตอร์ให้แก่โรงเรียนซึ่งยังไม่มีไฟฟ้าใช้บ้าง เป็นโรงเรียนหลังคาจากบ้าง ซื้อคอมพิวเตอร์โดยไม่มีซอฟต์แวร์หรือโปรแกรมใช้บ้าง ไม่มีการฝึกอบรมบ้าง

โครงการเหล่านี้ไม่ต้องบอกก็รู้ว่าเป็นความล้มเหลว และ ทำให้สูญเสียเงินของหน่วยงานไปโดยเปล่าประโยชน์ ส่วนคนที่คิดโครงการหรือคนที่รับสนองโครงการจะได้ประโยชน์บ้างหรือไม่ นั่น ขอให้ไว้ให้คิดเอาเอง

ในกรณีของการผลักดันจากผู้บริหารระดับกลางนั้น ปัญหาที่เป็นไปอีกแนวหนึ่ง คือ มักจะเป็นโครงการที่มุ่งที่หน่วยงานของตนเองเป็นหลักเท่านั้น ไม่สนใจหน่วยงานอื่น หรือ แผนกอื่น ไม่สนใจการที่จะร่วมมือกับหน่วยงานอื่น ดังนั้นโครงการที่กำหนดขึ้นจึงมีลักษณะคับแคบ เปรียบเสมือนกับการสร้างหน่วยงานของตนให้เป็นเกาะที่ไม่มีทางเชื่อมต่อกับหน่วยงานอื่นได้ หรือบางครั้งก็นิยม เรียกรการสร้างระบบงานที่จัดทำฐานข้อมูลเฉพาะของตนเองว่าเป็นการสร้างไซโล (silo) ของตนเองขึ้น ไม่มีทางที่จะแบ่งปันกับผู้อื่น

สำหรับในกรณีของการผลักดันจากทางฝ่ายนักคอมพิวเตอร์เองนั้น ปัญหาที่ไม่เหมือนกับที่กล่าวมาแล้วอีก คือนักคอมพิวเตอร์นั้นสนใจทางด้านเทคโนโลยีค่อนข้างมาก ดังนั้นก็จะพยายามมุ่งสร้างโครงการที่จะให้มีเทคโนโลยีที่ล้ำเลิศ หรือก้าวหน้ามาก ยิ่งได้เป็นคนแรก ๆ ที่ทำได้ดี เพราะจะได้มีชื่อเสียง หรือ จะได้สนุกกับการทำอะไรแปลก ๆ

รวมแล้วจุดเริ่มต้นโครงการเหล่านี้นำไปสู่ความล้มเหลวทั้งนั้น

ทางแก้ไขแรกสุดก็คือ เราจำเป็นต้องวางแผนแม่บท หรือ Master Plan ทางด้านไอที และ ระบบสารสนเทศโดยรวมของหน่วยงานให้ชัดเจนตั้งแต่แรก ทุกหน่วยงานควรจะต้องมีแผนแม่บททางด้านไอที หรือ ทางด้านคอมพิวเตอร์ เพื่อใช้เป็นแนวทางในการขยายงาน หรือ การพัฒนาระบบในอนาคต แผนแม่บทนี้ควรจะชี้ว่าระบบต่าง ๆ ที่หน่วยงานจำเป็นต้องมีนั้นมีอะไรบ้าง เป็นระบบแบบไหน จะเชื่อมโยงกันอย่างไร ใช้มาตรฐานอะไร ใช้ข้อมูลอะไร ใช้รหัสข้อมูลอะไร ใครจะเป็นเจ้าของระบบหรือข้อมูลอะไร นอกจากนั้นยังมองไปถึงเรื่องของอุปกรณ์และการสื่อสารโทรคมนาคมด้วย

เมื่อมีแผนดำเนินการเป็นแม่บทแม่แบบอยู่เช่นนี้ การวางแผนพัฒนาระบบก็เป็นไปได้ง่าย เพราะเรารู้ว่าต้องทำอะไรก่อนอะไรหลัง รู้ว่าระบบที่กำลังจะทำงานนี้เกี่ยวข้องกับระบบใดบ้าง และรู้ว่าจะต้องประสานงานกับผู้อื่นในแบบใด

การจัดทำงบประมาณ หรือ การขอความสนับสนุนจากผู้บริหารก็ทำได้ง่ายขึ้น เพราะสามารถอ้างได้ว่าอยู่ในแผนที่ผ่านการอนุมัติเรียบร้อยแล้ว

ข้างฝ่ายผู้ใช้เองก็สามารถเตรียมตัวเตรียมใจได้ล่วงหน้า ผู้ที่ต้องการใช้ระบบก็จะเข้าใจดีว่าเมื่อใดจะถึงกำหนดเวลาที่จะได้ระบบมาใช้งาน ผู้ที่ไม่อยากใช้ระบบก็จะได้รับทำใจ หรือ รับผิดชอบหาความรู้เพื่อจะได้เตรียมการใช้ระบบเมื่อจัดทำเสร็จแล้ว

การทำแผนแม่บทนั้นเป็นเรื่องต้องเสียเวลามากอย่างแน่นอน แต่ขอยืนยันว่าทำแล้วคุ้ม เพราะเท่ากับเรามีทิศทางการใช้ไอทีในหน่วยงานอย่างแท้จริง ไม่ได้ทำงานแบบลมพัด

การศึกษาความเหมาะสม

แม้ว่าเราจะมีแผนแม่บทกำกับว่าหน่วยงานควรมีระบบอะไรบ้าง และ หน่วยงานควรมีทิศทางการใช้คอมพิวเตอร์ไปทางด้านไหน เมื่อถึงเวลาที่จะหยิบระบบแต่ละเรื่องมาพัฒนาก็ควรจะต้องศึกษาความเหมาะสมของระบบนั้นอีกสักครั้ง ไม่ใช่พอถึงเวลาก็ดำเนินการเลย

เหตุผลก็คือเทคโนโลยีนั้นเปลี่ยนแปลงรวดเร็วมาก ความคิดในเรื่องระบบที่เกิดขึ้นระหว่างการวางแผนนั้นอาจจะใช้ไม่ได้แล้วเมื่อถึงเวลาลงมือ ที่ว่าใช้ไม่ได้นั้นอาจจะไม่ถึงกับสรุปว่าไม่ควรทำระบบ แต่หมายความว่าแนวทางในการพัฒนาระบบที่คิดมาแต่เดิมนั้นอาจจะไม่เหมาะสมแล้วก็ได้ ดังนั้นจึงสมควรใช้เวลาอีกสักเล็กน้อยทบทวนแนวทางที่กำหนดไว้ในแผนงาน

เพื่อดูว่าแนวทางที่วางไว้นั้นดีหรือไม่ เช่นแนวทางที่จะจัดทำระบบเป็นแบบเอกเทศนั้นสมควรที่จะเปลี่ยนเป็นระบบแลนหรือไม่ หรือ ควรเปลี่ยนจากระบบแลนไปใช้ระบบ Network Centric หรือไม่

การศึกษาความเหมาะสมนั้นไม่จำเป็นจะต้องทำให้ใหญ่โตยุ่งยาก เพียงแต่ขอให้ นักวิเคราะห์ระบบที่มีความชำนาญทั้งทางด้านฮาร์ดแวร์ และ ซอฟต์แวร์ เป็นผู้ศึกษาทบทวนแนวทางที่วางไว้ในแผนงาน โดยมุ่งประเด็นไปที่ความเปลี่ยนแปลงขององค์กร ของเทคโนโลยี และ ของสิ่งแวดล้อม ว่าจะมีผลต่อแนวทางเดิมหรือไม่ หากจำเป็นจะต้องเปลี่ยนก็ควรเสนอมารให้เห็นชัด และ ช่วยกำหนดวงเงินงบประมาณใหม่ด้วยว่าจะต้องเปลี่ยนแปลงไปจากเดิมหรือไม่

การวางแผนพัฒนาระบบ

การวางแผนการพัฒนาระบบนั้นเป็นทั้งศาสตร์และศิลป์ ที่สำคัญคือเป็นเรื่องที่ค่อนข้างยากสำหรับคนที่มิประสบการณ่น้อย เพราะมองไม่ออกว่าจะต้องทำอะไรบ้าง ดังนั้นพอได้รับมอบหมายให้ทำงานโครงการก็ลงมือลุยไปเลย ไม่มีการรีรอ หรือ ดูว่าจะต้องทำอะไรบ้าง ใครที่ได้หัวหน้าโครงการลักษณะนี้ก็บอกได้เลยว่า โครงการนั้นล้มเหลวแน่ ๆ ยกเว้นแต่หัวหน้าโครงการจะทำงานแบบ one man show คือทำทุกอย่างเองหมด ดังนั้นก็ไม่จำเป็นจะต้องเขียนแผนออกมาดู เพราะจำทุกอย่างเอาไว้ในหัวได้

แต่อย่าลืมนะครับว่าเวลานี้การพัฒนาซอฟต์แวร์นั้นต้องใช้คนหลายคน ใช้ความรู้และประสบการณ์หลายด้าน ยกตัวอย่างเช่นการพัฒนาระบบปฏิบัติการอย่าง Windows นั้นได้ข่าวว่าต้องใช้กำลังคนเป็นพัน man-year ดังนั้นถ้าหากนั่งทำอยู่คนเดียวจะเสร็จก็เข้าไปศตวรรษที่ 30 โน่น แล้วที่จริงคนทำจะอยู่รอดไปถึงศตวรรษที่ 21 หรือเปล่าก็ยังไม่บอกได้

การวางแผนที่ดีนั้นจำเป็นจะต้องกำหนดทั้งตัวคน งบประมาณ และ ระยะเวลา ตรงนี้ก็คือสิ่งที่ทำให้การวางแผนนั้นยาก เพราะการพัฒนาระบบนั้นมีลักษณะเป็นนามธรรมมาก ดั่งได้เคยกล่าวไปแล้ว ไม่มีใครบอกได้ตั้งแต่แรกว่าจะต้องมีหน้าจอเท่าใด มีกี่รายงาน มีแฟ้มจำนวนเท่าใด แต่ถ้าบอกหรือคาดคะเนไม่ได้ การวางแผนก็ทำไม่ได้ก็อีกเหมือนกัน ดังนั้นนักวิเคราะห์ที่รับทำหน้าที่หัวหน้าโครงการจึงจำเป็นจะต้องคาดคะเนให้ได้ ทางหนึ่งก็คืออาศัยประสบการณ์ อีกทางหนึ่งก็ต้องอาศัยการค้นคว้า และ ความมีหูตากว้างขวาง นั่นก็คือนักวิเคราะห์ระบบ และ หัวหน้าโครงการจำเป็นจะต้องเป็นคนที่ขวนขวายเสาะแสวงหาความรู้ทางด้านระบบงานคอมพิวเตอร์อ่านอยู่เป็นประจำ

การกำหนดปัจจัยต่าง ๆ ทั้งหมดให้พร้อมนั้นเป็นเรื่องยาก บางครั้งเราก็ไม่มีคนที่เก่งมาร่วมทีม หรือบางครั้งการพัฒนาระบบจำเป็นจะต้องใช้คนที่มีความรู้ด้านเทคนิคด้านหนึ่ง แต่เราก็ไม่มีคนที่รู้เรื่องนั้นอยู่ในทีมเลย ดังนั้นหัวหน้าโครงการก็จำเป็นจะต้องหาทางส่งคนที่หมายตาไว้

ไปเรียนรู้เรื่องนั้นด้วย จะปล่อยให้เรียนเองนั้นอย่าหวังว่าจะทำได้ง่าย และอาจจะต้องเสียเวลากับการลองผิดลองถูกอีกมาก

เทคนิคที่ใช้ในการกำหนดเนื้องานที่ใช้กันอยู่เป็นประจำก็คือการทำ Work Break Down นั้นคือการย่อยวาระบบที่กำลังพัฒนาอยู่นั้นประกอบด้วยส่วนประกอบอะไรบ้าง และในการพัฒนาส่วนประกอบเหล่านั้น จะต้องทำอะไรบ้าง

การทำ WBS นี้ต้องอาศัยการฝึกฝนบ่อย ๆ วิธีหนึ่งสำหรับฝึกผู้ที่เป็นหัวหน้าโครงการรุ่นใหม่ก็คือให้ใช้โครงการเดิมที่ทำไปแล้วมาเป็นแบบอย่าง คือให้ทดลองย่อยระบบที่เคยทำแล้ว จากนั้นก็เปรียบเทียบกับของจริง ตรงนี้ต้องอาศัยหัวหน้าโครงการเดิมมาเป็นพี่เลี้ยงหรืออาจารย์สำหรับช่วยอธิบาย

เมื่อทำ WBS แล้ว การวิเคราะห์เวลาและค่าใช้จ่ายก็ง่ายขึ้น หัวหน้าโครงการที่ย่อยโครงการออกมาเป็นเนื้องานได้แล้ว สามารถจับคู่ระหว่างเนื้องานกับบุคลากรในทีมงานได้ โดยดูว่าใครชนิดหรือมีความสามารถทางใด ก็กำหนดให้ทำงานตามเนื้องานนั้น ๆ และถ้าหากหัวหน้าโครงการคุ้นเคยกับนิสัยใจคอของคนในทีมงานดี ก็จะสามารถบอกได้ว่าการทำงานนั้น ๆ จะต้องใช้เวลานานเท่าใดจึงจะเสร็จ

การกำหนดแผนดำเนินการอย่างละเอียดนี้จะต้องใช้เวลามากเหมือนกัน แต่ก็ควรจะทำ เพราะจะทำให้หัวหน้าโครงการทราบว่า งานทั้งหมดจะต้องทำอะไร เมื่อใด และใช้ใครทำ

อย่างไรก็ตามการสร้างแผนเช่นนี้ยังไม่สอดคล้องกับความเป็นจริงประการหนึ่ง เพราะหากยึดตามแนวนี้ เวลาสำหรับทำโครงการก็就会被ยืดออกไปมาก ในขณะที่แต่ละโครงการนั้นก็ถูกบีบให้เสร็จเร็ว ๆ

ดังนั้น หัวหน้าโครงการก็ต้องคำนึงถึงระยะเวลาของโครงการที่ถูกกำหนดมาควบคุมไปด้วย หากพบว่าเนื้องานและกำลังคนไม่เหมาะสม หรือไม่สอดคล้องกับระยะเวลา หัวหน้าโครงการก็มีวิธีการแก้ไขสามแนวทาง

แนวทางแรกคือ เจริญขอยืดเวลาออกไปให้สอดคล้องกับแผนงานที่น่าจะเป็นไปได้มากที่สุด โดยวิธีนี้เราจะสามารถพัฒนาระบบได้ครบถ้วน แต่จะไม่เสร็จทันกำหนดที่วางไว้เดิม

แนวทางที่สอง คือขอเพิ่มบุคลากรในทีมงานเพื่อให้สามารถทำงานส่วนที่ไม่สามารถนำคนในทีมงานเดิมลงมาทำได้

ถ้าหากแนวทางแรกและที่สองไม่ได้รับความเห็นชอบด้วย ก็เหลือแนวทางที่สาม คือจะต้องลดขอบเขตของโครงการลงให้อยู่ในวิสัยที่จะพัฒนาระบบได้ภายในระยะเวลา งบประมาณ และด้วยทีมงานที่มีอยู่

เชื่อไหมครับว่า เรื่องที่กล่าวมาข้างต้นนี้เป็นปัญหาที่ผู้บริหารหลายคนมีความเข้าใจ แต่ไม่ยอมมอหลุ่มอล่วย คือไม่ยอมให้ใช้แนวทางที่อธิบายมานี้เลย คงยืนยันให้ทำตามขอบเขตเดิม และกำหนดเวลาเดิมที่วางไว้

น่าแปลกก็คือ ผู้บริหารเหล่านี้บางคนก็คืออดีตนักคอมพิวเตอร์ซึ่งก็มีความรู้ความเข้าใจ ปัญหาที่ตนเองเคยประสบมาดี แต่พอมาเป็นผู้บริหารก็กลับไม่ยอมเข้าใจ

ผลสุดท้ายก็คือ ความล้มเหลวแบบใดแบบหนึ่ง กล่าวคือถ้าหากพัฒนาระบบไม่ทันตามกำหนด ก็จะทำให้เกิดความเคร่งเครียดขัดแย้ง และ การกระทบกระทั่งกันมากขึ้นระหว่างเจ้าของงานกับผู้พัฒนาระบบ

แต่ผลที่เห็นชัด ๆ ไม่ว่าจะพัฒนาระบบทันหรือไม่ก็คือ ผลงานที่ได้ไม่มีคุณภาพมากเท่าที่ควร หรืออาจจะทำเสร็จแล้วใช้ไม่ได้ด้วยซ้ำไป

เรื่องนี้ยังไม่จบครับ มีแนวคิดอีกหลายเรื่องที่ต้องการนำมาเล่าสู่กันฟัง แต่คงจะต้องเก็บไว้เล่าในตอนต่อไปแล้วละครับ