

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับการวิเคราะห์และออกแบบระบบ

สภาพแวดล้อมของการพัฒนาระบบ

กุญแจสู่ความสำเร็จในการดำเนินธุรกิจคือความสามารถในการรวบรวม การจัดระเบียบ และการตีความข้อมูล การวิเคราะห์และออกแบบระบบเป็นวิธีการพิสูจน์ที่จะช่วยให้ธุรกิจทั้งขนาดใหญ่และขนาดเล็กได้รับผลตอบแทนของการใช้ข้อมูลที่เต็มกำลังตามความสามารถ ในฐานะที่เป็นนักวิเคราะห์ระบบ ซึ่งเป็นบุคคลที่อยู่ในองค์กรที่มีความเกี่ยวข้องอย่างมากกับการวิเคราะห์และออกแบบระบบ และจะสนุกกับอาชีพที่อุดมไปด้วยเส้นทางที่จะช่วยเพิ่มทั้งทักษะด้านคอมพิวเตอร์และทักษะความสัมพันธ์ระหว่างบุคคล

การพัฒนาระบบวงจรชีวิต (Systems Development Life Cycle : SDLC) เป็นศูนย์กลางในการพัฒนาของข้อมูลที่มีประสิทธิภาพ ซึ่งจะเน้นสี่กระบวนการที่สำคัญของการพัฒนาระบบ ที่ใช้ SDLC โดยมีขั้นตอนคือ (1) การวางแผนและการเลือกระบบ (2) การวิเคราะห์ระบบ (3) การออกแบบระบบ และ (4) การนำระบบไปใช้และการดำเนินงาน โดยจะคำนึงถึงว่าขั้นตอนเหล่านี้อาจจะแตกต่างกันในแต่ละองค์กรขึ้นอยู่กับเป้าหมายขององค์กรนั้น และการพัฒนาระบบด้วย SDLC

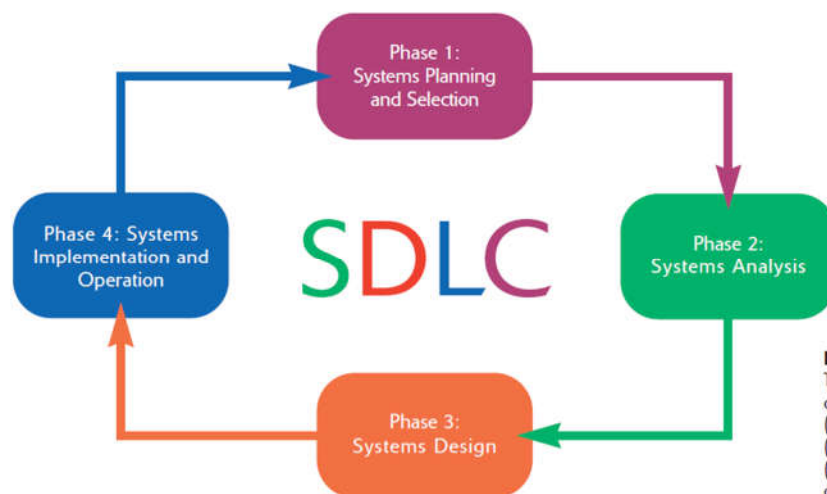


FIGURE 1-1
The four steps of the systems development life cycle (SDLC): (1) planning and selection, (2) analysis, (3) design, and (4) implementation and operation.

การวิเคราะห์และออกแบบระบบสารสนเทศ

การวิเคราะห์และการออกแบบระบบข้อมูลเป็นวิธีการที่ใช้โดย บริษัท IBM, Pepsi Co และ Sony เพื่อสร้างและบำรุงรักษาระบบสารสนเทศที่ดำเนินการทำงานทางธุรกิจขั้นพื้นฐาน เช่น การติดตามของซื้อลูกค้าและที่อยู่ คำสั่งประมวลผล และการจ่ายเงินพนักงาน เป้าหมายหลักของการวิเคราะห์และการออกแบบระบบคือการ

ปรับปรุงระบบขององค์กร โดยทั่วไปจะดำเนินการใช้ซอฟต์แวร์ที่สามารถช่วยให้พนักงานประสบความสำเร็จทางธุรกิจที่สำคัญ โดยจะทำงานงานได้ง่ายขึ้นและมีประสิทธิภาพ ในฐานะที่เป็นนักวิเคราะห์ระบบ จะเป็นศูนย์กลางในการพัฒนาซอฟต์แวร์นี้ การวิเคราะห์และออกแบบระบบสารสนเทศจะขึ้นอยู่กับ

1. ความเข้าใจเป้าหมายขององค์กร โครงสร้าง และกระบวนการ
2. ความรู้ที่เกี่ยวกับวิธีการใช้ประโยชน์จากเทคโนโลยีสารสนเทศเพื่อความได้เปรียบทางธุรกิจ

เพื่อที่จะประสบความสำเร็จในการดำเนินการนี้ ควรที่จะปฏิบัติตามวิธีการที่มีโครงสร้าง SDLC ที่ได้แสดงในรูปที่ 1-1 ซึ่งเป็นแนวทางทั้งสี่กระบวนการ ที่จะสามารถดำเนินการเพื่อระบุการวิเคราะห์ การออกแบบ และการนำไปใช้ของระบบสารสนเทศ

หลักการการวิเคราะห์และออกแบบระบบ

เป้าหมายหลักของการวิเคราะห์และการออกแบบระบบคือการปรับปรุงระบบองค์กร บ่อยครั้งที่กระบวนการนี้เกี่ยวข้องกับการพัฒนาหรือการซื้อซอฟต์แวร์ และการฝึกอบรมพนักงานที่จะใช้ซอฟต์แวร์นั้น ซอฟต์แวร์ซึ่งจะเรียกได้อีกอย่างว่าระบบ ที่ถูกออกแบบมาเพื่อรองรับการทำงานขององค์กรที่เฉพาะเจาะจงหรือกระบวนการ เช่นการจัดการสินค้าคงคลัง การทำบัญชีเงินเดือน หรือการวิเคราะห์ตลาด เป้าหมายของซอฟต์แวร์คือนำข้อมูลเป็นสารสนเทศ ยกตัวอย่างเช่นการพัฒนาซอฟต์แวร์สำหรับแผนกสินค้าคงคลังที่ร้านหนังสือที่อาจจะติดตามจำนวนหนังสือในสต็อกของที่ขายดีที่สุดล่าสุด ซอฟต์แวร์สำหรับแผนกการเงิน อาจติดตามการเปลี่ยนแปลงอัตราค่าจ้างของพนักงาน ความหลากหลายของออกจากชั้นวางของซอฟต์แวร์ประยุกต์สามารถซื้อได้ ซึ่งรวมถึง WordPerfect, Excel และ PowerPoint แต่ซอฟต์แวร์นั้นอาจจะไม่เหมาะสมกับความต้องการที่เฉพาะเจาะจงขององค์กร และเพื่อให้องค์กรที่จะต้องพัฒนาซอฟต์แวร์ขึ้นมาเอง

นอกเหนือจากซอฟต์แวร์ประยุกต์ ระบบสารสนเทศรวมถึง

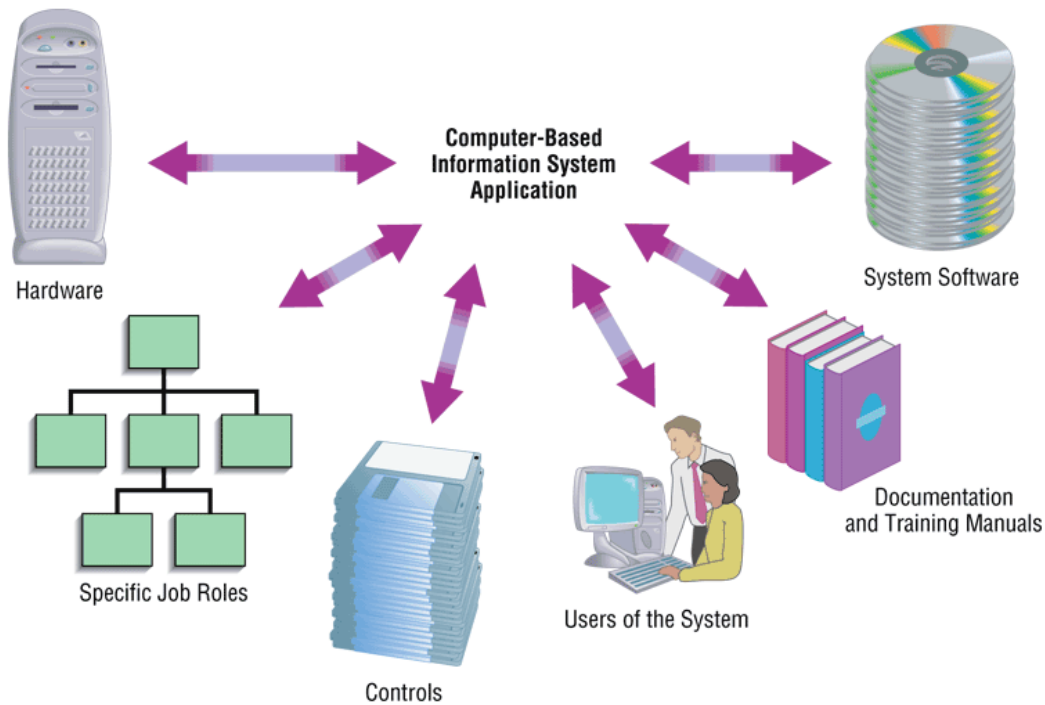
1. ฮาร์ดแวร์และซอฟต์แวร์ระบบที่ทำให้ซอฟต์แวร์ประยุกต์ทำงานได้ โดยซอฟต์แวร์ระบบจะช่วยให้การทำงานของเครื่องคอมพิวเตอร์ ในขณะที่ซอฟต์แวร์ประยุกต์ใช้ช่วยผู้ใช้ดำเนินการทำงาน เช่น เขียนเอกสารเตรียมกระดาษคำนวณ และการเชื่อมโยงกับอินเทอร์เน็ต
2. เอกสารและการฝึกอบรมซึ่งเป็นวัสดุที่สร้างขึ้นโดยนักวิเคราะห์ระบบเพื่อช่วยพนักงานในการใช้ซอฟต์แวร์ที่พัฒนาโดยนักวิเคราะห์ระบบ
3. บทบาทงานเฉพาะที่เกี่ยวข้องกับระบบโดยรวม เช่นคนที่ใช้คอมพิวเตอร์และซอฟต์แวร์ใช้ในการดำเนินงาน
4. การควบคุมซึ่งเป็นส่วนของซอฟต์แวร์ที่พัฒนาขึ้นเพื่อช่วยป้องกันการทุจริตและการโจรกรรม

5. คนที่ใช้ซอฟต์แวร์ในการดำเนินงานต่าง ๆ

องค์ประกอบของระบบสารสนเทศคอมพิวเตอร์ที่ใช้โปรแกรมประยุกต์ที่มีสรุปในรูปที่ 1-2 ซึ่งจะมุ่งเน้นไปทุกมิติของระบบโดยรวมที่มีความสำคัญโดยเฉพาะอย่างยิ่งในการพัฒนาซอฟต์แวร์ประยุกต์ ซึ่งเป็นหน้าที่ความรับผิดชอบในฐานะที่เป็นนักวิเคราะห์ระบบ

เป้าหมายก็คือเพื่อช่วยให้เข้าใจและปฏิบัติตามกระบวนการทางวิศวกรรมซอฟต์แวร์ที่นำไปสู่การสร้างระบบสารสนเทศ ดังแสดงในรูปที่ 1-3 โดยการพิสูจน์แล้วว่าวิธีการ เทคนิค และเครื่องมือ ทั้งสามกระบวนการนี้จะเป็นศูนย์กลางในดำเนินการทางกระบวนการทางวิศวกรรมซอฟต์แวร์

Figure 1.2 Components of a Computer-Based Information System Application



กระบวนการวิศวกรรมซอฟต์แวร์

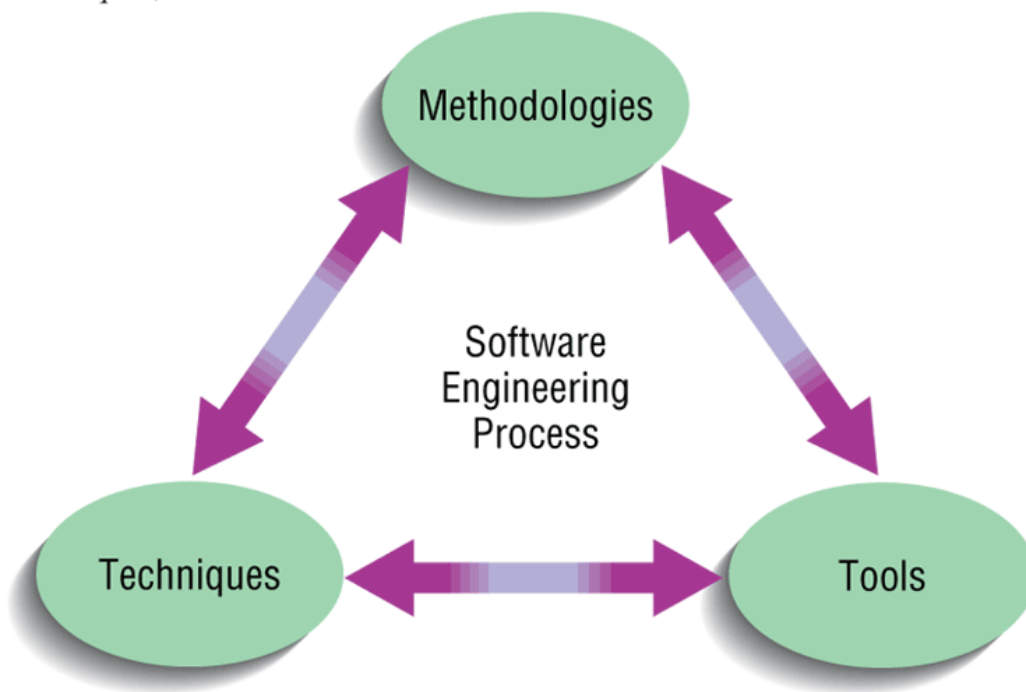
วิธีการเป็นลำดับของวิธีการขั้นตอน โดยจะเป็นขั้นตอนที่จะช่วยพัฒนาผลิตภัณฑ์ในขั้นสุดท้ายของระบบสารสนเทศวิธีการรวมส่วนใหญ่เทคนิคการพัฒนาหลายประการ เช่นการสังเกตและการสัมภาษณ์โดยตรงกับผู้ใช้ของระบบปัจจุบัน

เทคนิคนั้นเป็นกระบวนการ ในฐานะที่เป็นนักวิเคราะห์จะปฏิบัติตามเพื่อช่วยให้ที่ทำงานที่ดีที่สมบูรณ์และเข้าใจกับคนอื่น ๆ ในทีมงานโครงการ เทคนิคการให้การสนับสนุนสำหรับหลากหลายของงานรวมทั้งการดำเนินการ

สัมพันธ์กับผู้ใช้ในปัจจุบันและในอนาคตของระบบสารสนเทศ เพื่อการตรวจสอบระบบที่จะดำเนินการอย่างไร การวางแผน และการจัดการกิจกรรมในโครงการพัฒนาระบบ วัตถุประสงค์สำหรับวิธีการทำงานของระบบ และการออกแบบรายงาน เช่นใบแจ้งหนี้ ระบบจะสร้างสำหรับผู้ใช้ในการดำเนินงาน

เครื่องมือสำหรับโปรแกรมคอมพิวเตอร์ เช่นเครื่องมือวิศวกรรมซอฟต์แวร์ใช้คอมพิวเตอร์ช่วย (CASE) ซึ่งจะเป็นเครื่องมือที่ใช้งานง่ายและมีการใช้เทคนิคเฉพาะ ประกอบด้วยสามองค์ประกอบ นั่นคือ วิธีการ เทคนิค และ เครื่องมือ ซึ่งเป็นการทำงานร่วมกันในรูปแบบขององค์กร โดยจะใช้แนวทางในการวิเคราะห์ระบบและการออกแบบ

Figure 1.3 The software engineering process uses methodologies, techniques, and tools.



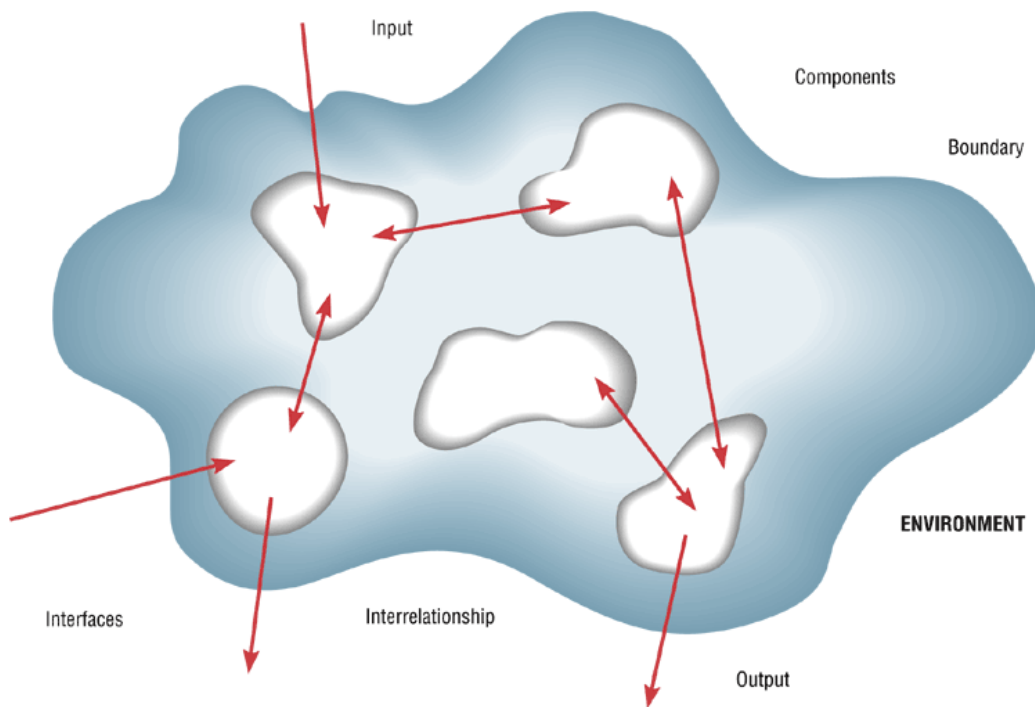
ระบบ

ระบบคือชุดความสัมพันธ์ของการดำเนินงานทางธุรกิจ (หรือองค์ประกอบทางธุรกิจ) ที่ใช้ในหน่วยงานทางธุรกิจ และที่ทำงานร่วมกันเพื่อจุดประสงค์บางอย่าง ตัวอย่างเช่น ระบบในแผนกการเงินจะติดตามการชำระเงิน ในขณะที่ระบบสินค้าคงคลังจะติดตามสินค้า การทำงานของทั้งสองระบบจะแยกจากกัน ซึ่งระบบจะมีเก๋าลักษณะด้วยกัน โดยเจ็ดลักษณะแสดงในรูปที่ 1-4 และได้คำอธิบายรายละเอียดของแต่ละลักษณะดังต่อไปนี้ แต่จากรูปจะเห็นว่าระบบที่มีอยู่ภายในโลกที่มีขนาดใหญ่สภาพแวดล้อม ขอบเขต (boundary) แยกระบบจากสภาพแวดล้อม

(environment) ระบบจะรับการนำเข้า (input) มาจากภายนอก ดำเนินการทำ (process) และส่งผลลัพธ์ (output) กลับไปยังสภาพแวดล้อม ลูกศรในรูปที่แสดงให้เห็นการทำงานร่วมกันระหว่างระบบและโลกภายนอก

1. ส่วนประกอบ (Components)
2. ส่วนประกอบสัมพันธ์ (Interrelated Components)
3. ขอบเขต (Boundary)
4. วัตถุประสงค์ (Purpose)
5. สภาพแวดล้อม (Environment)
6. ส่วนต่อประสาน (Interfaces)
7. การจำกัด (Constraints)
8. ข้อมูลเข้า (Input)
9. ผลลัพธ์ (Output)

Figure 1.4 Seven Characteristics of a System



วิธีการที่ทันสมัยในการวิเคราะห์และออกแบบระบบ

ในปัจจุบันนี้ การพัฒนาระบบมุ่งเน้นไปที่การบูรณาการระบบ ซึ่งการบูรณาการระบบช่วยให้ฮาร์ดแวร์และซอฟต์แวร์จากผู้ขายที่แตกต่างกันจะทำงานร่วมกันโปรแกรมประยุกต์ได้ นอกจากนี้ยังช่วยพัฒนาระบบที่มีอยู่ใน

ภาษากระบวนการจะทำงานร่วมกับระบบใหม่ที่สร้างขึ้นกับสภาพแวดล้อมการเขียนโปรแกรมด้วยภาพ นักพัฒนาใช้สภาพแวดล้อมการเขียนโปรแกรมด้วยภาพเช่น โปรแกรม Visual Basic ในการออกแบบส่วนติดต่อผู้ใช้สำหรับระบบที่ทำงานบนแพลตฟอร์มรับ-ให้บริการ ในสภาพแวดล้อมของรับ-ให้บริการ ซอฟต์แวร์บางส่วนทำงานบนเครื่องบริการ คอมพิวเตอร์ที่มีความสามารถสูงที่ออกแบบมาเพื่อช่วยให้ผู้คนจำนวนมากเข้าถึงซอฟต์แวร์และข้อมูลที่เก็บไว้ในเครื่องบริการ และซอฟต์แวร์บางส่วนที่ทำงานบนเครื่องผู้รับบริการ เครื่องผู้รับบริการเป็นเครื่องคอมพิวเตอร์ที่ให้บนโต๊ะทำงาน ฐานข้อมูลมักจะมีอยู่ในเครื่องให้บริการ อินเทอร์เน็ตจะบริหารจัดการในรูปแบบของการรับ-ให้บริการ ด้วยโปรแกรมคั่นดูบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่บ้าน ซึ่งจะสามารถได้รับเพิ่มข้อมูลและโปรแกรมประยุกต์จากคอมพิวเตอร์ที่แตกต่างกันทั่วโลก เครื่องคอมพิวเตอร์ส่วนบุคคลที่บ้านเป็นเครื่องรับบริการ และคอมพิวเตอร์ทั้งหมดที่ใช้อินเทอร์เน็ตเป็นเครื่องให้บริการ อีกทางเลือกหนึ่งสำหรับองค์กร อาจซื้อระบบทั่วทั้งองค์กรทั้งจากบริษัท เช่น SAP (ระบบ โปรแกรมประยุกต์ และผลิตภัณฑ์ในการประมวลผลข้อมูล) หรือ Oracle ระบบทั่วทั้งองค์กรจะมีขนาดใหญ่ ระบบที่ซับซ้อนที่ประกอบด้วยของชุดของโมดูลระบบที่เป็นอิสระ นักพัฒนาจะเอาระบบมารวมกัน โดยการเลือกและการนำโมดูลที่เฉพาะเจาะจงไปใช้ ระบบทั่วทั้งองค์กรมักจะมีซอฟต์แวร์ที่ให้การสนับสนุนงานที่แตกต่างกันในองค์กรมากกว่าการใช้ในงานเดียวหรือสองงาน ยกตัวอย่างเช่นระบบทั่วทั้งองค์กรสามารถจัดการงานบุคคล งานจ่ายเงินเดือน งานสิทธิและผลประโยชน์ และงานเกษียณอายุภายในระบบเดียวที่เป็นแบบบูรณาการได้ ในความเป็นจริงนั้น จะหายากมากสำหรับองค์กรที่จะใช้ในการพัฒนาระบบภายในองค์กรอีกต่อไป

บทบาทของคุณในการพัฒนาระบบ

แม้ว่าจะมีหลายคนที่อยู่ในองค์กรที่มีส่วนร่วมในการวิเคราะห์และการออกแบบระบบ นักวิเคราะห์ระบบมีความรับผิดชอบหลักในส่วนนี้ อาชีพเป็นระบบนักวิเคราะห์จะมีผลกระทบที่สำคัญเกี่ยวกับการดำเนินงานในองค์กร งานในตำแหน่งนี้เติบโตอย่างรวดเร็วและคุ้มค่าที่พบในทั้งบริษัทขนาดใหญ่และขนาดเล็ก IDC กลุ่มที่ปรึกษาชั้นนำคาดการณ์การเจริญเติบโตที่ในการจ้างงานด้านเทคโนโลยีสารสนเทศ (IT) จะเกินร้อยละ 3 ต่อปีตั้งแต่ปี 2013 โดยสำนักงานสถิติแรงงานได้คาดการณ์การเพิ่มขึ้นในงานด้าน IT จากปี 2004 - 2014 ซึ่งในช่วงเวลานี้แรงงานอาชีพ IT ได้เพิ่มงานใหม่จำนวน 1 ล้านงานในประเทศสหรัฐอเมริกา ผู้ทำงานด้าน IT ยังคงอยู่ในความต้องการ

บทบาทหลักของนักวิเคราะห์ระบบคือการศึกษาปัญหาและความต้องการขององค์กรเพื่อตรวจสอบ คนวิธีการ และเทคโนโลยีสารสนเทศเป็นสิ่งที่ดีที่สุดที่จะสามารถรวมกันเพื่อนำมาปรับปรุงในองค์กร นักวิเคราะห์ระบบจะช่วยให้ผู้ใช้ระบบและผู้จัดการธุรกิจอื่น ๆ ที่กำหนดต้องการการให้บริการให้หรือการเพิ่มประสิทธิภาพ

ข้อมูล นักวิเคราะห์ระบบเป็นกุญแจสำคัญในกระบวนการพัฒนาระบบ เพื่อที่จะประสบความสำเร็จในการเป็นนักวิเคราะห์ระบบ จะต้องมีการพัฒนาทักษะสี่ประเภท ได้แก่ การวิเคราะห์ เทคนิค การบริหารจัดการ และความสัมพันธ์ระหว่างบุคคล ทักษะในการวิเคราะห์ช่วยให้เข้าใจองค์กรและหน้าที่การทำงานในองค์กร ในการที่จะหาโอกาสและปัญหา และการวิเคราะห์และแก้ปัญหา หนึ่งในสิ่งที่สำคัญที่สุดในทักษะการวิเคราะห์ที่สามารถพัฒนา คือ การคิดเชิงระบบหรือสามารถในการมององค์กร และระบบสารสนเทศให้เป็นระบบ ความคิดเชิงระบบให้กรอบที่จะเห็นความสัมพันธ์ที่สำคัญในระบบสารสนเทศ องค์กรที่มีอยู่แล้วและสภาพแวดล้อมที่องค์กรมีอยู่ ทักษะทางเทคนิคที่ช่วยให้เข้าใจศักยภาพและข้อจำกัดของเทคโนโลยีสารสนเทศ ในฐานะที่เป็นนักวิเคราะห์จะต้องสามารถมองเห็นระบบสารสนเทศที่จะช่วยให้ผู้ใช้แก้ปัญหาที่เกิดขึ้นและที่จะเป็นแนวทางในการออกแบบและการพัฒนาระบบ และนอกจากนี้จะต้องสามารถที่จะทำงานร่วมกับภาษาการเขียนโปรแกรม เช่น ภาษา C และ ภาษา Java ระบบปฏิบัติการต่าง ๆ เช่น Windows และ Linux และแพลตฟอร์มฮาร์ดแวร์คอมพิวเตอร์เช่น IBM และ Mac ทักษะการจัดการช่วยในการจัดการโครงการ การจัดการทรัพยากร การจัดการความเสี่ยง และการจัดการความเปลี่ยนแปลง ทักษะความสัมพันธ์ระหว่างบุคคลช่วยให้ทำงานกับผู้ใช้งานได้เช่นเดียวกับนักวิเคราะห์และนักเขียนโปรแกรมอื่น ๆ ในฐานะที่เป็นนักวิเคราะห์ระบบจะมีบทบาทสำคัญเป็นผู้ประสานงานระหว่างผู้ใช้นักเขียนโปรแกรม และผู้เชี่ยวชาญด้านระบบอื่น ๆ การสื่อสารด้วยการเขียนและการพูดที่มีประสิทธิภาพรวมทั้งความสามารถในการประชุม การสัมภาษณ์ผู้ใช้ และการฟังเป็นทักษะที่สำคัญที่นักวิเคราะห์จะต้องมี นักวิเคราะห์ที่มีประสิทธิภาพจะประสบความสำเร็จได้นั้น จะต้องมิตักษะเหล่านี้ทั้งสี่ประเภทดังรูปที่ 1-8 (จะเป็นการโฆษณาทั่วไปสำหรับตำแหน่งนักวิเคราะห์ระบบ

ลองพิจารณาตัวอย่างของประเภทของปัญหาองค์กรที่อาจต้องเผชิญในฐานะที่เป็นนักวิเคราะห์ระบบ เริ่มจากการทำงานในแผนกระบบสารสนเทศของบริษัทนิตยสาร บริษัทจะมีปัญหาในการติดตามการปรับปรุงและรายชื่อที่ถูกต้องของสมาชิก และลูกค้าบางส่วนจะได้รับนิตยสารสองครั้งแทนที่จะได้หนึ่งครั้ง บริษัทนี้จะสูญเสียเงินและสมาชิกถ้าปัญหาเหล่านี้ยังคงเกิดขึ้นอยู่ เพื่อที่จะทำให้ระบบการติดตามมีประสิทธิภาพมากขึ้น ผู้ใช้ของระบบคอมพิวเตอร์ในปัจจุบัน และในขณะเดียวกันผู้จัดการฝ่ายการเงินส่งปัญหาไปให้กับนักวิเคราะห์ระบบและเพื่อนร่วมงานในแผนกระบบสารสนเทศ และประการที่สองการทำงานในแผนกระบบสารสนเทศที่มหาวิทยาลัยที่ซึ่งจะเรียกร้องไปยังองค์กรตามที่อยู่ของปัญหาองค์กร เช่นการส่งจดหมายทางไปรษณีย์สำหรับเกรดของนักเรียนโดยมีการจำหน่ายของที่อยู่ที่ไม่ถูกต้อง

เมื่อมีการพัฒนาระบบสารสนเทศในการจัดการกับปัญหาที่เกิดขึ้นเหล่านี้แล้ว องค์กรและนักวิเคราะห์ระบบมีหลายตัวเลือก โดยสามารถไปที่บริษัทให้บริการเทคโนโลยีสารสนเทศ เช่น Accenture หรือ EDS บริษัท HP มีระบบที่พัฒนาขึ้นสำหรับองค์กร สามารถซื้อระบบที่มีอยู่แล้ว และสามารถนำไปใช้กับระบบขององค์กรได้ทั้ง

องค์กร เช่น SAP สามารถใช้ซอฟต์แวร์ที่เป็นโอเพนซอร์ส หรือให้พนักงานในบริษัททำการพัฒนาระบบขึ้นมาใช้ หรืออีกทางเลือกหนึ่งโดยองค์กรอาจจะตัดสินใจที่จะใช้การพัฒนาและการดำเนินการระบบจากบริษัทภายนอก

ประเภทของระบบสารสนเทศและการพัฒนาระบบ

1. การทำธุรกรรมระบบการประมวลผล (TPS) ทำให้การจัดการข้อมูลเกี่ยวกับกิจกรรมทางธุรกิจ (การทำรายการ)
2. ระบบสารสนเทศเพื่อการจัดการ (MIS) แปลงข้อมูลดิบจากระบบการประมวลผลการทำธุรกรรมในรูปแบบที่มีความหมาย
3. ระบบสนับสนุนการตัดสินใจ (DSS) ออกแบบมาเพื่อช่วยให้ผู้มีอำนาจตัดสินใจ ให้มีสภาพแวดล้อมแบบโต้ตอบสำหรับการตัดสินใจ

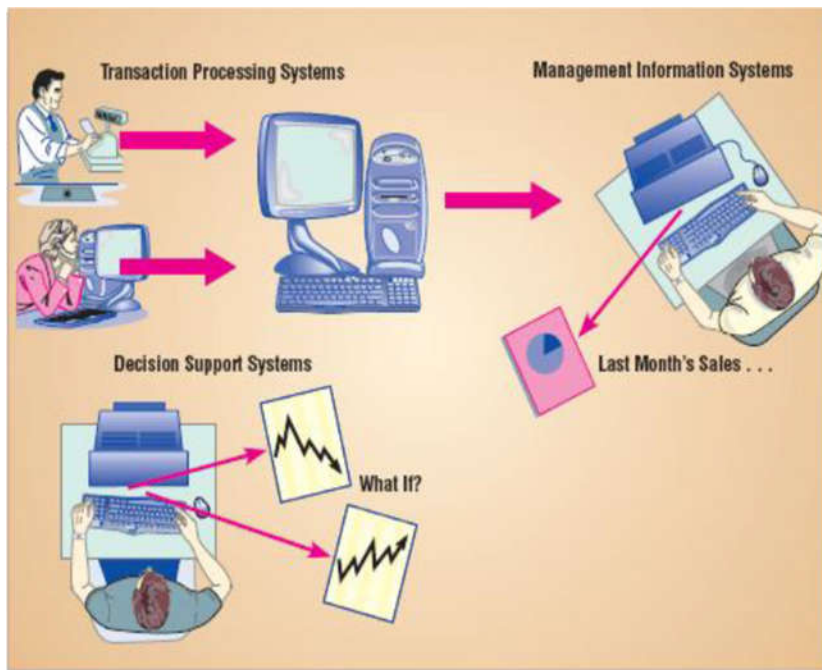


FIGURE 1-9
Depictions of three classes of information systems: TPS, MIS, and DSS.

การพัฒนาระบบสารสนเทศและวงจรชีวิตการพัฒนาระบบ

องค์กรจะใช้ขั้นตอนมาตรฐานที่เรียกว่าวิธีการพัฒนาระบบ เพื่อที่จะพัฒนาและสนับสนุนระบบสารสนเทศ เช่นเดียวกับหลายกระบวนการ การพัฒนาระบบสารสนเทศมักจะเป็นไปตามวงจรชีวิต ตัวอย่างเช่น ผลิตภัณฑ์ในเชิงพาณิชย์ เช่นรองเท้า Nike หรือรถ Honda ได้ทำตามวงจรชีวิตดังนี้ เริ่มจากการพัฒนา ทดสอบ และนำเข้าสู่ตลาด มีการเพิ่มยอดขาย จนถึงที่สุด และลดลง ในท้ายที่สุดผลิตภัณฑ์ก็จะถูกเอาออกจากตลาดและจะถูกแทนที่ด้วยสิ่งอื่น วงจรชีวิตการพัฒนาระบบ (SDLC) เป็นวิธีการทั่วไปสำหรับการพัฒนาระบบในหลายองค์กร นับเป็น

ขั้นตอนของการพัฒนาระบบสารสนเทศ บางคนมีความคิดสำหรับระบบสารสนเทศและสิ่งที่จะต้องทำอย่างไร องค์กรที่จะใช้ระบบในการตัดสินใจที่จะทุ่มเทพยายามที่จำเป็นในการแสวงหา การศึกษาอย่างรอบคอบได้ ดำเนินการตามวิธีของระบบที่จะสนับสนุน ผู้เชี่ยวชาญได้พัฒนากลยุทธ์สำหรับการออกแบบระบบใหม่ซึ่งจะเป็นไปได้ทั้งการสร้างหรือการซื้อ เมื่อเสร็จแล้วระบบจะติดตั้งในองค์กร และหลังจากการฝึกอบรมที่เรียบร้อยแล้ว ผู้ใช้ เริ่มต้นในการทำงานระบบใหม่ในการทำงานในชีวิตประจำวัน ทุกองค์กรใช้รูปแบบวงจรชีวิตที่แตกต่างกันเล็กน้อย ขั้นตอนเหล่านี้มีที่ใดก็ได้จากสามถึงเกือบสี่ขั้นตอน ในที่นี้เราเน้น SDLC เป็นสี่ขั้นตอนคือ (1) การวางแผนและการเลือก (2) การวิเคราะห์ (3) การออกแบบ และ (4) การนำไปใช้

แม้ว่าวงจรชีวิตใด ๆ ที่ปรากฏขึ้นในครั้งแรกที่จะเรียงตามลำดับขั้นตอน แต่ก็เป็นไปได้ ขั้นตอนที่ระบุ และลำดับจะหมายถึงการนำไปปรับใช้ตามความจำเป็นสำหรับโครงการ ยกตัวอย่างเช่น ในขั้นตอนของ SDLC ใดก็ตาม โครงการสามารถกลับไปขั้นตอนก่อนหน้าในกรณีที่เป็น ในทำนองเดียวกันถ้าในเชิงพาณิชย์สินค้าที่ขายไม่ได้ดีหลังจากการเปิดตัว ก็อาจจะเป็นชั่วคราวที่จะออกจากตลาดและปรับปรุงให้ดีขึ้นก่อนที่จะเปิดตัวใหม่อีกครั้ง ในวงจรชีวิตการพัฒนาระบบ ก็ยังเป็นไปได้ที่จะทำกิจกรรมบางอย่างในขั้นตอนหนึ่งในแบบคู่ขนานกับกิจกรรม บางส่วนของขั้นตอนอื่น บางครั้งวงจรชีวิตเป็นการทำซ้ำ ว่ามีขั้นตอนมีการทำซ้ำตามที่ต้องการจนได้รับการยอมรับจากระบบ นักวิเคราะห์บางคนคิดว่าวงจรชีวิตระบบที่จะเป็นเกลียวที่ทำอย่างต่อเนื่องผ่านขั้นตอนในระดับที่แตกต่างกันของรายละเอียดดังแสดงในรูปที่ 1-10 ธรรมชาติของวงกลมในไดอะแกรมของรูปที่ 1-10 แสดงจุดจบของชีวิตในระบบนำไปสู่การเริ่มใหม่ของโครงการอื่น ๆ ซึ่งจะแทนที่ระบบที่มีอยู่ อย่างไรก็ตามการเข้าใจ วงจรชีวิต การพัฒนาระบบได้ใช้ในองค์กรจะเป็นลำดับของกิจกรรมที่ถูกทำและถูกวางแผนเอาไว้สำหรับแต่ละโครงการที่จะพัฒนา ทักษะที่ต้องการของนักวิเคราะห์ระบบจะประยุกต์ใช้ในโมเดลวงจรชีวิตทั้งหมด

ขั้นตอนของวงจรชีวิตการพัฒนาระบบ

1. การวางแผนระบบและการคัดเลือก

ขั้นตอนแรกในระบบ SDLC ระบบการวางแผนและการเลือกมีสองกิจกรรมหลัก กิจกรรมแรกระบุถึงความต้องการระบบใหม่หรือปรับปรุงระบบ ความต้องการสารสนเทศขององค์กรที่มีการตรวจสอบ และโครงการที่จะตอบสนองความต้องการเหล่านี้จะมีการระบุ ระบบสารสนเทศขององค์กรมีความต้องการที่อาจส่งผลมาจาก

- มีการร้องขอที่จะจัดการกับปัญหาในกระบวนการปัจจุบัน
- มีความต้องการที่จะดำเนินการทำงานเพิ่มเติม
- มีความเข้าใจในเทคโนโลยีสารสนเทศสามารถนำไปใช้ประโยชน์ในโอกาสที่มีอยู่

นักวิเคราะห์ระบบจัดลำดับและแปลความต้องการเป็นแผนที่เขียนสำหรับระบบสารสนเทศ (IS) แผนกรวมทั้งระยะเวลาในการพัฒนาระบบใหม่ที่สำคัญ ร้องขอสำหรับระบบใหม่ได้จากผู้ใช้ที่มีต้องการใหม่หรือระบบที่มีความต้องการเพิ่มขึ้น ในระหว่างขั้นตอนการวางแผนและการคัดเลือกระบบ องค์กรจะกำหนดว่าทรัพยากรที่ควรทุ่มเทให้กับการพัฒนา หรือการเพิ่มประสิทธิภาพของแต่ละระบบสารสนเทศจะอยู่ภายใต้การพิจารณา

การศึกษาความเป็นไปก่อนที่จะดำเนินการระยะที่สองของ SDLC เพื่อตรวจสอบผลกระทบทางเศรษฐกิจและองค์กรที่มีต่อระบบ

2. การวิเคราะห์ระบบ

ระยะที่สองของวงจรชีวิตการพัฒนาระบบคือการวิเคราะห์ระบบ ช่วงนี้นักวิเคราะห์ศึกษาขั้นตอนในปัจจุบันขององค์กรและระบบสารสนเทศอย่างละเอียดการใช้ในการดำเนินการทั่วไป เช่น บัญชีแยกประเภท การจัดส่ง รายการสั่งซื้อ การตั้งเวลาเครื่อง และเงินเดือน การวิเคราะห์มีหลายขั้นตอนย่อย ขั้นตอนย่อยแรก จะวิเคราะห์การทำงานกับผู้ใช้เพื่อระบุความต้องการของผู้ใช้จากระบบที่นำเสนอ ในขั้นตอนย่อยนี้จะรวมถึงการศึกษาอย่างระมัดระวังของระบบปัจจุบัน คู่มือการใช้ และ ที่ใช้คำนวณด้วยเครื่องคอมพิวเตอร์ ซึ่งจะแทนที่หรือเพิ่มความต้องการในส่วนหนึ่งของโครงการนี้ จากนั้น การศึกษาความต้องการและโครงสร้างที่สัมพันธ์กัน การจัดส่วนที่ซ้ำกัน ในฐานะที่เป็นส่วนหนึ่งของโครงสร้างจะสร้างการออกแบบเบื้องต้นทางเลือกให้ตรงกับความต้องการ จากนั้นจะเปรียบเทียบทางเลือกเหล่านี้เพื่อตรวจสอบว่าตรงตามความต้องการที่ดีที่สุดภายในต้นทุนแรงงาน และระดับเทคนิค องค์กรดำเนินการตามกระบวนการของการพัฒนา ผลของขั้นตอนการวิเคราะห์เป็นรายละเอียดของทางเลือกที่แนะนำโดยทีมวิเคราะห์ เมื่อคำแนะนำเป็นที่ยอมรับโดยองค์กร ก็จะสามารถทำแผนการที่จะจัดซื้อฮาร์ดแวร์และซอฟต์แวร์ระบบตามจำเป็นในการสร้างหรือการใช้งานระบบตามที่เสนอ

3. ออกแบบระบบ

ระยะที่สามของ SDLC เรียกว่าการออกแบบระบบ ในระหว่างการออกแบบระบบ นักวิเคราะห์จะแปลงคำอธิบายของการแก้ปัญหาทางเลือกที่แนะนำไปเป็นตรรกะแล้วจะกำหนดระบบทางกายภาพจะออกแบบทุกแง่มุมของระบบจากหน้าจอร์ับข้อมูลและส่งข้อมูล การออกรายงาน ฐานข้อมูล และกระบวนการคอมพิวเตอร์ การออกแบบเชิงตรรกะไม่ได้ผูกติดอยู่กับฮาร์ดแวร์ใด ๆ ที่เฉพาะเจาะจงและแพลตฟอร์มระบบของซอฟต์แวร์ ในทางทฤษฎีระบบที่ออกแบบสามารถนำไปใช้กับฮาร์ดแวร์และซอฟต์แวร์ระบบใด ๆ ได้ การออกแบบเชิงตรรกะจะมุ่งเน้นในด้านระบบธุรกิจ นั่นคือ วิธีการที่ระบบจะส่งผลกระทบต่อการทำงานภายในองค์กร ทั้งการออกแบบทางตรรกะสำหรับผลิตภัณฑ์ และการออกแบบทางกายภาพสำหรับการเปรียบเทียบ จะดูได้จากการเปรียบเทียบว่าการตัดสินใจที่เฉพาะเจาะจงหลาย ๆ อย่าง ที่จะทำที่จะย้ายจากรูปแบบตรรกะไปเป็นผลิตภัณฑ์ทางกายภาพ

สถานการณ์จะคล้ายกันในการออกแบบระบบสารสนเทศ ในการออกแบบทางกายภาพจะเปลี่ยนจากการออกแบบเชิงตรรกะไปเป็นทางกายภาพ หรือทางเทคนิค ตัวอย่างเช่น จะต้องแปลงไดอะแกรมที่ตรงกับความต้องการ การไหลของข้อมูล และการประมวลผลของข้อมูลในระบบ ไปเป็นการออกแบบระบบโครงสร้างที่สามารถจะแบ่งออกเป็นหน่วยที่เล็กลงและมีขนาดเล็กสำหรับการแปลงคำสั่งที่เขียนในภาษาการเขียนโปรแกรม การออกแบบส่วนต่าง ๆ ของระบบในการดำเนินการทางกายภาพที่จำเป็นในการอำนวยความสะดวกในการเก็บข้อมูล การประมวลผล และการส่งออกสารสนเทศ ในระหว่างการออกแบบทางกายภาพที่ทีมงานนักวิเคราะห์ตัดสินใจว่าจะใช้การเขียนโปรแกรมภาษาคอมพิวเตอร์โดยมีคำสั่งการเขียนในระบบฐานข้อมูลและโครงสร้างเพิ่มข้อมูลจะถูกใช้สำหรับข้อมูล และจะใช้แพลตฟอร์มฮาร์ดแวร์ ระบบปฏิบัติการ และสภาพแวดล้อมในระบบเครือข่ายที่จะทำงานภายใต้ระบบได้ การตัดสินใจเหล่านี้ ในที่สุด ได้เริ่มต้นของการวางแผนฮาร์ดแวร์และซอฟต์แวร์ ในตอนท้ายของขั้นตอนการวิเคราะห์ และในตอนนี้สามารถได้รับเทคโนโลยีใหม่ ๆ ที่ไม่ได้อยู่ในองค์กร ผลิตภัณฑ์สุดท้ายของขั้นตอนการออกแบบจะเป็นข้อกำหนดของระบบทางกายภาพ ที่นำเสนอในรูปแบบ เช่น ไดอะแกรมหรือรายงานเป็นลายลักษณ์อักษร ที่พร้อมที่จะส่งไปยังนักเขียนโปรแกรมและผู้พัฒนาระบบอื่น ๆ ในการพัฒนา

4. การทำให้เกิดผลและการดำเนินงานของระบบ

ขั้นตอนสุดท้ายของ SDLC เป็นกระบวนการแบบสองขั้นตอน คือ ระบบการนำไปใช้และการดำเนินงาน ในระหว่างการนำเอาระบบไปใช้และการดำเนินงาน จะเปลี่ยนจากการระบุระบบไปเป็นระบบที่ทำงานได้ซึ่งจะทำการทดสอบแล้วและนำไปใช้ การนำไปใช้ รวมถึงการเขียนโปรแกรม การทดสอบ และการติดตั้ง ในระหว่างการเขียนโปรแกรม นักเขียนโปรแกรมเขียนโปรแกรมที่จะพัฒนาระบบ ในระหว่างการทดสอบ นักเขียนโปรแกรมและนักวิเคราะห์ทดสอบแต่ละโปรแกรมและระบบทั้งหมดเพื่อที่จะหาข้อแก้ไขและข้อผิดพลาด ในระหว่างการติดตั้งระบบ ระบบใหม่นั้นจะกลายเป็นส่วนหนึ่งของกิจกรรมในชีวิตประจำวันขององค์กร โปรแกรมประยุกต์ที่ได้มีการติดตั้งหรือโหลด บนฮาร์ดแวร์ที่มีอยู่หรือฮาร์ดแวร์ใหม่ จากนั้นผู้ใช้นำไปใช้เป็นระบบใหม่ และได้รับการฝึกอบรม การเริ่มต้นการวางแผนสำหรับทั้งการทดสอบและการติดตั้งเป็นช่วงต้นของการวางแผนโครงการและขั้นตอนการเลือกเพราะทั้งสองต้องใช้อย่างกว้างขวางการวิเคราะห์เพื่อพัฒนาว่าวิธีการที่เหมาะสม

กิจกรรมการนำไปใช้ของระบบยังรวมถึงการสนับสนุนผู้ใช้ เช่น เอกสารสรุป โปรแกรมการฝึกอบรม และความช่วยเหลือผู้ใช้งานอย่างต่อเนื่อง เอกสารและโปรแกรมการฝึกอบรมจะสรุป ในระหว่างการนำไปใช้ เอกสารสร้างขึ้นมาตลอดวงจรชีวิต แลการฝึกอบรม (และการศึกษา) เกิดขึ้นจากการเริ่มของโครงการ การนำไปใช้ของระบบสามารถดำเนินต่อไปตราบเท่าที่ระบบที่มีอยู่อย่างต่อเนื่องเพราะการสนับสนุนผู้ใช้นี้ยังเป็นส่วนหนึ่งของการนำไปใช้ แม้จะมีความพยายามที่ดีที่สุดของนักวิเคราะห์ระบบ ผู้จัดการ และโปรแกรมเมอร์ แต่การติดตั้งไม่ได้เป็น

กระบวนการง่าย ๆ หลายระบบการออกแบบที่ดีจะล้มเหลวเนื่องจากกระบวนการติดตั้งเป็นความผิดพลาด แม้กระทั่งระบบการออกแบบที่สามารถล้มเหลวหากการนำไปใช้ที่ได้ทำโดยทีมโครงการทำให้เกิดผล

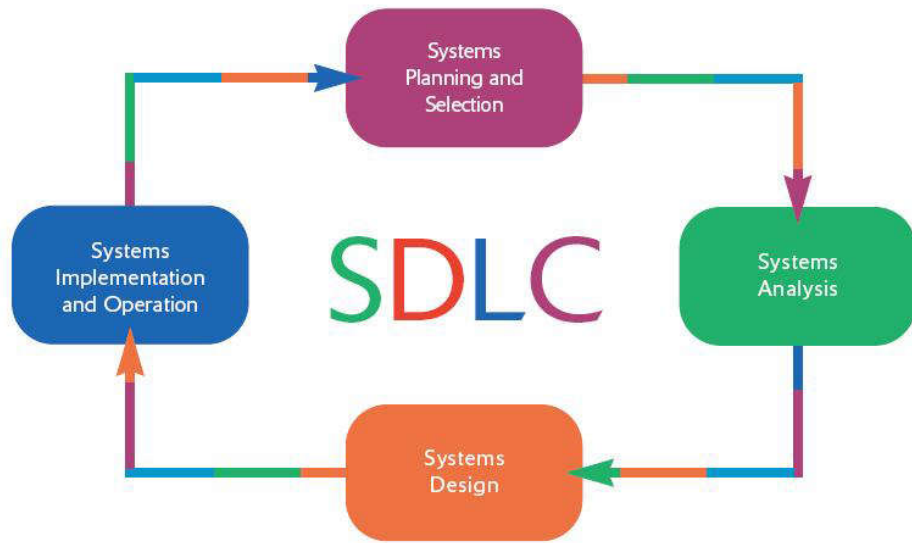


FIGURE 1-10
THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)

แนวทางการพัฒนา

การสร้างต้นแบบโดยใช้เครื่องมือวิศวกรรมซอฟต์แวร์ใช้คอมพิวเตอร์ช่วย (Computer-Aided Software Engineering : CASE) เครื่องมือการพัฒนาระบบร่วมกัน (Joint Application Design : JAD) การพัฒนาระบบงานแบบรวดเร็ว (Rapid Application Development : RAD) การออกแบบมีส่วนร่วม (Participatory Design : PD) และ การใช้วิธีการคล่องตัว (Agile Methodologies) เป็นตัวแทนของวิธีการที่แตกต่างกันที่ความคล่องตัวและปรับปรุงกระบวนการวิเคราะห์และออกแบบระบบจากมุมมองที่แตกต่างกัน

1. การสร้างต้นแบบ การออกแบบและการสร้างให้มีการลดขนาดลง แต่มีการทำงานรุ่นของระบบที่ต้องการเป็นที่รู้จักกันต้นแบบ ต้นแบบสามารถที่จะพัฒนาด้วยเครื่องมือ CASE ผลิตภัณฑ์ซอฟต์แวร์ที่ทำโดยอัตโนมัติในขั้นตอนวงจรชีวิตการพัฒนาระบบ เครื่องมือ CASE ทำการสร้างต้นแบบให้ง่ายขึ้นและมีความคิดสร้างสรรค์มากขึ้นโดยการสนับสนุนการออกแบบของหน้าจอและรายงานและส่วนอื่น ๆ ของระบบส่วนติดต่อผู้ใช้ เครื่องมือ CASE ยังสนับสนุนหลายเทคนิคไดอะแกรมที่จะได้เรียนรู้ เช่น แผนภาพการไหลข้อมูลและแผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล นักวิเคราะห์ระบบทำงานกับผู้ใช้ เพื่อระบุความต้องการเริ่มต้นหรือขั้นพื้นฐานสำหรับระบบ นักวิเคราะห์จากนั้นสร้างต้นแบบได้อย่างรวดเร็ว เมื่อต้นแบบเสร็จสมบูรณ์แล้ว ผู้ใช้จะทำงานกับต้นแบบนี้ และบอกนักวิเคราะห์ระบบในสิ่งที่ชอบและไม่ชอบเกี่ยวกับต้นแบบนี้ นักวิเคราะห์ใช้ความคิดเห็นนี้ไป

ปรับปรุงต้นแบบและนำรุ่นใหม่กลับไปให้กับผู้ใช้ ขั้นตอนนี้จะทำแบบนี้ซ้ำแล้วซ้ำอีกต่อไปจนกว่าผู้ใช้มีความพึงพอใจค่อนข้างกับสิ่งที่ได้ใช้ ข้อดีที่สำคัญของเทคนิคการสร้างต้นแบบนี้คือ (1) เข้าไปมีส่วนร่วมกับผู้ใช้ในการวิเคราะห์และออกแบบ และ (2) ได้รูปแบบความต้องการแบบรูปธรรม มากกว่าที่จะได้จากวาจาหรือนามธรรม นอกจากนี้จะเป็นแบบไม่ขึ้นอยู่กับใครแล้ว ต้นแบบนี้ยังอาจถูกนำมาใช้เพื่อเพิ่ม SDLC ตัวอย่างเช่นเป็นต้นแบบของระบบขั้นสุดท้ายอาจได้รับการพัฒนาในช่วงต้นของการวิเคราะห์ที่จะช่วยให้นักวิเคราะห์ระบุสิ่งที่ผู้ใช้ต้องการ จากนั้นระบบสุดท้ายคือการพัฒนาตามข้อกำหนดของแบบเดิม

2. เครื่องมือวิศวกรรมซอฟต์แวร์ใช้คอมพิวเตอร์ช่วย (CASE) วิศวกรรมซอฟต์แวร์ใช้คอมพิวเตอร์ช่วย (Computer-aided software engineering : CASE) หมายถึงเครื่องมือซอฟต์แวร์อัตโนมัติที่ใช้โดยนักวิเคราะห์ระบบในการพัฒนาระบบสารสนเทศ เครื่องมือเหล่านี้สามารถใช้อัตโนมัติหรือสนับสนุนกิจกรรมตลอดทั้งกระบวนการพัฒนาระบบที่มีวัตถุประสงค์ของการเพิ่มผลผลิตและการปรับปรุงคุณภาพโดยรวมของระบบ CASE จะช่วยให้มีระเบียบวินัยวิศวกรรมชนิดในการพัฒนาซอฟต์แวร์และระบบอัตโนมัติของกระบวนการวงจรชีวิตซอฟต์แวร์ทั้งหมด ที่บางครั้งกับความคุ้นเคยของเครื่องมือซอฟต์แวร์แบบบูรณาการ โดยทั่วไป CASE ช่วยผู้สร้างระบบในการจัดการความซับซ้อนของโครงการระบบสารสนเทศและช่วยให้มั่นใจว่าระบบที่มีคุณภาพสูงถูกสร้างขึ้นในเวลาและภายในงบประมาณ ผู้ขายของผลิตภัณฑ์ CASE ได้เปิดระบบผ่านการใช้งานมาตรฐานฐานข้อมูลและประโยชน์ในการแปลงข้อมูลที่จะแบ่งปันสารสนเทศในผลิตภัณฑ์และเครื่องมือที่ง่ายขึ้น ฐานข้อมูลแบบบูรณาการและมาตรฐานที่เรียกว่าพื้นที่เก็บข้อมูลเป็นวิธีที่พบในการให้สินค้าและบูรณาการเครื่องมือและได้รับปัจจัยสำคัญในการช่วยให้การจัดการ CASE โครงการที่มีขนาดใหญ่ ที่ซับซ้อนมากขึ้นได้ง่ายขึ้นและต่อเนื่องรวมข้อมูลข้ามเครื่องมือต่าง ๆ และผลิตภัณฑ์ ประเภททั่วไปเครื่องมือ CASE รวมถึง

- เครื่องมือวาดแผนภาพที่ให้กระบวนการของระบบ ข้อมูล และโครงสร้างการควบคุมที่จะแสดงแบบกราฟิก
- คอมพิวเตอร์แสดงและสร้างรายงานที่ช่วยให้ต้นแบบของระบบรู้ถึงผู้ใช้ การแสดง (หรือรูปแบบ) และการสร้างรายงานทำให้ง่ายขึ้นสำหรับนักวิเคราะห์ระบบ เพื่อระบุความต้องการของระบบ และความสัมพันธ์
- เครื่องมือวิเคราะห์ที่ทำตรวจสอบอัตโนมัติ ในการทำไม่สมบูรณ์ ไม่สอดคล้องกัน หรือไม่ถูกต้องในแผนภาพ รูปแบบ และรายงาน
- ที่เก็บข้อมูลส่วนกลางสามารถรวมที่เก็บข้อมูลของภาระบุ แผนภาพ รายงาน และโครงการการจัดการสารสนเทศ

- การสร้างเอกสารจะช่วยให้การผลิตทั้งทางเทคนิคและเอกสารผู้ใช้ในรูปแบบมาตรฐานที่สร้างโปรแกรมช่วยให้สร้างโปรแกรมแบบอัตโนมัติ และฐานข้อมูลจากเอกสาร จากแผนภาพจากรูปแบบ และจากรายงานที่ออกแบบมา

Joint Application Design (JAD)

ในช่วงปลายปี 1970 ระบบการพัฒนาบุคลากรที่ IBM พัฒนาระบบการใหม่ในการรวบรวมความต้องการของระบบสารสนเทศและการออกแบบระบบการตรวจสอบ กระบวนการนี้เรียกว่าการพัฒนาแบบร่วมมือกัน (JAD) ความคิดที่อยู่เบื้องหลัง JAD เป็นโครงสร้างขั้นตอนการกำหนดความต้องการของการวิเคราะห์และการตรวจสอบสิ่งที่เกิดขึ้นเป็นส่วนหนึ่งของการออกแบบ ผู้ใช้ผู้จัดการ และนักพัฒนาระบบจะถูกนำมาวมกันสำหรับชุดของการประชุมที่มีโครงสร้างแบบเร่งรัดดำเนินการโดยผู้นำ JAD โดยการรวบรวมคนที่ได้รับผลกระทบโดยตรงจากระบบสารสนเทศเข้ามาอยู่ในห้องเดียวกันในเวลาเดียวกันเพื่อการทำงานร่วมกันจะเห็นด้วยกับความต้องการของระบบ และรายละเอียดการออกแบบ เวลาและ ทรัพยากรขององค์กรที่มีการจัดการที่ดีขึ้น สมาชิกกลุ่มมีแนวโน้มที่จะพัฒนาความเข้าใจร่วมกันของสิ่งที่ระบบสารสนเทศนั้นควรจะทำ JAD ได้กลายเป็นเรื่องธรรมดาในอุตสาหกรรมบางอย่างเช่นการประกัน และในบริษัทที่เฉพาะเจาะจง เช่น CIGNA

Rapid Application Development (RAD)

การสร้างต้นแบบ CASE และ JAD เป็นเครื่องมือที่สำคัญที่สนับสนุนการพัฒนาระบบงานอย่างรวดเร็ว (RAD) หลักการของวิธี RAD เป็นความล่าช้าในการผลิตเอกสารรายละเอียดการออกแบบระบบจนกระทั่งหลังจากที่มีความชัดเจนในต้องการของผู้ใช้ ต้นแบบทำหน้าที่เป็นคำอธิบายการทำงานของความต้องการ RAD ที่เกี่ยวข้องกับการได้รับการยอมรับของผู้ใช้ในส่วนติดต่อผู้ใช้และการพัฒนาระบบที่สำคัญ ที่มีความสามารถในการทำให้เร็วที่สุดเท่าที่เป็นไปได้ RAD ถูกนำมาใช้กันอย่างแพร่หลายโดยบริษัทที่ปรึกษา ยังใช้เป็นวิธีการในบริษัทโดยบริษัท เช่น Boeing Company

RAD ทำงานกับคอมพิวเตอร์อย่างเต็มประสิทธิภาพเพื่อได้รับความสามารถของมนุษย์ในการสร้างและสร้างใหม่กับระบบที่กำลังทำงานอย่างรวดเร็ว ในทางกลับกันวิธี RAD สามารถมองเห็นหลักการพัฒนาระบบที่สำคัญซึ่งอาจส่งผลในปัญหาที่เกิดขึ้นกับระบบที่จะถูกพัฒนาด้วยวิธีนี้

RAD พัฒนามาจากการบรรจบกันของแนวโน้มสองอย่าง นั่นคือ ความเร็วที่ได้เพิ่มขึ้น และความวุ่นวายของการดำเนินธุรกิจในปี 1980 และต้นปี 1990 และมีความพร้อมของเครื่องมือที่ใช้คอมพิวเตอร์ในการ

ขับเคลื่อนสูงเพื่อสนับสนุนการพัฒนาระบบและบำรุงรักษาอย่างง่าย ในเงื่อนไขของการทำธุรกิจในการเปลี่ยนแปลง การแข่งขันในสภาพแวดล้อมของโลกที่มีความวุ่นวายมากขึ้น การจัดการในหลาย ๆ องค์กรเริ่มตั้งคำถามว่าจะรออีก 2 - 3 ปีเพื่อพัฒนาระบบก็จะล้าสมัยเมื่อเสร็จสิ้น อีกด้านหนึ่งเครื่องมือ CASE และซอฟต์แวร์ต้นแบบถูกเผยแพร่ไปทั่วองค์กรทำให้เป็นเรื่องง่ายสำหรับผู้ใช้ในการดูว่าระบบจะมีลักษณะอย่างไรก่อนที่จะเสร็จสมบูรณ์ ทำไมไม่ใช้เครื่องมือเหล่านี้เพื่อที่จะชี้ไปยังปัญหาของระบบที่กำลังพัฒนาขึ้นได้มีผลผลิตสภาพแวดล้อมทางธุรกิจที่เปลี่ยนแปลงไปอย่างรวดเร็ว ดังนั้น RAD จึงเกิดขึ้นมา ขั้นตอนที่เหมือนกันที่ทำตามแบบอย่างของ SDLC และยังทำตามแบบ RAD แต่ขั้นตอนจะรวมกันเพื่อผลิตเทคนิคการพัฒนาความคล่องตัวมากขึ้น ขั้นตอนการวางแผนและการออกแบบใน RAD จะสั้นลงโดยเน้นการทำงานเกี่ยวกับหน้าที่ของระบบและความต้องการของส่วนติดต่อผู้ใช้ที่ค่าใช้จ่ายของการวิเคราะห์รายละเอียดทางธุรกิจและความกังวลต่อปัญหาในการทำงานของระบบ นอกจากนี้ RAD มักจะมีลักษณะที่ระบบจะทำการพัฒนาที่แยกจากระบบอื่น ๆ จึงช่วยลดกิจกรรมที่ใช้เวลานานในการประสานงานกับมาตรฐานที่มีอยู่และระบบในระหว่างการออกแบบและการพัฒนา สิ่งที่น่าสนใจใน RAD โดยทั่วไปมีน้อยลงในลำดับและโครงสร้างของกระบวนการวงจรชีวิตและอื่น ๆ ในการทำงานที่แตกต่างกันในแบบคู่ขนานกับส่วนอื่น ๆ และเกี่ยวกับการใช้การสร้างต้นแบบอย่างกว้างขวาง การทำวนไปวนมาในวงจรชีวิตของ RAD นั้นมีข้อจำกัดในขั้นตอนการออกแบบและการพัฒนาซึ่งเป็นกลุ่มของการทำงานในแนวทางของ RAD ที่เกิดขึ้นถึงแม้ว่าจะเป็นไปได้ใน RAD เพื่อที่จะกลับไปวางแผนในครั้งเดียวแล้วการออกแบบจะได้เริ่มขึ้นนั้นจะทำได้ ในทำนองเดียวกันแม้ว่าจะเป็นไปได้ที่จะกลับไปสู่การพัฒนาจากขั้นตอนการปรับเปลี่ยนระบบ (เมื่อระบบกลับไปที่ผู้ใช้) RAD ถูกออกแบบมาเพื่อลดการทำซ้ำที่จุดนี้ในวงจรชีวิต ระดับสูงของความมุ่งมั่นของผู้ใช้และการมีส่วนร่วมตลอด RAD หมายความว่าระบบที่พัฒนาออกมาควรจะได้รับ การยอมรับมากขึ้นอย่างรวดเร็วโดยชุมชนผู้ใช้ (และด้วยเหตุนี้การนำไปใช้จะง่ายขึ้นในระหว่างการปรับเปลี่ยนระบบ) กว่าที่จะพัฒนาระบบแล้วโดยใช้เทคนิคแบบดั้งเดิม ใช้การสร้างต้นแบบที่จะชะลอการผลิตการออกแบบระบบจนกระทั่งหลังจากที่ต้องการของผู้ใช้มีความชัดเจน

Participatory Design (PD)

การออกแบบอย่างมีส่วนร่วม (PD) ได้พัฒนาในยุโรปเหนือ ซึ่งแสดงให้เห็นถึงแนวทางในทางเลือกที่ทำงานได้ โดยใช้ SDLC หนึ่งใน บริษัท ที่รู้จักกันดีว่ามีการใช้วิธีนี้คือ StatoilHydro Norwegian oil company PD เน้นในบทบาทของผู้ใช้มากขึ้นกว่าการทำเทคนิคแบบดั้งเดิมในทวีปอเมริกาเหนือ เช่น การวิเคราะห์โครงสร้างและการออกแบบโครงสร้าง ในบางกรณี PD อาจเกี่ยวข้องกับชุมชนผู้ใช้ทั้งหมดในกระบวนการพัฒนา ผู้ใช้แต่ละคนมีความเท่าเทียมกันในการกำหนดความต้องการของระบบและในการอนุมัติการออกแบบระบบ ในกรณีอื่น ๆ

กลุ่มได้รับการเลือกตั้งของผู้ใช้ควบคุมกระบวนการ ผู้ใช้เหล่านี้เป็นตัวแทนที่มีขนาดใหญ่ ชุมชน เหมือนกับสถานิติบัญญัติแสดงให้เห็นถึงความอยากและความต้องการของการเลือกตั้งโดยปกติภายใต้ PD นักวิเคราะห์ระบบทำงานกับผู้ใช้ การจัดการขององค์กรและที่ปรึกษาด้านนอกให้คำแนะนำมากกว่าการควบคุม PD คือผลมาจากส่วนหนึ่งของบทบาทของการใช้แรงงานและการจัดการในสถานที่ทำงานของยุโรปทางตอนเหนือที่ใช้แรงงานเป็นระเบียบมากขึ้น ประกอบอิทธิพลมากขึ้น และมีส่วนร่วมอย่างใกล้ชิดมากขึ้นกับการเปลี่ยนแปลงทางเทคโนโลยีกว่าจะเป็นจริงในทวีปอเมริกาเหนือ

Agile Method

ในขณะที่คิดวิธีการอื่น ๆ อีกมากมายที่จะวิเคราะห์และออกแบบระบบที่ได้รับการพัฒนาขึ้นกว่าปีที่ผ่านมา วิธีการเหล่านี้รวมถึงการเขียนโปรแกรม eXtreme วิธีการ Crystal family การพัฒนาซอฟต์แวร์ยืดหยุ่น Scrum และ Feature Driven Development ในเดือนกุมภาพันธ์ปี 2001 หลายที่ได้แสดงแนวทางทางเลือกเหล่านี้ซึ่งได้พบกันที่ยุทธาโทในประเทศสหรัฐอเมริกาและถึงฉันทามติในหลายหลักการพื้นฐานต่าง ๆ ก็จะมีหลายวิธีการอยู่ ฉันทามตินี้กลายเป็นเอกสารที่พวกเขาเรียกว่า “The Agile Manifesto” โดยวิธี Agile เหล่านี้ได้แบ่งออกเป็น 3 หลักการที่สำคัญ (1) มุ่งเน้นไปที่วิธีการปรับตัวมากกว่าวิธีการคาดเดา (2) มุ่งเน้นไปที่บุคคลมากกว่าบทบาท และ (3) กระบวนการปรับตัวเอง การนำการปรับตัวมาใช้ มากกว่าที่จะอ้างอิงวิธีการคาดเดา เพื่อที่จะสังเกตการณ์ ซึ่งวิธีการที่ขึ้นอยู่กับวิศวกรรมวิธีการทำงานที่ดีที่สุดเมื่อกระบวนการและผลิตภัณฑ์มีการคาดเดาซอฟต์แวร์จะไม่มีแนวโน้มไปตามที่คาดเดาไว้ เป็นอย่างที่สุด โดยเฉพาะอย่างยิ่งในทุกวันนี้สภาพแวดล้อมทางธุรกิจที่ปั่นป่วน วิธีการปรับตัวอื่น ๆ ที่มีความจำเป็นแล้ว และวิธีการ Agile จะขึ้นอยู่กับความสามารถในการปรับตัวได้อย่างรวดเร็วการมุ่งเน้นไปที่บุคคลมากกว่าบทบาทนี้ยังมีการวิจารณ์ของเทคนิคที่ขึ้นอยู่กับวิศวกรรม โดยที่บุคคลจะกลายเป็นการแทนกัน ในมุมมองของวิธีการ Agile คนเป็นบุคคลที่มีความสามารถ บุคคลไม่ใช่บทบาท แต่ละบุคคลมีความสามารถเฉพาะบุคคลที่จะนำไปพัฒนาโครงการ สุดท้ายวิธีการ Agile สนับสนุนกระบวนการพัฒนาซอฟต์แวร์การปรับตัวด้วยตนเอง ในขณะที่วิธีการถูกนำมาใช้ ควรที่จะปรับโดยทีมพัฒนาเฉพาะ ที่ทำงานในโครงการเฉพาะในบริบทเฉพาะ ไม่มีวิธีการใดที่มีประสิทธิภาพเหมาะกับนักพัฒนาทั้งหมดในทุกโครงการกับทุกครั้ง

ซอฟต์แวร์นั้นจะเป็นส่วนใหญ่ของระบบโปรแกรมประยุกต์ทางธุรกิจใด ๆ แม้ว่าซอฟต์แวร์ส่วนมากจะเขียนแค่ครั้งเดียวในบริษัทโดยนักวิเคราะห์ระบบและนักเขียนโปรแกรมที่เป็นเจ้าของบริษัท การทำแบบนี้จะไม่มีแนวโน้ม ในปัจจุบันนี้ อุตสาหกรรมซอฟต์แวร์ที่เจริญรุ่งเรือง สามารถซื้อซอฟต์แวร์เพียงสำหรับธุรกิจใด ๆ ได้ แต่ทุกธุรกิจมีความเป็นเอกลักษณ์ และไม่มีซอฟต์แวร์ที่เหมาะสมกับบริษัทที่มี และความต้องการของบริษัทจริง ๆ ซอฟต์แวร์จะต้องมีการปรับเปลี่ยนเพื่อให้เหมาะสมกับความต้องการที่เฉพาะเจาะจงของบริษัท หลายครั้ง โปรแกรม

ประยุกต์ทางธุรกิจจะเป็นการรวมกันของชิ้นส่วนของซอฟต์แวร์ ซอฟต์แวร์ที่ซื้อมาหรือได้มาจากผู้ขายหลายอย่างที่แตกต่างกันและบูรณาการโดยพนักงานเทคโนโลยีสารสนเทศภายในบริษัท แต่ที่เจ้าหน้าที่หาความต้องการของซอฟต์แวร์ที่ต้องใช้ได้ทีไหน จะได้เรียนรู้เกี่ยวกับหลายแหล่งที่มาของซอฟต์แวร์ที่สามารถมีอยู่กับนักวิเคราะห์ระบบในปัจจุบันนี้

การจัดจ้างคนภายนอก

ถ้าหากองค์กรอื่นพัฒนาหรือทำงานโปรแกรมประยุกต์สำหรับคอมพิวเตอร์ในองค์กร การทำงานแบบนี้ที่เรียกว่า การจัดจ้างคนภายนอก ซึ่งการจัดจ้างคนภายนอกรวมถึงความคิดวิธีการในการจัดการทำงาน ที่ร้ายแรงมากจะเกิดขึ้นกับการพัฒนาบริษัทและการทำงานของโปรแกรมประยุกต์บนคอมพิวเตอร์ นั่นคือ แค่อการจัดหาการนำเข้าและการนำออก ตัวอย่างที่พบเป็น บริษัทที่ทำงานโปรแกรมประยุกต์การจ่ายเงินเดือนให้กับลูกค้าไม่ได้มีการพัฒนาระบบบัญชีเงินเดือนเป็นอิสระในบริษัท แต่ให้ข้อมูลเงินเดือนของพนักงานให้กับบริษัท และมีค่าใช้จ่ายผลตอบแทน บริษัทจะคืนเสร็จเงินเดือนที่สมบูรณ์ รายงานบัญชีเงินเดือนและภาษี และงบอื่น ๆ สำหรับพนักงานสำหรับหลาย ๆ องค์กรวิธีการที่มีประสิทธิภาพมากที่สุดในการจัดการการดำเนินงานการจ่ายเงินเดือนผ่านการจัดจ้างคนภายนอก ในอีกตัวอย่างของการเตรียมการจัดจ้างคนภายนอก ซึ่งจะจ้างบริษัทที่จะทำงานโปรแกรมประยุกต์ที่บริษัทบนเครื่องคอมพิวเตอร์ ในบางกรณี องค์กรจะจ้างเช่นการจัดการจัดการซึ่งจะทำให้ระบบสารสนเทศบางส่วนหรือทั้งหมดหายไป และโอนย้ายส่วนใหญ่หรือทั้งหมดของระบบสารสนเทศพนักงานไปให้กับบริษัท เพื่อที่จะนำเข้ามาเพื่อทำงานในการคำนวณขององค์กร ทำไมองค์กรที่จะจัดจ้างภายนอกดำเนินงานระบบสารสนเทศได้ ตัวอย่างการจ่ายเงินเดือน การจัดจ้างภายนอกอาจจะทำให้ต้นทุนมีประสิทธิภาพ หากบริษัทมีความเชี่ยวชาญในการทำงานการจ่ายเงินเดือนสำหรับบริษัทอื่น ๆ ก็สามารถใช้ประโยชน์จากเศรษฐกิจที่จะประสบความสำเร็จจากการทำงานในการใช้โปรแกรมประยุกต์ด้านคอมพิวเตอร์ที่มีเสถียรภาพสำหรับหลายองค์กรได้ในราคาที่ต่ำ แต่ทำไมองค์กรที่จะทำให้การประมวลผลสารสนเทศนั้นหายไปทั้งหมดและนำมาในบริษัท ภายนอกมาจัดการกับโปรแกรมประยุกต์ด้านคอมพิวเตอร์

เหตุผลแรกอาจจะเอาชนะกับปัญหาการดำเนินงานขององค์กรในการเผชิญหน้าในระบบสารสนเทศ ยกตัวอย่างเช่น เมือง Grand Rapids Michigan จ้างบริษัทภายนอกเพื่อให้ทำงานในศูนย์คอมพิวเตอร์เมื่อสามสิบปีที่ผ่านมา เพื่อในการจัดการพนักงานของศูนย์คอมพิวเตอร์ที่ดีขึ้น สัญญาการทำงานร่วมกันและข้อจำกัดข้าราชการพลเรือนนั้น มีข้อบังคับทำให้ยากที่จะไล่คนออกจากงานนำเข้ามาในองค์กรการจัดการสิ่งอำนวยความสะดวกในการเรียกใช้การดำเนินงานด้านคอมพิวเตอร์ของตน ดังนั้นเมืองนี้จะนำองค์กรการจัดการสาธารณสุขประโศก เพื่อที่จะทำงานทางคอมพิวเตอร์ได้ และสามารถที่จะกำจัดปัญหาของพนักงานในเวลาเดียวกัน เหตุผลอื่น ๆ

สำหรับการจัดจ้างภายนอกทั้งหมดนั้นคือ การจัดการขององค์กรอาจจะรู้สึกว่าการกิจหลักไม่ได้เกี่ยวข้องกับการจัดการระบบสารสนเทศ และอาจจะบรรลุการทำงานด้านคอมพิวเตอร์ที่มีประสิทธิภาพมากขึ้นโดยการดำเนินงานที่มีประสบการณ์ของบริษัทที่มุ่งเน้นทางคอมพิวเตอร์ Kodak ได้ตัดสินใจในช่วงปลายปี 1980ว่าจะไม่อยู่ในธุรกิจโปรแกรมประยุกต์ด้านคอมพิวเตอร์และหันมาบริหารจัดการเครื่องเมนเฟรมที่ IBM และจัดการคอมพิวเตอร์ส่วนบุคคลที่ Businessland การจัดจ้างภายนอกนั้น เป็นธุรกิจขนาดใหญ่ บางองค์กรจัดจ้างภายนอกสำหรับการพัฒนาสารสนเทศ และในหลายหน้าที่ทางสารสนเทศที่ค่าใช้จ่ายเป็นพันล้านดอลลาร์ ตลาดการจัดจ้างภายนอกแบบดั้งเดิมในตอนนี้มีมูลค่าในอุตสาหกรรมถึง 280 ล้านดอลล่า และตลาดการย้ายฐานการผลิตไปต่างประเทศมีมูลค่า 70 ล้านดอลล่า ผู้ค้าการจัดจ้างภายนอกส่วนบุคคลเข้าสู่ระบบสัญญาที่มีขนาดใหญ่สำหรับการให้บริการ IBM และ HP เป็นสองบริษัทที่ใหญ่ที่สุด เป็นที่รู้จักทั่วโลกที่เป็นบริษัทจัดจ้างภายนอก ทั้งสองบริษัท มีสัญญาการจัดจ้างภายนอกกับบริษัทหลายที่การจัดจ้างภายนอก เป็นทางเลือกที่นักวิเคราะห์มีความต้องการอย่างแน่นอนที่จะตระหนักถึง เมื่อมีการสร้างทางเลือกในการพัฒนากลยุทธ์สำหรับระบบ ในฐานะนักวิเคราะห์ควรพิจารณาองค์ประกอบความสามารถที่มีที่จะให้การบริการจัดจ้างภายนอก อาจจะต้องอย่างน้อยองค์กรได้พัฒนาแล้ว และใช้การทำงานโปรแกรมประยุกต์คล้ายกับในสิ่งที่ผู้ใช้ได้ขอร้อง บางที่การจัดจ้างภายนอกเพื่อแทนระบบอาจจะเป็นทางเลือกหนึ่งซึ่งจะรู้ว่าความต้องการของระบบก่อนที่จะพิจารณาการจัดจ้างภายนอก นั้นหมายความว่า มีความระมัดระวังในการประเมินผู้จัดหาของการบริการการจัดซื้อภายนอกคืออะไร ที่จะสามารถตอบสนองต่อความต้องการ แต่การไม่ตัดสินใจที่จะใช้การจัดซื้อภายนอก จะต้องพิจารณาส่วนประกอบของซอฟต์แวร์ที่จะแทนที่ระบบควรจะซื้อและไม่พัฒนาขึ้น

แหล่งที่มาของซอฟต์แวร์

สามารถจัดกลุ่มองค์กรที่ผลิตซอฟต์แวร์ออกเป็นหกประเภทหลัก

1. บริษัทที่ให้บริการด้านเทคโนโลยีสารสนเทศ
2. ผู้ผลิตซอฟต์แวร์สำเร็จรูป
3. ผู้ผลิตซอฟต์แวร์ที่แก้ปัญหาขององค์กร
4. การให้บริการการประมวลผลแบบกลุ่มเมฆ
5. ซอฟต์แวร์การจัดจ้างภายนอก และ
6. การพัฒนาในบริษัท

1. บริษัทที่ให้บริการด้านเทคโนโลยีสารสนเทศ หากบริษัทต้องการระบบสารสนเทศแต่ไม่ได้มีความเชี่ยวชาญหรือบุคลากรในการพัฒนาระบบในบริษัท และไม่มีระบบที่มีไว้พร้อมแล้วที่เหมาะสม บริษัทมีแนวโน้มที่จะปรึกษาบริษัทที่ให้บริการด้านเทคโนโลยีสารสนเทศ (IT) บริษัท ที่ให้บริการด้านเทคโนโลยีสารสนเทศช่วย

บริษัทในการพัฒนาระบบสารสนเทศที่กำหนดเองสำหรับใช้ภายในบริษัท พัฒนา แม่ข่าย และใช้งานโปรแกรมประยุกต์สำหรับลูกค้าหรือให้บริการอื่น ๆ ในตารางที่ 2-1 รายชื่อของสิบอันดับบริษัท ซอฟต์แวร์ระดับโลก ซึ่งจากสามในสิบมีความเชี่ยวชาญในการให้บริการซึ่งรวมถึงการพัฒนาระบบที่กำหนดเอง บริษัท เหล่านี้จ้างคนที่มีความเชี่ยวชาญในการพัฒนาระบบสารสนเทศ ที่ปรึกษาทำงานกับธนาคารที่เข้าใจสถาบันการเงินอย่างดีเท่ากับระบบสารสนเทศ ที่ปรึกษาใช้เทคนิควิธีการเดียวกันหลาย ๆ ทาง และเครื่องมือที่บริษัทใช้ในการพัฒนาระบบในบริษัท อาจจะทำให้ประหลาดใจที่จะเห็น IBM ที่ถูกระบุว่าเป็นผู้ผลิตซอฟต์แวร์ชั้นนำระดับโลก ซึ่งอาจคิดว่า IBM เป็นบริษัทด้านฮาร์ดแวร์เป็นหลัก แต่ IBM ได้ย้ายออกไปจากความเชื่อมั่นในการพัฒนาฮาร์ดแวร์เป็นเวลาหลายปี การซื้อของที่ปรึกษาด้านเทคโนโลยีสารสนเทศของ PricewaterhouseCoopers โดย IBM ในปี 2002 เป็นการอธิบายถึงการย้ายไปในด้านการบริการและให้คำปรึกษา IBM ยังเป็นที่รู้จักกันดีสำหรับการพัฒนาเครื่องบริการเว็บและซอฟต์แวร์มัลติเดิลแวร์ บริษัทที่ให้บริการด้านเทคโนโลยีสารสนเทศชั้นนำอื่น ๆ รวมถึง บริษัทที่ปรึกษาแบบดั้งเดิม เช่น Accenture รายการนี้ยังรวมถึง HP ซึ่งเป็นบริษัทอื่น ๆ ที่ก่อนหน้านี้มุ่งเน้นไปที่ฮาร์ดแวร์ที่ทำให้การเปลี่ยนไปเป็นบริษัทผู้ให้บริการด้านเทคโนโลยีสารสนเทศ

2. ผู้ผลิตซอฟต์แวร์สำเร็จรูป การเจริญเติบโตของอุตสาหกรรมซอฟต์แวร์มีปรากฏการณ์ตั้งแต่จุดเริ่มต้นในช่วงกลางทศวรรษที่ 1960 ตอนนี้อย่างน้อยของบริษัทยักษ์ใหญ่ที่สุดในโลกสำรวจโดยนิตยสาร Software เป็นบริษัทที่ผลิตซอฟต์แวร์โดยเฉพาะ บริษัทซอฟต์แวร์จะพัฒนาในสิ่งที่เรียกว่าบางครั้งระบบบรรจุเสร็จก่อน หรือระบบที่มีไว้พร้อม Microsoft's Project และ Intuit's Quicken QuickPay และ QuickBooks เป็นตัวอย่างของซอฟต์แวร์ที่นิยม อุตสาหกรรมการพัฒนาซอฟต์แวร์สำเร็จรูปบริการกลุ่มตลาดจำนวนมาก การนำเสนอซอฟต์แวร์จากทั่วไป ซอฟต์แวร์สำหรับเสร็จที่มีหลากหลาย เช่นบัญชีแยกประเภททั่วไป ที่เฉพาะมากขึ้น เช่น ซอฟต์แวร์ที่จะช่วยในการจัดการศูนย์รับเลี้ยงเด็กกลางวัน บริษัทซอฟต์แวร์จะพัฒนาซอฟต์แวร์เพื่อให้ทำงานบนแพลตฟอร์มคอมพิวเตอร์ที่แตกต่าง จากไมโครคอมพิวเตอร์ไปสู่คอมพิวเตอร์เมนเฟรมขนาดใหญ่ จำนวนของพนักงานในบริษัทจะมีตั้งแต่เพียงไม่กี่คนไปถึงหลายพัน บริษัทซอฟต์แวร์ให้คำปรึกษากับผู้ใช้ระบบหลังจากเริ่มต้นการออกแบบซอฟต์แวร์จนเสร็จสมบูรณ์แล้วและหลังจากที่รุ่นแรกของระบบได้รับการสร้างขึ้น ระบบจะมีการทดสอบในองค์กรที่ใช้งานจริงที่เพื่อที่จะหาปัญหาหรือตรวจสอบปัญหาการปรับปรุงใด ๆ ที่สามารถทำได้ จนกระทั่งการทดสอบเสร็จแล้ว ระบบจะไม่นำเสนอขายให้กับประชาชนทั่วไป บางระบบซอฟต์แวร์สำเร็จรูปของไม่สามารถที่จะแก้ไขได้เพื่อตอบสนองความเฉพาะเจาะจง แต่ละความต้องการขององค์กรโดยเฉพาะได้ ผู้ผลิตระบบงานดังกล่าว บางครั้งเรียกว่าระบบแบบครบวงจร ผู้ผลิตของระบบแบบครบวงจรจะทำการเปลี่ยนแปลงซอฟต์แวร์ก็ต่อเมื่อมีจำนวนของผู้ใช้มากพอที่จะขอการเปลี่ยนแปลงที่เฉพาะเจาะจง ซอฟต์แวร์ประยุกต์สำเร็จรูปอื่น ๆ นั้นสามารถแก้ไขหรือขยาย แต่จะทำได้โดยผู้ผลิตหรือผู้ใช้เพื่อให้พอดีกับความต้องการขององค์กรอย่างใกล้ชิดที่สุด แม้ว่าหลายองค์กรจะทำงานคล้ายกัน ไม่มีทั้งสององค์กรจะทำการสิ่งเดียวกันในวิถีทางเดียวกัน ระบบแบบครบวงจรอาจจะดีพอสำหรับการทำงานบางระดับ แต่ระบบนี้จะไม่สมบูรณ์แบบซึ่งตรงกับวิธีการที่องค์กรได้ในการทำธุรกิจ

การประมาณการที่เหมาะสมคือซอฟต์แวร์สำเร็จรูปที่สามารถตอบสนองความต้องการที่ดีที่สุดร้อยละ 70 ของความต้องการองค์กร ดังนั้นแม้ในกรณีที่ดีที่สุด ร้อยละ 30 ของระบบซอฟต์แวร์ที่ใช้ไม่ดีที่สุดที่ตรงกับข้อกำหนดขององค์กร

3. ผู้ผลิตซอฟต์แวร์ที่แก้ปัญหาขององค์กร ซอฟต์แวร์สำหรับแก้ปัญหาองค์กร ดังได้กล่าวก่อนหน้านี้ มีองค์กรมากขึ้นเรื่อย ๆ ที่มีการเลือกที่สมบูรณ์การแก้ปัญหาโดยใช้ซอฟต์แวร์ที่เรียกว่าการแก้ปัญหาองค์กรหรือระบบการวางแผนทรัพยากรขององค์กร (ERP) เพื่อสนับสนุนการดำเนินงานและกระบวนการทางธุรกิจ การแก้ปัญหาโดยใช้ซอฟต์แวร์ ERP เหล่านี้ ประกอบด้วยชุดของโมดูลแบบบูรณาการ แต่ละโมดูลสนับสนุนการทำงานของธุรกิจส่วนบุคคลแบบดั้งเดิม เช่น การบัญชี, การจัดจำหน่าย, การผลิต, และทรัพยากรมนุษย์ ความแตกต่างระหว่างโมดูลและวิธีการแบบดั้งเดิมคือ โมดูลจะมีการบูรณาการโดยให้ความสำคัญกับกระบวนการทางธุรกิจมากกว่าขอบเขตการทำงานของธุรกิจ ตัวอย่างเช่น ชุดของโมดูลจะสนับสนุนการนำข้อมูลเข้าเป็นลำดับทั้งหมด จากการได้รับใบสั่งสินค้าเพื่อที่จะปรับสินค้าคงคลังเพื่อจัดส่งไปเรียกเก็บเงิน วิธีการแบบดั้งเดิมจะใช้ระบบที่แตกต่างกันในขอบเขตการทำงานที่แตกต่างกันของธุรกิจ เช่น ระบบการเรียกเก็บเงินในบัญชี และระบบสินค้าคงคลังในคลังสินค้า การใช้ระบบ ERP บริษัทสามารถบูรณาการ ทุกส่วนของกระบวนการทางธุรกิจในระบบสารสนเทศแบบครบวงจร ทุกแง่มุมของรายการเดียวที่เกิดขึ้นต่อเนื่องในระบบสารสนเทศเดียวมากกว่าในชุดที่ไม่ปะติดปะต่อกัน ระบบจะแบ่งแยก โดยมุ่งเน้นไปที่ขอบเขตการทำงานทางธุรกิจ ประโยชน์ของวิธีการแก้ปัญหาขององค์กรรวมถึงจะมีพื้นที่เก็บข้อมูลเดียว ข้อมูลทุกด้านของกระบวนการทางธุรกิจและความยืดหยุ่นของโมดูล จะมีพื้นที่เก็บข้อมูลเดียวเพื่อให้แน่ใจว่าข้อมูลนั้นสอดคล้องกันมากขึ้นและข้อมูลมีความถูกต้อง ซึ่งจะมีการซ่อมบำรุงน้อย โมดูลจะมีความยืดหยุ่นเพราะโมดูลที่เพิ่มเติมเข้าไปนั้นสามารถเพิ่มตามความจำเป็นเมื่อพื้นฐานที่มีอยู่ในระบบแล้ว โมดูลที่เพิ่มเข้ามาได้ทันที จะบูรณาการเข้ากับระบบที่มีอยู่ ซอฟต์แวร์แก้ปัญหาขององค์กรยังเกี่ยวข้องกับข้อเสียบางอย่าง ระบบมีความซับซ้อน ดังนั้นการดำเนินงานการติดตั้งนั้น ใช้เวลานานในการดำเนินเพื่อจะเสร็จสมบูรณ์ องค์กรมักจะไม่มีความเชี่ยวชาญที่จำเป็นในบริษัทในติดตั้งระบบ ดังนั้น จำเป็นจะต้องพึ่งพาที่ปรึกษาหรือพนักงานของบริษัทผู้ผลิตซอฟต์แวร์ซึ่งอาจมีราคาแพง ในบางกรณีองค์กรต้องเปลี่ยนวิธีการทำธุรกิจเพื่อที่จะได้รับประโยชน์จากการพัฒนาไปสู่การแก้ปัญหาขององค์กร ผู้ผลิตรายใหญ่จำนวนมากนำเสนอซอฟต์แวร์แก้ปัญหาองค์กร ผู้ขายที่รู้จักกันดีน่าจะเป็น SAP AG ซึ่งเป็นบริษัทเยอรมันที่รู้จักสำหรับผู้นำผลิตภัณฑ์ R/3 โดย SAP ย่อมาจากระบบ (System) โปรแกรมประยุกต์ (Application) และผลิตภัณฑ์ (Product) ในการประมวลผลข้อมูล SAP AG ก่อตั้งขึ้นในปี 1972 แต่ส่วนใหญ่เติบโตตั้งแต่ปี 1992 ในปี 2009 SAP ของสหรัฐอเมริกาเป็นผู้จัดจำหน่ายที่ใหญ่ที่สุดในอันดับที่เจ็ดของซอฟต์แวร์ในโลก ผู้ผลิตอื่น ๆ ที่สำคัญของการแก้ปัญหาขององค์กร คือ Oracle Corp. ซึ่งเป็นบริษัทอยู่ที่สหรัฐอเมริกา ซึ่งบางทีอาจจะเป็นที่รู้จักกันดีสำหรับซอฟต์แวร์ฐานข้อมูล Oracle อยู่ในลำดับที่สี่ในรายการของสิบอันดับสำหรับบริษัท ซอฟต์แวร์ในปี 2009) ในปลายปี 2004 Oracle ได้ PeopleSoft, Inc ซึ่งเป็นบริษัทที่ก่อตั้งขึ้นในสหรัฐอเมริกาในปี 1987 PeopleSoft ได้เริ่มต้นด้วยการแก้ปัญหาขององค์กรที่มุ่งเน้นไปที่การจัดการ

ทรัพยากรมนุษย์และการขยายตัวเพื่อให้ครอบคลุมการเงิน การจัดการวัสดุ การกระจาย และการผลิตก่อนที่ Oracle ชื่อบริษัทนี้ เพียงแค่ก่อนที่จะถูกซื้อโดย Oracle, PeopleSoft มีการสนับสนุนความแข็งแกร่งของบริษัท ในปี 2003 ผ่านการซื้อของผู้ขาย ERP รายอื่น ๆ นั่นคือ J.D. Edwards ในปี 2009 SAP มีร้อยละ 31 ของตลาด โปรแกรมประยุกต์ขององค์กรหลักทั่วโลก ในฐานะที่เป็นตลาดระดับไฮเอนด์ได้กลายเป็นอิมิตัวกับระบบ ERP ซึ่งส่วนใหญ่ผู้ขาย ERP กำลังมองหาเพื่อธุรกิจขนาดกลางและขนาดเล็กทำมีการเจริญเติบโต

4. การให้บริการการประมวลผลแบบคลาวด์ เป็นอีกวิธีหนึ่งสำหรับองค์กรที่จะขอรับแอปพลิเคชัน คือ การเช่าหรือขออนุญาตจากผู้ให้บริการบุคคลที่สามที่ใช้แอปพลิเคชันในไคลด์ระยะไกล ผู้ใช้สามารถเข้าถึงแอปพลิเคชันผ่านทางอินเทอร์เน็ตหรือผ่านเครือข่ายส่วนตัวเสมือน (VPN) ได้ ผู้ให้บริการแอปพลิเคชันซื้อ ติดตั้ง บำรุงรักษา และอัปเดตแอปพลิเคชัน ผู้ใช้ชำระค่าใช้จ่ายต่อการใช้งานหรืออนุญาตให้ใช้ซอฟต์แวร์โดยทั่วไปเดือนต่อเดือน แม้ว่าการปฏิบัตินี้ได้รับการรู้จักกันโดยชื่อที่แตกต่างกันหลายปีที่ผ่านมา ซึ่งเรียกว่า การให้บริการการประมวลผลแบบกลุ่มเมฆ

5. Open-Source Software โอเพนซอร์สซอฟต์แวร์จะแตกต่างจากซอฟต์แวร์ชนิดอื่น ๆ โอเพนซอร์สซอฟต์แวร์ที่แตกต่างกันเพราะที่มีอยู่นั้นเป็นสาธารณะ ไม่เป็นเพียงผลิตภัณฑ์สุดท้าย แต่ที่รหัสต้นฉบับ (Source Code) นอกจากนี้ยังแตกต่างกันเพราะได้รับการพัฒนาโดยชุมชนของผู้ที่สนใจแทนที่จะเป็นพนักงานของบริษัท โดยเฉพาะ โอเพนซอร์สซอฟต์แวร์มีหน้าที่เช่นเดียวกับซอฟต์แวร์เชิงพาณิชย์ เช่น ระบบปฏิบัติการ, อีเมล, ระบบฐานข้อมูล, เว็บเบราว์เซอร์ และอื่น ๆ บางส่วนเป็นที่รู้จักกันดีและชื่อโอเพนซอร์สซอฟต์แวร์ที่นิยมเป็นลินุกซ์ (ระบบปฏิบัติการ), MySQL (ระบบฐานข้อมูล) และ Firefox (เว็บเบราว์เซอร์) นอกจากนี้โอเพนซอร์สซอฟต์แวร์ยังนำไปใช้กับส่วนประกอบของซอฟต์แวร์และออบเจ็กต์ โอเพนซอร์สซอฟต์แวร์ได้รับการพัฒนาและดูแลรักษาโดยชุมชนของผู้คน ชุมชนเหล่านี้บางครั้งอาจจะมีขนาดใหญ่มาก นักพัฒนามักจะใช้ทรัพยากรเว็บทั่วไป เช่น SourceForge.net เพื่อการจัดกิจกรรมของพวกเขา ในเดือนธันวาคม 2010 SourceForge.net ที่เป็นแม่ข่าย มีมากกว่า 260,000 โครงการ และมีมากกว่า 2.7 ล้านคนที่เป็นสมาชิกที่ลงทะเบียน ไม่มีปัญหาเกี่ยวกับการที่การเคลื่อนไหวของโอเพนซอร์สซอฟต์แวร์จะไม่ประสบความสำเร็จ ถ้าไม่มีสภาพพร้อมใช้ของอินเทอร์เน็ตสำหรับการให้บริการในการเข้าถึงและการจัดกิจกรรมการพัฒนา

หากซอฟต์แวร์ไม่มีข้อจำกัด อาจจะสงสัยว่าใครหาเงินได้อย่างไร โดยการพัฒนาโอเพนซอร์สซอฟต์แวร์ บริษัทและบุคคลสามารถสร้างรายได้กับโอเพนซอร์สซอฟต์แวร์ (1) โดยการให้การบำรุงรักษาและบริการอื่น ๆ หรือ (2) โดยการให้รุ่นหนึ่งของซอฟต์แวร์โดยไม่มีข้อจำกัดและขายรุ่นอื่น ๆ โดยมีคุณลักษณะเพิ่มเติมที่ บางการแก้ปัญหาของโอเพนซอร์สมีมากขึ้นจึงมีผลกระทบต่ออุตสาหกรรมซอฟต์แวร์กว่าอย่างอื่น ๆ Firefox เป็นตัวอย่าง ซึ่งได้รับความสำเร็จอย่างมากในตลาดเว็บเบราว์เซอร์ ที่คาดว่าจะมีร้อยละ 24 ของส่วนแบ่งตลาด ผลิตภัณฑ์โอเพนซอร์สซอฟต์แวร์อื่น ๆ เช่น MySQL ก็ยังได้รับการประสบความสำเร็จและส่วนแบ่งของโอเพนซอร์สซอฟต์แวร์ในอุตสาหกรรมซอฟต์แวร์ดูเหมือนว่าจะเติบโตอย่างต่อเนื่อง

6. การพัฒนาเองภายในองค์กร จะมีหลายประเภท องค์กรภายนอกที่ทำหน้าที่เป็นแหล่งซอฟต์แวร์ แต่ภายในองค์กรพัฒนายังคงเป็นทางเลือก แน่แน่นอนว่าการพัฒนาในองค์กรไม่จำเป็นต้องมีต้องพัฒนาซอฟต์แวร์ทั้งหมดที่จะสร้างระบบทั้งหมด การแก้ปัญหาจะเป็นแบบผสมผสานที่เกี่ยวข้องกับซอฟต์แวร์ที่ซื้อและซอฟต์แวร์ภายในตัวมีองค์ประกอบทั่วไป ส่วนประกอบซอฟต์แวร์ภายในบางตัวถูกนำกลับมาใช้ใหม่

เลือกซอฟต์แวร์สำเร็จรูป

เมื่อได้ตัดสินใจที่จะซื้อซอฟต์แวร์สำเร็จรูปมากกว่าที่จะเขียนโปรแกรมบางส่วนหรือทั้งหมดของซอฟต์แวร์สำหรับระบบใหม่ จะตัดสินใจว่าจะซื้ออย่างไร หลายเกณฑ์ที่ต้องพิจารณาและสิ่งนี้อาจเกิดขึ้นอยู่กับการซื้อซอฟต์แวร์ สำหรับแต่ละมาตรฐาน การเปรียบเทียบอย่างชัดเจนควรจะทำระหว่างซอฟต์แวร์สำเร็จและกระบวนการของการพัฒนาโปรแกรมประยุกต์เดียวกันในบริษัท มีดังนี้

1. ต้นทุน
2. การทำงาน
3. การสนับสนุนของผู้ขาย
4. ความเจริญเติบโตของผู้ขาย
5. ความยืดหยุ่น
6. เอกสาร
7. เวลาตอบสนอง
8. ความง่ายในการติดตั้ง

ความสำคัญที่สัมพันธ์กันของมาตรฐานเหล่านี้จะแตกต่างจากโครงการหนึ่งไปอีกโครงการหนึ่ง และจากองค์กรหนึ่งไปอีกองค์กรหนึ่ง ถ้าจะต้องเลือกสองเกณฑ์ที่อยู่ในสำคัญที่สำคัญที่สุด ทั้งสองอาจจะผู้ผลิตสนับสนุนและการเจริญเติบโตของผู้ขาย จะไม่ต้องการซอฟต์แวร์ลิขสิทธิ์จากผู้ขายที่มีชื่อเสียงสำหรับการสนับสนุนที่ไม่ดี ในทำนองเดียวกันคุณไม่ต้องการที่จะได้รับการเกี่ยวข้องกับผู้ใช้ที่อาจจะไม่อยู่ในธุรกิจในวันพรุ่งนี้ จะมีวิธีการจัดอันดับความสำคัญในเกณฑ์ที่เหลือขึ้นอยู่กับสถานการณ์เฉพาะต้นทุนจะเกี่ยวข้องกับการเปรียบเทียบค่าใช้จ่ายในการพัฒนาระบบเดียวกันในบริษัทเพื่อค่าใช้จ่ายในการซื้อหรือซอฟต์แวร์สำเร็จที่มีลิขสิทธิ์ ต้องแน่ใจว่ามีการเปรียบเทียบของต้นทุนในการจัดซื้อเพื่อการอัพเกรดหรือจ่ายค่าธรรมเนียมประจำปีสำหรับลิขสิทธิ์ที่มี ค่าใช้จ่ายที่จะต้องเสียในการรักษาซอฟต์แวร์ ค่าใช้จ่ายสำหรับการจัดซื้อและการพัฒนาในบริษัทสามารถนำมาเปรียบเทียบบนพื้นฐานของมาตรการความเป็นไปได้ทางเศรษฐกิจ

การทำงานหมายถึงงานของซอฟต์แวร์ที่สามารถทำ และการดำเนินการที่จำเป็น และคุณสมบัติของระบบ ที่ความต้องการ ซอฟต์แวร์สำเร็จรูปสามารถที่จะดำเนินการทั้งหมดหรือแค่เพียงบางส่วนของงานที่ผู้ใช้ต้องการ ถ้า บางอย่างแล้วยังจะสามารถดำเนินการงานหลักที่จำเป็นได้หรือไม่ จะได้ความต้องการของผู้ใช้จะเกิดขึ้นในตอนท้าย ของขั้นตอนการวิเคราะห์เพราะจะไม่สามารถประเมินซอฟต์แวร์สำเร็จรูปได้จนกระทั่งต้องการของผู้ใช้ ได้รับการ รวบรวมและเป็นโครงสร้าง การจัดซื้อซอฟต์แวร์ประยุกต์ไม่ได้ใช้แทน สำหรับการดำเนินการขั้นตอนการวิเคราะห์ ระบบก่อนหน้าเป็นการสนับสนุนจากผู้ผลิตหมายถึงว่าผู้ขายสามารถให้ การสนับสนุนและวิธีการได้มากเพียงใด การ สนับสนุนรวมถึงความช่วยเหลือในการติดตั้งซอฟต์แวร์

การอบรมผู้ใช้และพนักงานระบบเกี่ยวกับซอฟต์แวร์และการให้ความช่วยเหลือปัญหาเกิดขึ้นหลังจากการ ติดตั้ง เมื่อเร็ว ๆ นี้ บริษัท ซอฟต์แวร์จำนวนมากได้อย่างมีการลดปริมาณของการสนับสนุนไม่เสียค่าใช้จ่ายให้กับ ลูกค้าเพื่อให้ค่าใช้จ่ายในการใช้งาน โทรศัพท์ หรือสิ่งอำนวยความสะดวกสนับสนุนกระดานข่าวคอมพิวเตอร์ควรจะ ได้รับการพิจารณา สิ่งที่เกี่ยวข้องกับการสนับสนุนการเจริญเติบโตของผู้ขาย จุดหลังนี้ควรจะมีน้อยที่สุด อุตสาหกรรมซอฟต์แวร์เป็นแบบพลวัตมาก และซอฟต์แวร์โปรแกรมประยุกต์แบบนวัตกรรม จะถูกสร้างขึ้นโดย ผู้ประกอบการที่ทำงานจากที่สำนักงานบ้าน ซึ่งเป็นอุตสาหกรรมในครัวเรือนที่เป็นรูปแบบดั้งเดิม องค์กรดังกล่าว แม้จะมีซอฟต์แวร์ที่โดดเด่นแต่ก็มักจะไม่มียุทธศาสตร์หรือความสามารถในการบริหารจัดการธุรกิจที่จะอยู่ใน

การทำธุรกิจจะยาว นอกจากนั้นการแข่งขันได้เคลื่อนไหวของโดยบริษัทซอฟต์แวร์ที่สำคัญสามารถทำให้ ผลผลิตของบริษัทขนาดเล็กล้าสมัยไปได้หรือไม่เข้ากันกับระบบปฏิบัติการ บริษัทซอฟต์แวร์หนึ่งที่ได้พูดถึงการ ทำงานกับซอฟต์แวร์บนระบบปฏิบัติการ Windows, Mac OS หรือโทรศัพท์มือถือ ยังคงดำเนินต่อไปกับการ เปลี่ยนแปลงของฮาร์ดแวร์และซอฟต์แวร์ระบบที่อาจจะมีความซับซ้อนในบริษัท ขนาดเล็กที่จะสามารถจัดการได้ และ ซอฟต์แวร์ประยุกต์สำเร็จรูปที่คืนนั้นจะหายไป ความยืดหยุ่นหมายถึงวิธีการที่เข้าถึงได้ เพื่อในการปรับแต่ง ซอฟต์แวร์ หากซอฟต์แวร์ที่ไม่ยืดหยุ่นพอที่ผู้ใช้อาจจะต้องปรับตัวไปในทางที่เหมาะสมกับซอฟต์แวร์ มีแนวโน้มที่จะ ปรับตัวในลักษณะนี้หรือไม่ ซอฟต์แวร์ที่ซื้อสามารถแก้ไขได้ในหลายวิธี บางครั้งผู้ขายจะทำให้การเปลี่ยนแปลงที่ กำหนดเองได้ ถ้ามีความยินดีที่จะจ่ายสำหรับการออกแบบใหม่และการเขียนโปรแกรมใหม่ ผู้ขายบางคนออกแบบ ซอฟต์แวร์สำหรับการปรับแต่ง ตัวอย่างเช่น ซอฟต์แวร์อาจมีวิธีที่แตกต่างกันของการประมวลผลข้อมูลและในเวลา การติดตั้ง ลูกค้าเลือกที่จะเริ่มต้น นอกจากนี้จอแสดงผลและรายงานอาจมีการออกแบบใหม่ได้ง่ายดาย ถ้าโมดูล เหล่านี้ถูกเขียนในภาษา 4GL รายงาน รูปแบบ และการแสดงผล อาจจะปรับแต่งได้อย่างง่ายโดยใช้กระบวนการ โดยชื่อ บริษัท และชื่อเรื่องที่เลือกสำหรับรายงาน จอแสดงผล รูปแบบ และส่วนหัวของคอลัมน์ จะเลือกจากตาราง ของพารามิเตอร์ได้ อาจต้องการที่จะใช้งานบางส่วนของเทคนิคการปรับแต่งเหล่านี้เหมือนกันสำหรับระบบที่พัฒนา ในบริษัทดังนั้นซอฟต์แวร์สามารถที่จะปรับได้ง่ายจึงเหมาะสำหรับหน่วยธุรกิจที่แตกต่างกัน สายการผลิต หรือ

หน่วยงาน การจัดทำเอกสารรวมถึงคู่มือการผู้ใช้ เช่นเดียวกับเอกสารทางเทคนิค เอกสารจะทำให้มีวิธีการเข้าใจ และปรับปรุงให้เป็นปัจจุบันอย่างไร ค่าใช้จ่ายเท่าไร สำหรับการทำให้หลายสำเนา ถ้ามีความจำเป็นต้องทำ เวลาในการตอบสนองหมายถึงใช้เวลานานเท่าใดสำหรับซอฟต์แวร์สำเร็จรูปที่จะตอบสนองต่อการร้องขอของผู้ใช้ใน ช่วงเวลาสื่อสารแบบปฏิสัมพันธ์กัน ตัวชี้วัดของเวลาจะใช้เวลานานเท่าใดของซอฟต์แวร์ที่จะทำงานจนเสร็จสมบูรณ์ ท้ายที่สุดความสะดวกในการติดตั้งเป็นตัวชี้วัดของความยากลำบากของการบรรจุซอฟต์แวร์และทำให้ทำงานได้

1. การตรวจสอบข้อมูลซอฟต์แวร์ที่ซื้อ

วิธีหนึ่งที่จะได้รับสารสนเทศทั้งหมด จะต้องมีซอฟต์แวร์สำเร็จรูปที่จะเก็บรวบรวมได้จากผู้ขาย บางส่วนของสารสนเทศนี้อาจจะมีอยู่ในเอกสารซอฟต์แวร์และวรรณกรรมการตลาดทางเทคนิค ข้อมูลอื่น ๆ สามารถให้ตามคำขอ ตัวอย่างเช่น สามารถส่งคำท้วงติงให้กับผู้ขายในแบบสอบถามที่เป็นคำถามเฉพาะเจาะจง เกี่ยวกับโปรแกรมสำเร็จรูป แบบสอบถามนี้อาจเป็นส่วนหนึ่งของการขอซื้อเสนอ (RFP) หรือขอใบเสนอราคา (RFQ) ดำเนินการองค์กรที่ต้องการเมื่อซื้อสินค้าที่สำคัญ ถ้าตัดสินใจว่าฮาร์ดแวร์ใหม่หรือซอฟต์แวร์ระบบมีความเป็นไปได้ที่จะความสามารถในการทำงานได้ จะให้จะออกข้อเสนอ (RFP) ให้กับผู้ขาย RFP จะถามผู้ขายเกี่ยวกับการเสนอฮาร์ดแวร์และซอฟต์แวร์ระบบที่ตรงกับต้องการของระบบใหม่ การออก RFP เปิดโอกาสให้ผู้ขายได้ ดำเนินการวิจัยที่ต้องการเพื่อที่จะตัดสินใจในทางเลือกต่างๆ การขอให้แต่ละการเสนอราคาที่ส่งมาจากผู้ขายมี ข้อมูลบางอย่างที่จำเป็นสำหรับการตัดสินใจว่าจะอะไรจะดีที่สุดที่สอดคล้องกับความต้องการ ตัวอย่างเช่น สามารถขอข้อมูลผลการดำเนินงานที่เกี่ยวข้องกับความเร็วและจำนวนการดำเนินงานต่อวินาที สามารถถามเกี่ยวกับความ น่าเชื่อถือของเครื่องและความพร้อมในการบริการและไม่ว่าการติดตั้งตั้งอยู่ใกล้เคียงที่จะสามารถหาข้อมูลเพิ่มเติม ได้ สามารถขอให้มีส่วนร่วมในการสาธิตของฮาร์ดแวร์ การเสนอราคายังจะรวมถึงข้อมูลเกี่ยวกับค่าใช้จ่าย

แน่นอนที่สุดการใช้งานของซอฟต์แวร์และการทำงานผ่านการทดสอบตามเกณฑ์สำหรับการเลือก ซอฟต์แวร์จะได้ทางที่ดีที่สุดสำหรับการประเมิน การทดสอบไม่เพียงแต่ซอฟต์แวร์เท่านั้น แต่จะเกี่ยวกับเอกสาร เอกสารการฝึกอบรม และแม้แต่สิ่งอำนวยความสะดวกสนับสนุนทางเทคนิคด้วยในการหาความต้องการ สามารถหาผู้ผลิตซอฟต์แวร์ที่คาดหวังเป็นส่วนหนึ่งของขั้นตอนการประมูลคือ (ที่ติดตั้ง) ค่าธรรมเนียมหรือค่าใช้จ่ายตามที่ได้ตกลงไว้ (ซอฟต์แวร์ที่มีระยะเวลาจำกัดในคอมพิวเตอร์ วิธีนี้สามารถกำหนดวิธีการที่ซอฟต์แวร์จะทำงานใน สภาพแวดล้อมได้อย่างไร หรือไม่ได้อยู่ในสภาพแวดล้อมที่ดีที่สุดที่มีอยู่ บางอย่างที่น่าเชื่อถือมากที่สุดและเป็น แหล่งที่เชี่ยวชาญของข้อเสนอแนะเป็นผู้ใช้อื่น ๆ ของซอฟต์แวร์ ผู้ขายมักจะให้รายชื่อของลูกค้า (ซึ่งจะบอกเกี่ยวกับ ความพึงพอใจของลูกค้าซึ่งอาจจะข้ามความต้องการของลูกค้า) และบุคคลที่คาดหวังที่จะติดต่อเป็นลูกค้าที่ คาดหวังไว้แล้ว ที่นี้จะป็นเครือข่ายส่วนบุคคลของการติดต่อ ที่พัฒนาโดยกลุ่มมืออาชีพ เพื่อน สมาคมการค้า หรือ

สโมสรรธุรกิจท้องถิ่นซึ่งจะสามารถเป็นแหล่งทรัพยากรได้ โดยไม่ลังเลที่จะหารายชื่อ ลูกค้าปัจจุบันหรือลูกค้าในอดีตตั้งสามารถให้ความเข้าใจอย่างลึกซึ้งซึ่งในการใช้งานของโปรแกรมสำเร็จรูปสำหรับองค์กร จะได้รับความคิดเห็นเกี่ยวกับโปรแกรมสำเร็จรูปที่เป็นไปได้ ก็จะสามารถใช้บริการทดสอบซอฟต์แวร์อิสระที่ประเมินซอฟต์แวร์เป็นระยะ ๆ และเก็บรวบรวมความคิดเห็นกับผู้ใช้ การสำรวจดังกล่าวที่มีอยู่ที่มีค่าใช้จ่ายไม่ว่าจะเป็นบริการสมัครสมาชิกหรือตามความต้องการ บางครั้งการสำรวจที่เป็นกลางปรากฏในสิ่งพิมพ์ทางการค้าบ่อยครั้ง มีในบทความในสิ่งพิมพ์ทางการค้า แม้แต่ความคิดเห็นซอฟต์แวร์เป็นจริงโดยผู้ผลิตซอฟต์แวร์ และไม่เป็นกลาง ถ้าหากกำลังเปรียบเทียบกับหลายซอฟต์แวร์สำเร็จรูป สามารถกำหนดคะแนนแต่ละโปรแกรมสำเร็จรูปในแต่ละเกณฑ์และเปรียบเทียบคะแนนใช้วิธีเชิงปริมาณ

การนำมาใช้ใหม่

การนำมาใช้ใหม่คือการใช้ทรัพยากรซอฟต์แวร์ที่เขียนขึ้นก่อนหน้านี้ในโปรแกรมประยุกต์ใหม่ เพราะหลายส่วนของโปรแกรมประยุกต์ที่ค่อนข้างทั่วไปสามารถข้ามไปใช้งานได้กับโปรแกรมประยุกต์อื่นได้ ซึ่งดูเหมือนว่าใช้งานง่าย ประหยัดเงินได้ดี ก็สามารถทำได้ในหลายทาง ถ้าส่วนของโปรแกรมไม่ได้เขียนใหม่ในแต่ละครั้งที่มีต้องการ การนำกลับมาใช้ใหม่ควรเพิ่มประสิทธิภาพการทำงานของโปรแกรมเมอร์ เพราะสามารถที่จะใช้ซอฟต์แวร์ที่มีอยู่สำหรับการทำงานบางอย่างได้ นั่นหมายความว่าสามารถดำเนินการมากขึ้นในการทำงานในเวลาเดียวกัน การนำกลับมาใช้ใหม่ก็จะลดเวลาการพัฒนา ลดกำหนดเวลาทำการ เพราะส่วนของซอฟต์แวร์ที่มีอยู่มีอยู่นั้นได้ทำการทดสอบแล้ว การนำกลับมาใช้ใหม่นั้นมีแนวโน้มที่จะส่งผลให้ซอฟต์แวร์ที่มีคุณภาพสูงขึ้นด้วย โดยลดอัตราข้อบกพร่อง ลดค่าใช้จ่ายในการซ่อมบำรุง แม้ว่าการนำกลับมาใช้ใหม่สามารถนำไปใช้ด้านที่แตกต่างกันของซอฟต์แวร์ โดยปกติจะมีการใช้กันมากที่สุดในเทคโนโลยีสองอย่างที่มีการพัฒนาแตกต่างกัน นั่นคือ การพัฒนาเชิงวัตถุ (object-oriented) และ การพัฒนาองค์ประกอบพื้นฐาน (component-based) ตัวอย่างเช่น การพิจารณาคลาสออบเจกต์ที่สร้างขึ้นเพื่อจำลองการทำงานของพนักงาน คลาสออบเจกต์ Employee จะมีทั้งข้อมูลเกี่ยวกับพนักงานและคำแนะนำที่จำเป็นสำหรับการคำนวณเงินเดือนในแต่ละประเภทงาน คลาสออบเจกต์สามารถนำมาใช้ในโปรแกรมใด ๆ ที่เกี่ยวข้องกับพนักงาน แต่หากมีการเปลี่ยนแปลงในการคำนวณเงินเดือนสำหรับพนักงานที่แตกต่างกัน การเปลี่ยนแปลงที่จะต้องได้รับการสร้างขึ้นเพื่อคลาสออบเจกต์ และไม่ต้องเปลี่ยนทุกโปรแกรมประยุกต์ที่ใช้ ตามคำนิยามที่ใช้คลาสออบเจกต์ Employee มีมากกว่าหนึ่งโปรแกรมก็จะถือว่าเป็นการนำกลับมาใช้ใหม่

การพัฒนาองค์ประกอบพื้นฐาน มีความคล้ายคลึงกับการพัฒนาเชิงวัตถุที่ให้ความสำคัญกับการสร้างชิ้นส่วนทั่วไปของซอฟต์แวร์ที่สามารถนำมาใช้สลับกันในโปรแกรมที่แตกต่างกัน ส่วนประกอบอาจจะเป็นขนาดเล็ก

หรือวัตถุประสงค์ที่มีขนาดใหญ่พอๆกันกับที่เป็นส่วนของซอฟต์แวร์ที่จัดการการทำงานทางธุรกิจเดียว เช่น การแปลงสกุลเงิน แนวคิดที่อยู่เบื้องหลังการพัฒนาคอร์ปอเรชันพื้นฐาน คือการนำโปรแกรมประยุกต์จากองค์ประกอบที่แตกต่างกันในระดับที่แตกต่างกันของความซับซ้อนและขนาด ผู้ค้าจำนวนมากทำงานสำหรับการพัฒนาคลังของส่วนประกอบ ที่สามารถนำมาใช้และประกอบตามความจำเป็นในการใช้งานที่ต้องการ หลักฐานบางอย่างที่แสดงให้เห็นว่า การนำกลับมาใช้ใหม่ทำได้อย่างมีประสิทธิภาพ โดยเฉพาะอย่างยิ่งสำหรับคลัสเตอร์ซอฟต์แวร์ ตัวอย่างเช่น การศึกษาในห้องปฏิบัติการพบว่า การนำกลับมาใช้ใหม่ของคลัสเตอร์ซอฟต์แวร์ส่งผลให้ผลผลิตที่เพิ่มขึ้น ลดความหนาแน่นของข้อบกพร่อง และลดการทำงานซ้ำ สำหรับ HP โปรแกรมที่นำกลับมาใช้ใหม่ผลในการตัดเวลาในการตลาดสำหรับผลิตภัณฑ์บางโดย ปัจจัยที่สามอย่างหรือมากกว่า จากสิบแปดเดือนน้อยกว่าห้าเดือน อย่างไรก็ตาม เพื่อการนำกลับมาใช้ใหม่ในการทำงานในการตั้งค่าขององค์กร มีหลายประเด็นที่แตกต่างกันที่มีอยู่ ปัญหาทางเทคนิค รวมถึงการขาดของวิธีการปัจจุบันในการสร้างและความชัดเจนในการกำหนดและการติดตามชิ้นส่วนที่นำกลับมาใช้ใหม่สำหรับการจัดวางในคลังและมีจำนวนน้อยของแหล่งทรัพยากรซอฟต์แวร์ที่นำมาใช้ใหม่และเชื่อถือได้ที่มีอยู่ ในขณะนี้ ปัญหาขององค์กรที่สำคัญรวมถึงการขาดความรับผิดชอบที่จะนำกลับมาใช้ใหม่ รวมทั้งขาดการฝึกอบรมที่เหมาะสม และผลตอบแทนที่จำเป็นในการส่งเสริม ซึ่งการขาดการสนับสนุนจากองค์กรสำหรับการจัดตั้งนำกลับมาใช้ใหม่ และความยากลำบากในการวัดประโยชน์ทางเศรษฐกิจจากการที่นำกลับมาใช้ใหม่ เพราะค่าใช้จ่ายที่ควรคำนึงถึงในการพัฒนาคอร์ปอเรชันที่นำกลับมาใช้ใหม่นั้น องค์กรส่วนใหญ่ไม่สามารถแข่งขันทางเศรษฐกิจกับองค์กรธุรกิจที่จัดตั้งขึ้นมา ที่มุ่งเน้นการขายส่วนประกอบที่เป็นสายหลักของธุรกิจ การประสบความสำเร็จขึ้นอยู่กับความสามารถในการใช้ประโยชน์จากค่าใช้จ่ายของส่วนประกอบโดยข้ามผู้ใช้ขนาดใหญ่และพื้นฐานโครงการ (รูปที่ 2-3) ปัญหาทางกฎหมายที่สำคัญและสัญญาที่เกี่ยวข้องกับการนำกลับมาใช้ใหม่ของคลัสเตอร์ซอฟต์แวร์ และ ส่วนประกอบเพิ่มเติมที่ใช้ในโปรแกรมประยุกต์อื่น ๆ ที่ยังต้องได้รับการแก้ไข เมื่อผู้บริหารขององค์กรตัดสินใจที่จะดำเนินการนำกลับมาใช้ใหม่ให้เป็นกลยุทธ์ สิ่งที่สำคัญสำหรับองค์กรเพื่อให้ตรงกับวิธีการนำกลับมาใช้ใหม่กับเป้าหมายทางธุรกิจเชิงกลยุทธ์ ประโยชน์ของการนำกลับมาใช้ใหม่เติบโตตามประสบการณ์ขององค์กรมากขึ้นที่ได้รับจากการนำกลับมาใช้ใหม่ แต่ค่าใช้จ่ายและปริมาณของทรัพยากรที่จำเป็นเพื่อนำกลับมาใช้ใหม่นั้นทำงานได้ดี การนำซอฟต์แวร์กลับมาใช้ใหม่มีขั้นตอนพื้นฐานสามขั้นตอน นั่นคือการกำหนดสาระสำคัญ การจัดเก็บ และการปรับมุมมองใหม่การกำหนดสาระสำคัญที่เกี่ยวข้องกับการออกแบบส่วนที่นำกลับมาใช้ใหม่ของซอฟต์แวร์ที่เริ่มจากสิ่งที่มีอยู่ในซอฟต์แวร์หรือจากการเริ่มต้นใหม่ การจัดเก็บข้อมูลเกี่ยวข้องกับการที่มีซอฟต์แวร์อยู่แล้วสำหรับที่จะใช้ต่อไปได้ แม้ว่าดูเหมือนเป็นปัญหาง่ายสามารถในการจัดเก็บข้อมูล อันที่จริงแล้วเป็นสิ่งที่ทำทายนมาก ปัญหาไม่ได้เพียงแค่วางซอฟต์แวร์ที่มีอยู่แล้วในชั้นที่เก็บรักษา ปัญหาคือติดตามได้อย่างถูกต้องและรายการเพื่อให้ผู้อื่นสามารถหาได้เมื่อต้องการที่จะใช้ เมื่อพบ การปรับ

มุมมองใหม่ หรือการนำกลับมาใช้ใหม่เป็นที่เข้าใจกับผู้พัฒนาซึ่งเป็นผู้ที่ต้องการใช้ในระบบก็จะเริ่มมีความสำคัญ ซอฟต์แวร์ที่มีความซับซ้อน และซอฟต์แวร์ที่พัฒนาสำหรับระบบโดยเฉพาะอย่างยิ่งภายใต้สถานการณ์เฉพาะระบบ อาจจะไม่ได้อีกต่อไป เป็นสิ่งที่ใช้ได้ สิ่งที่คุณเหมือนว่าจะเป็นสิ่งที่อยู่โดยทั่วไปเรียกว่า ลูกค้า ที่จริงอาจจะเป็นสิ่งที่คุณข้างแตกต่างกันขึ้นอยู่กับบริบทที่จะพัฒนา อาจจะเป็นเรื่องง่ายที่เพียงแค่สร้างให้มีสิ่งที่ต้องการ มากกว่าที่จะการลงทุนเวลาและพลังงานที่ใช้ในการสร้างความเข้าใจที่ดีของซอฟต์แวร์โดยที่ให้คนอื่นพัฒนาขึ้นมา ส่วนสำคัญของกลยุทธ์ที่นำกลับมาใช้ใหม่ที่ได้กล่าวถึงก่อนหน้านี้ คือ การสร้างผลตอบแทน การจูงใจ และการสนับสนุนขององค์กร เพื่อการนำกลับมาใช้ใหม่เพื่อช่วยทำให้คุ้มค่ามากกว่าการขึ้นมาเอง องค์กรสามารถใช้หนึ่งในสี่วิธีการที่จะนำกลับมาใช้ใหม่ วิธีการที่นำกลับมาใช้ใหม่เฉพาะกิจไม่ได้เป็นวิธีการจริง ๆ เป็นแค่จากมุมมองขององค์กร ด้วยวิธีนี้ บุคคลมีอิสระในการหาหรือพัฒนาการนำมาใช้ใหม่ด้วยตัวเอง แต่มีไม่มาก ถ้ามีรางวัลจากองค์กรเสนอการนำกลับมาใช้ใหม่ การจัดเก็บจะไม่เป็นปัญหาเพราะบุคคลที่ติดตาม และแจกจ่ายซอฟต์แวร์ของตัวเอง ที่เป็นเฉพาะกิจเช่น วิธีการขับเคลื่อนเป็นรายบุคคล ซึ่งเป็นเรื่องยากที่จะวัดผลประโยชน์ที่อาจเกิดขึ้นกับบริษัท วิธีการอื่นที่จะนำมาใช้ก็คือการอำนวยความสะดวกการนำกลับมาใช้ใหม่ ด้วยวิธีการนี้ นักพัฒนาไม่จำเป็นต้องปฏิบัติตามการนำกลับมาใช้ แต่จะมีกำลังใจที่จะทำเช่นนั้น องค์กรทำให้มีเครื่องมือและเทคนิคบางอย่างที่ช่วยให้การพัฒนาและการแบ่งปันที่นำกลับมาใช้ใหม่และพนักงานคนหนึ่งหรือมากกว่าอาจจะได้รับมอบหมายบทบาทในการประชาสัมพันธ์ และส่งเสริมการโปรแกรม มีส่วนน้อยจะทำในการติดตามคุณภาพและการใช้งานของการนำกลับมาใช้ใหม่ แต่โดยรวมการลงทุนขององค์กรมีขนาดเล็ก การนำกลับมาใช้ใหม่ที่มีการจัดการเป็นโครงสร้างมากขึ้น และมีราคาแพงมากขึ้น วิธีในการจัดการ การนำกลับมาใช้ใหม่ของซอฟต์แวร์ ด้วยการนำกลับมาใช้ใหม่มีการจัดการ การพัฒนา การแบ่งปัน และการยอมรับของการนำกลับมาใช้ใหม่ได้คำสั่งการไปแล้ว องค์กรกำหนดกระบวนการและนโยบายในการตรวจสอบการนำกลับมาใช้ใหม่มีความชำนาญและมีผลลัพธ์ที่สามารถวัดได้ องค์กรนอกจากนี้ยังกำหนดนโยบายและขั้นตอนในการตรวจสอบคุณภาพของการนำกลับมาใช้ใหม่ด้วย โดยเน้นสำคัญในสิ่งที่มีอยู่แล้วที่สามารถนำกลับมาใช้ใหม่จากแหล่งต่าง ๆ รวมทั้งจากคลังโปรแกรมมอรรถประโยชน์ที่มาพร้อมกับระบบปฏิบัติการ จากบริษัทที่ขายให้ จากชุมชนโอเพนซอร์ส จากที่เก็บภายใน จากการกำจัดสิ่งที่ไม่ได้บรรทัดเดิมที่มีอยู่ และอื่น ๆ วิธีการที่แพงที่สุดและครอบคลุมเพื่อนำกลับมาใช้ใหม่คือการนำกลับมาใช้ใหม่ที่ได้ออกแบบไว้แล้ว นอกจากนี้ที่จะนำกลับมาใช้ใหม่และการวัดประสิทธิภาพ วิธีการที่นำมาใช้ใหม่ได้รับการออกแบบใช้ขั้นตอนพิเศษของการออกแบบเพื่อนำกลับมาใช้ใหม่เป็นอย่างที่ถูกรออกแบบมาสำหรับการใช้งานที่เฉพาะเจาะจง ให้ความสำคัญมากขึ้น ในการพัฒนาการนำกลับมาใช้ใหม่กว่าในการหาสิ่งที่มีอยู่ที่จะเป็นอีกทางเลือกที่จะนำกลับมาใช้ใหม่ สำนักงานขององค์กรการนำกลับมาใช้ใหม่อาจจะจัดตั้งขึ้นเพื่อตรวจสอบและจัดการวิธีการโดยรวม ภายใต้วิธีการดังกล่าว ร้อยละ 90 ซอฟต์แวร์อาจจะนำกลับมาใช้ใหม่ในการใช้งานที่แตกต่างกัน วิธีการที่จะนำกลับมาใช้ใหม่มีข้อดีและ

ข้อเสีย ไม่มีวิธีการใดเป็นทางแก้ปัญหาก็จะแก้ปัญหาคำถามนำกลับมาใช้ใหม่สำหรับทุกองค์กรและทุกสถานการณ์ การนำกลับมาใช้ใหม่ที่ประสบความสำเร็จจะต้องมีความเข้าใจถึงวิธีการที่เหมาะสมสำหรับนำมาใช้ให้ตรงกับ เป้าหมายขององค์กรที่มีขนาดใหญ่และกลยุทธ์เช่นเดียวกับความเข้าใจของสังคมโลกและทางเทคนิคที่นำมาใช้ใหม่ ต้องพอเหมาะพอดี แม้ว่าการนำกลับมาใช้ใหม่มีคุณค่ากับองค์กรจำนวนมาก ก็อาจจะกลับกลายเป็นไม่มีคุณค่าใน การพัฒนาทั้งหมดในองค์กรใด ๆ เลย นักพัฒนามือใหม่ มีแนวโน้มที่จะนำรหัสกลับมาใช้ใหม่และส่วนประกอบมา กว่ากว่าเป็นนักพัฒนาที่มีประสบการณ์มากกว่า นักพัฒนามือใหม่มีไม่ชอบความเสี่ยงและไม่ต้องการที่จะทำให้เกิด ความผิดพลาด ดังนั้นจึงมีแนวโน้มที่จะนำรหัสที่มีอยู่แล้วมาใช้ใหม่ซึ่งได้รับการทดสอบแล้วและการตรวจสอบแล้ว นักพัฒนาที่มีประสบการณ์มากขึ้นมีแนวโน้มที่จะไว้วางใจทักษะการเขียนโปรแกรมของตัวเองมากกว่าที่ไว้วางใจ ทักษะของคนอื่น ๆ ดังนั้นจะชอบที่จะเขียนและทดสอบรหัสของตัวเอง ความแตกต่างในการนำกลับมาใช้ใหม่ทุก ประเภทที่แตกต่างกันของทีมงานพัฒนาเป็นเรื่องธรรมดา ทีมงานของโครงการชั่วคราวซึ่งจะอยู่ในช่วงเวลาสั้นเป็น มีแนวโน้มที่จะใช้การนำกลับมาใช้ใหม่กว่าการจัดตั้งทีมงานของโครงการอย่างถาวรการใช้ทรัพยากรที่เขียนไว้ก่อน หน้าซอฟต์แวร์ในการใช้งานใหม่