

บทที่ 2

สรุปสาระสำคัญจากเอกสารที่เกี่ยวข้อง

2.1 ระบบสารสนเทศ

ระบบสารสนเทศ (Information System) หมายถึง ระบบที่มีการนำคอมพิวเตอร์ มาช่วยในการรวบรวม จัดเก็บ หรือจัดการกับข้อมูลข่าวสารเพื่อให้ข้อมูลนั้นกลายเป็นสารสนเทศที่ดี เป็นระบบพื้นฐานของการทำงานต่างๆ ในรูปแบบของการเก็บ (input) การจัดการ (processing) เผยแพร่ (output) และมีส่วนเก็บข้อมูล (storage) ระบบสารสนเทศเป็นการรวมกลุ่มของฮาร์ดแวร์, ซอฟต์แวร์, มนุษย์, กระบวนการ, ฐานข้อมูล และอุปกรณ์ต่างๆ เพื่อจัดเก็บข้อมูลสารสนเทศให้องค์กรบรรลุตามเป้าหมายที่ตั้งไว้

ระบบสารสนเทศประกอบด้วยองค์ประกอบดังนี้

- 1) ข้อมูล (Data) หมายถึง ค่าของความจริงที่ปรากฏขึ้น โดยค่าความจริงที่ได้จะนำมาจัดการปรับแต่งหรือประมวลผลเพื่อให้ได้สารสนเทศที่ต้องการ
- 2) สารสนเทศ (Information) คือ กลุ่มของข้อมูลที่ถูกตามกฎเกณฑ์ตามหลักความสัมพันธ์ เพื่อให้ข้อมูลเหล่านั้นมีประโยชน์และมีความหมายมากขึ้น
- 3) การจัดการ (Management) คือ การบริหารอย่างเป็นระบบ เป็นการกำหนดเป้าหมายและทิศทางการจัดการขององค์กรนั้น ซึ่งต้องมีการวางแผน กำหนดการ และจัดการทรัพยากรภายในองค์กร เพื่อให้บรรลุถึงวัตถุประสงค์ขององค์กรนั้นๆ

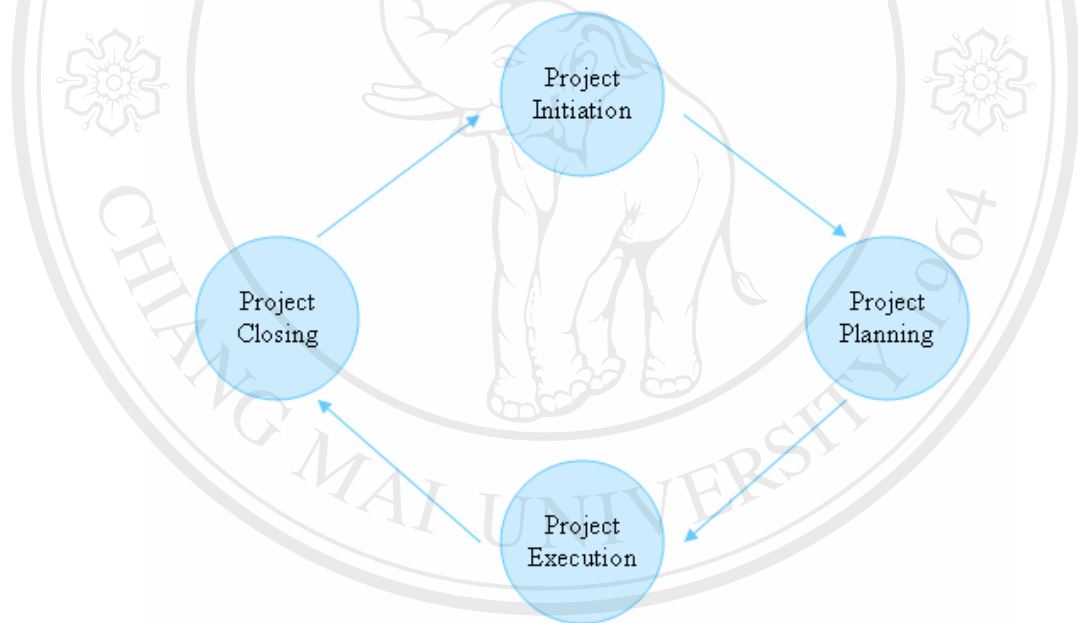
ระบบสารสนเทศจะสามารถทำงานได้ตรงตามความต้องการ รวดเร็ว และถูกต้องได้นั้น จำเป็นต้องมีการสร้างระบบสารสนเทศที่มีประสิทธิภาพ โดยใช้กระบวนการพัฒนาระบบที่เรียกว่า ขั้นตอนการพัฒนาซอฟต์แวร์ (Systems Development Life Cycle หรือ SDLC) ดังนี้

1. วิเคราะห์และกำหนดความต้องการของระบบงาน (System Analysis and Specification)
2. ออกแบบขั้นตอนการแก้ไขปัญหา (System Design)
3. เขียนชุดคำสั่ง (Program Coding)
4. ทดสอบการทำงานของระบบงาน (System or Program Testing)
5. ใช้งานและบำรุงรักษาระบบ (System Implementation and Maintenance)
6. จัดทำเอกสารประกอบระบบ (Documentation)

2.2 การบริหารโครงการซอฟต์แวร์

กิตติ ภัคดีวัฒนะกุล [2] ได้กล่าวว่า การบริหารโครงการ หมายถึง การประยุกต์ใช้องค์ความรู้ ทักษะ เครื่องมือ และเทคนิค เพื่อดำเนินกิจกรรมตามความต้องการของโครงการให้บรรลุวัตถุประสงค์ที่กำหนดไว้ โครงการในที่นี้คือ โครงการผลิตซอฟต์แวร์ ซึ่งจำเป็นต้องอาศัยการบริหารโครงการที่มีประสิทธิภาพ เนื่องจากโครงการเป็นงานที่ต้องดำเนินการภายใต้ข้อจำกัดหลายอย่าง ไม่ว่าจะเป็นแรงงาน ต้นทุนและเวลา หากการบริหารโครงการบกพร่อง จะส่งผลเสียต่อโครงการอย่างมาก กล่าวคือ อาจทำให้การส่งมอบซอฟต์แวร์ไม่ทันเวลา ใช้ต้นทุนเกินที่คาดการณ์ไว้ และซอฟต์แวร์ไม่มีคุณภาพ (ไม่ตรงตามข้อกำหนดความต้องการ)

โครงการทุกประเภท เป็นสิ่งที่จัดตั้งขึ้นเพียงชั่วคราวเท่านั้น ตลอดช่วงอายุของโครงการจะมีทั้งหมด 4 ระยะ ได้แก่



รูปที่ 2-1 แสดงวงจรชีวิตโครงการ

1) ระยะเวลาเริ่มโครงการ (Project Initiation) หลังจากที่ยอดโครงการได้คัดเลือกโครงการที่เหมาะสมที่สุดแล้ว ทีมงานที่รับผิดชอบต้องเริ่มต้นโครงการด้วยการกำหนดขอบเขตและขนาดของโครงการ รวมทั้งกำหนดกิจกรรมหรืองานที่จะต้องทำในแต่ละขั้นตอนของการผลิตซอฟต์แวร์อีกด้วย

2) ระยะเวลาวางแผนโครงการ (Project Planning) เป็นระยะที่ผู้บริหารโครงการต้องมีการกำหนดกิจกรรมในแต่ละขั้นตอนของการผลิตซอฟต์แวร์อย่างชัดเจน รวมถึงการประมาณการใช้

ทรัพยากรต่างๆ ไม่ว่าจะเป็นเวลา ต้นทุนและแรงงาน นอกจากนี้ บริหารโครงการยังต้องจัดการความเสี่ยง ตลอดจนกิจกรรมอื่นๆ อีกมากมาย

3) ระยะเวลาดำเนินโครงการ (Project Execution) เป็นระยะที่ทีมงานดำเนินกิจกรรม การผลิตซอฟต์แวร์ตามแผนการดำเนินงานที่จัดไว้ ดังนั้นผู้บริหารโครงการจะต้องมีการติดตามการทำงาน ดูแล สั่งการ และควบคุมการทำงานของลูกทีมให้ดำเนินงานตามแผนงานที่กำหนดไว้ นอกจากนี้ผู้บริหารโครงการจะต้องคอยติดตามการเปลี่ยนแปลงที่เกิดขึ้นในระหว่างการดำเนินงาน เพื่อปรับปรุงแก้ไขรายละเอียดในแผนงานให้เป็นปัจจุบันที่สุด อีกทั้งจะต้องหาแนวทางในการแก้ปัญหาที่เกิดขึ้นเนื่องจากผลกระทบของการเปลี่ยนแปลง

4) ระยะเวลาปิดโครงการ (Project Closing) เป็นระยะสุดท้ายของการบริหารโครงการ เป็นการดำเนินงานหลังจากติดตั้งระบบซอฟต์แวร์เพื่อใช้งานเสร็จสิ้นแล้ว

2.3 เทคโนโลยีรักษาความปลอดภัย SSL

การรักษาความปลอดภัยให้กับข้อมูลจึงเป็นสิ่งที่หลีกเลี่ยงไม่ได้ ข่าวสารที่จะแลกเปลี่ยนกันนั้นมีความปลอดภัย ป้องกันไม่ให้เกิดความผิดพลาดอันส่งผลให้การติดต่อสื่อสารล้มเหลว นั้นเราจะใช้หลักการรักษาความปลอดภัยของข้อมูลในรูปแบบ RSA โดยใช้โปรโตคอล SSL ซึ่งเป็นหลักการรักษาความปลอดภัยที่ใช้หลักทางคณิตศาสตร์ที่มีรูปแบบการคำนวณที่เข้าใจง่าย อีกทั้งยังมีความปลอดภัยสูง

SSL เป็นโปรโตคอลจัดการความปลอดภัยในระบบอินเทอร์เน็ตที่ใช้ในการสื่อสารข้อมูลกันระหว่างไคลเอนต์(client) กับเซิร์ฟเวอร์(server) ปกติแล้วข้อมูลที่ส่งไปหากันจะไม่มี การเข้ารหัสข้อมูลแต่อย่างใด ทำให้การดักจับข้อมูลเป็นไปได้โดยง่าย แต่ถ้าเป็นระบบที่ใช้ SSL ข้อมูลจากไคลเอนต์ที่จะส่งไปที่เซิร์ฟเวอร์จะถูกเข้ารหัสก่อนที่จะส่งไปที่เซิร์ฟเวอร์ ทำให้ข้อมูลที่รับส่งกันมีความปลอดภัยมากยิ่งขึ้น

SSL นั้นได้รับการยอมรับอย่างกว้างขวาง ในการใช้สำหรับตรวจสอบและเข้ารหัสลับการติดต่อสื่อสารระหว่างไคลเอนต์ และเซิร์ฟเวอร์ หน้าที่ของ SSL จะแบ่งออกเป็น 3 ส่วนใหญ่ๆคือ

1) การตรวจสอบเซิร์ฟเวอร์ว่าเป็นตัวจริง

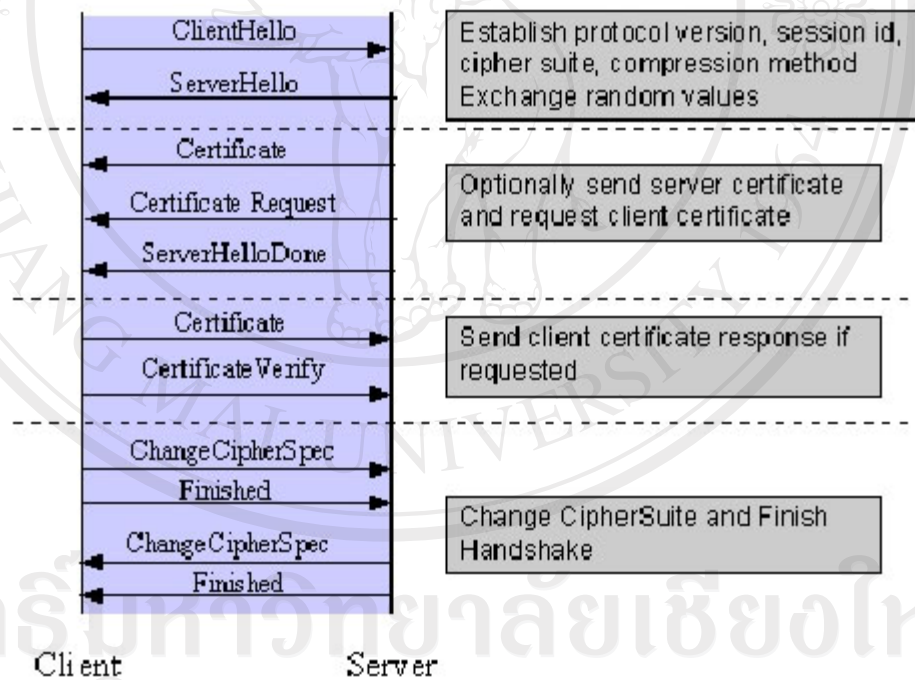
ตัวโปรแกรมไคลเอนต์ที่มีขีดความสามารถ ในการสื่อสารแบบ SSL จะสามารถตรวจสอบเครื่องเซิร์ฟเวอร์ที่ตนกำลังจะไปเชื่อมต่อได้ว่า เซิร์ฟเวอร์นั้นเป็น เซิร์ฟเวอร์ตัวจริงหรือไม่ โดยใช้เทคนิคการเข้ารหัสแบบ public key ในการตรวจสอบใบรับรอง (certificate) และ public ID ของเซิร์ฟเวอร์นั้น (โดยที่มืองค์กรที่ไคลเอนต์เชื่อถือเป็นผู้ออกใบรับรองและ public ID ให้แก่เซิร์ฟเวอร์นั้น)

2) การตรวจสอบว่าไคลเอนต์เป็นความจริง

เซิร์ฟเวอร์ที่มีขีดความสามารถในการสื่อสารแบบ SSL จะใช้เทคนิคเช่นเดียวกับในหัวข้อที่แล้วในการตรวจสอบไคลเอนต์หรือผู้ใช้งานว่าเป็นความจริงหรือไม่ โดยจะตรวจสอบใบรับรองและ public ID (ที่มีองค์กรที่เซิร์ฟเวอร์เชื่อถือเป็นผู้ออกให้) ของไคลเอนต์หรือผู้ใช้นั้น

3) การเข้ารหัสลับการเชื่อมต่อ

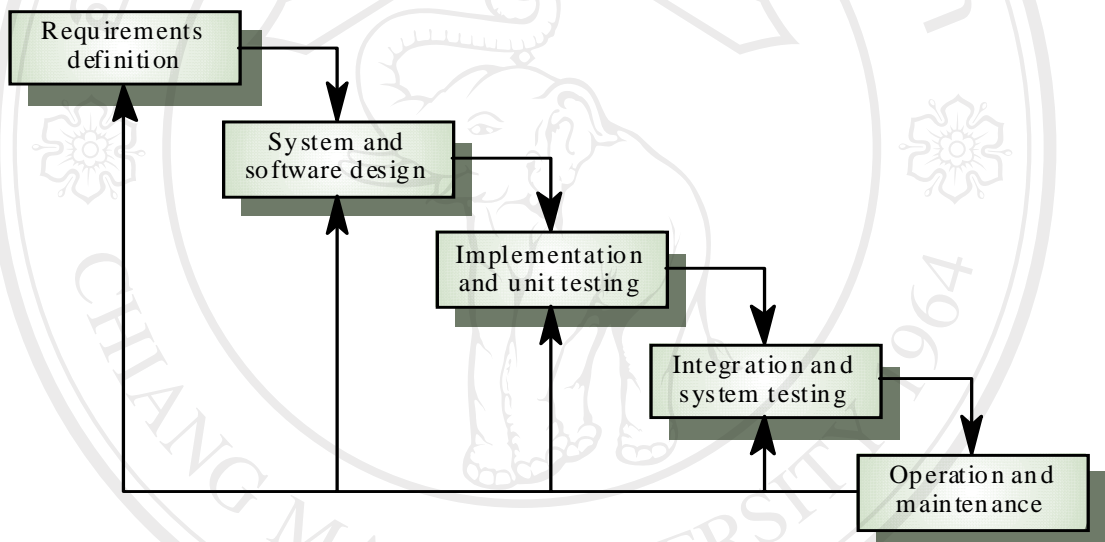
ในกรณีนี้ ข้อมูลทั้งหมดที่ถูกส่งระหว่างไคลเอนต์และเซิร์ฟเวอร์จะถูกเข้ารหัสลับ โดยโปรแกรมที่ส่งข้อมูลเป็นผู้เข้ารหัสและโปรแกรมที่รับข้อมูลเป็นผู้ถอดรหัส (โดยใช้วิธี public key) นอกจากการเข้ารหัสลับในลักษณะนี้แล้ว SSL ยังสามารถปกป้องความถูกต้องสมบูรณ์ของข้อมูลได้อีกด้วย กล่าวคือ ตัวโปรแกรมรับข้อมูลจะทราบได้หากข้อมูลถูกเปลี่ยนแปลงไปในขณะกำลังเดินทางจากผู้ส่งไปยังผู้รับ



รูปที่ 2-2 กระบวนการเริ่มต้นการติดต่อสื่อสารของโปรโตคอล SSL

2.4 แบบจำลองน้ำตก

แบบจำลองน้ำตก (Waterfall Model) เป็นแบบจำลองที่ประกอบไปด้วยขั้นตอนการดำเนินงานที่เรียงต่อเนื่องกันเป็นลำดับ ขั้นตอนต่อไปจะเริ่มดำเนินการได้จะต้องรอให้ขั้นตอนก่อนหน้าเสร็จสิ้นก่อน โดยขั้นตอนพื้นฐานในการดำเนินงานผลิตซอฟต์แวร์ในแบบ Waterfall Model มี 5 ขั้นตอน ได้แก่ การกำหนดความต้องการ (Requirement Definition) การออกแบบซอฟต์แวร์และระบบ (System and Software Design) การลงมือทำและทดสอบระดับหน่วย (Implementation and Unit Testing) การประสานระบบและทดสอบระบบ (Integration and System Testing) การนำไปใช้และบำรุงรักษา (Operation and Maintenance) ดังรูป



รูปที่ 2-3 แสดงแบบจำลองกระบวนการผลิตซอฟต์แวร์แบบ Waterfall Model

เนื่องจากต้องรอให้ขั้นตอนก่อนหน้าดำเนินการเสร็จก่อน จึงทำให้แบบจำลองน้ำตก มีข้อเสียคือ ไม่สามารถรองรับความต้องการของลูกค้าที่เปลี่ยนแปลงอย่างรวดเร็วได้ อย่างไรก็ตาม ข้อดีของแบบจำลองชนิดนี้ก็คือ มีการสร้างเอกสารในทุกๆ ขั้นตอนหรือทุกระยะ

2.5 มาตรฐานคุณภาพซอฟต์แวร์ไทย

มาตรฐานคุณภาพซอฟต์แวร์ไทย (TQS) หรือ Thai Quality Software คือ เกณฑ์คุณภาพของการผลิตซอฟต์แวร์ โดยอ้างอิงจากมาตรฐาน ISO / IEC 12207 ซึ่งเป็นมาตรฐานนานาชาติ ในส่วนที่เกี่ยวกับการควบคุมคุณภาพในกระบวนการผลิตซอฟต์แวร์ โดยมีจุดมุ่งหมายเพื่อยกระดับ

คุณภาพและปรับปรุงความสามารถในการพัฒนาซอฟต์แวร์ เพิ่มโอกาสและขีดความสามารถทางการแข่งขันทางการตลาดทั้งในและต่างประเทศ

ข้อดีของการพัฒนาระบบตามมาตรฐาน TQS

ปัจจุบันมาตรฐาน TQS ได้มีบทบาทสำคัญต่อผู้ผลิตซอฟต์แวร์ในประเทศไทย ข้อดีของการใช้ TQS สามารถกล่าวโดยสังเขปได้ดังนี้

- * ได้รับความรับรองคุณภาพ TQS สำหรับการพัฒนาโครงการซอฟต์แวร์
- * เพิ่มโอกาสในการขยายฐานลูกค้าและสร้างรายได้ของธุรกิจ
- * เพิ่มโอกาสได้รับการสนับสนุนจากแหล่งสินเชื่อหรือแหล่งเงินทุนมากขึ้น
- * สร้างความมั่นใจให้กับลูกค้า

หากจะเปรียบเทียบกับมาตรฐานอื่นๆ TQS ก็มีวิธีการตรวจทานขั้นตอนคุณภาพเช่นเดียวกัน กล่าวคือ การตรวจหาเอกสารอ้างอิง (Traceability Documents) เอกสารอ้างอิงดังกล่าวเป็นการยืนยันว่าองค์กรได้ดำเนินการตามขั้นตอนที่ระบุไว้ว่า จะทำตามมาตรฐานคุณภาพขององค์กรเป็นหลักตามขอบเขตที่ระบุไว้ในกระบวนการมาตรฐาน ซึ่งผู้ตรวจประเมิน (Assessor) ผู้เชี่ยวชาญและชำนาญขั้นตอนมาตรฐานต่างๆ จะเป็นผู้ชี้แนะว่าเอกสารอ้างอิงที่ท่านมีครบตามกระบวนการและขั้นตอนที่องค์กรของท่านระบุว่าต้องมีหรือไม่ โดยผู้ตรวจประเมินจะไม่ยืนยันว่าขั้นตอนหรือเอกสารที่ท่านมีนั้นผิดหรือถูก เนื่องจากความแตกต่างขององค์กรและความจำเป็นทางธุรกิจที่ต่างกันออกไป