

บทที่ 3

ระเบียบวิธีการศึกษา

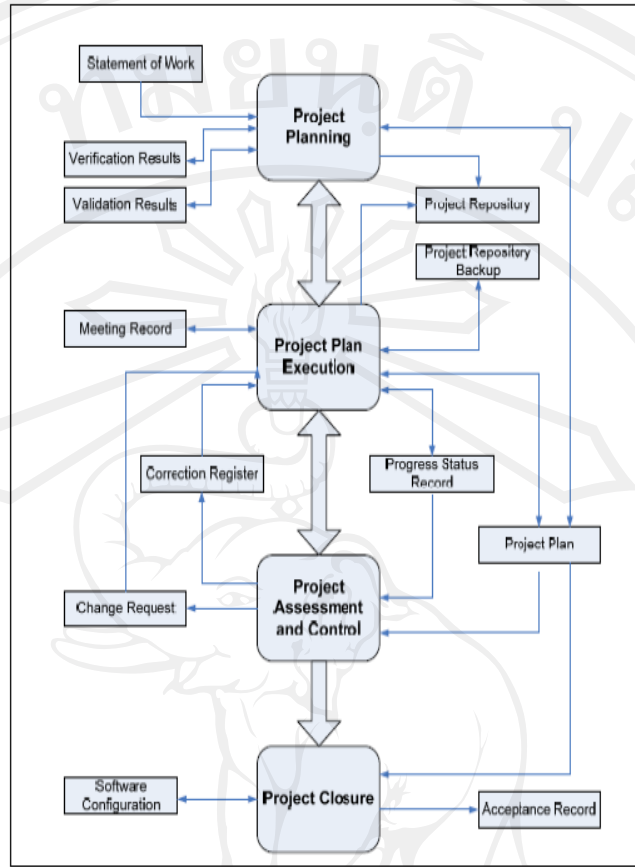
3.1 แนวทางการค้นคว้าและพัฒนา

เพื่อให้การค้นคว้าระบบให้บริการรับส่งสินค้าแบบส่งถึงที่ผ่านทางเว็บเบราว์เซอร์บนโทรศัพท์เคลื่อนที่เป็นอย่างมีประสิทธิภาพ ผู้ค้นคว้าจึงเลือกพัฒนาซอฟต์แวร์ตามหลักการมาตรฐานอุตสาหกรรมซอฟต์แวร์ ISO 29110 VSE มีกระบวนการในการพัฒนาซอฟต์แวร์ที่เป็นระบบและได้รับการรับรองให้เป็นมาตรฐานสากล โดยจะเป็นการเริ่มต้นในเชิงกิจกรรมของการปรับปรุงกระบวนการ หรือ SPI (Software Process Improvement) ได้ให้ความสำคัญในกระบวนการที่จะต้องทำการปรับปรุงให้เป็นระบบอยู่ 2 กระบวนการหลักด้วยกัน คือ กระบวนการด้านการบริหารโครงการ (Project Management) ดังรูปที่ 3.1 และกระบวนการด้านการสร้างซอฟต์แวร์ (Software Implementation) ดังรูปที่ 3.2 ซึ่งจะประกอบไปด้วยกระบวนการย่อยภายในอีกลำดับหนึ่ง โดยทั้งสองกระบวนการได้ถูกออกแบบให้เหมาะสมกับองค์กรขนาดเล็ก จึงมีความเหมาะสมในการประยุกต์ใช้ได้ทันที

3.2 กระบวนการบริหารโครงการ (Project Management Process)

ประกอบด้วยกระบวนการย่อย 4 กระบวนการดังนี้

3.2.1 การวางแผนโครงการ (Project Planning) เป็นขั้นตอนแรกที่มีความเกี่ยวข้องกับความเป็นมาของโครงการ โดยมีแนวความคิดจากแหล่งข้อมูลต่างๆ ที่ต้องการให้มีโครงการเกิดขึ้น และเป็นเรื่องเกี่ยวข้องกับการเปลี่ยนปัจจัยทางการบริหาร การวางแผนโครงการจะเริ่มตั้งแต่การกำหนดวัตถุประสงค์ของโครงการ วิธีการจัดการและวิธีทำงานต่างๆ เพื่อให้บรรลุตามเป้าหมาย โดยการแยกโครงการออกเป็นกลุ่มงานย่อย และดำเนินการตามแผนที่วางไว้ รวมไปถึงการกำหนดเวลาของแต่ละงาน ตลอดจนความสัมพันธ์ระหว่างงาน ตามที่ระบุในเอกสาร Project Planning ที่แสดงในภาคผนวก



รูปที่ 3.1 กระบวนการบริหารโครงการ (Project Management Process) ของ ISO 29110 VSE

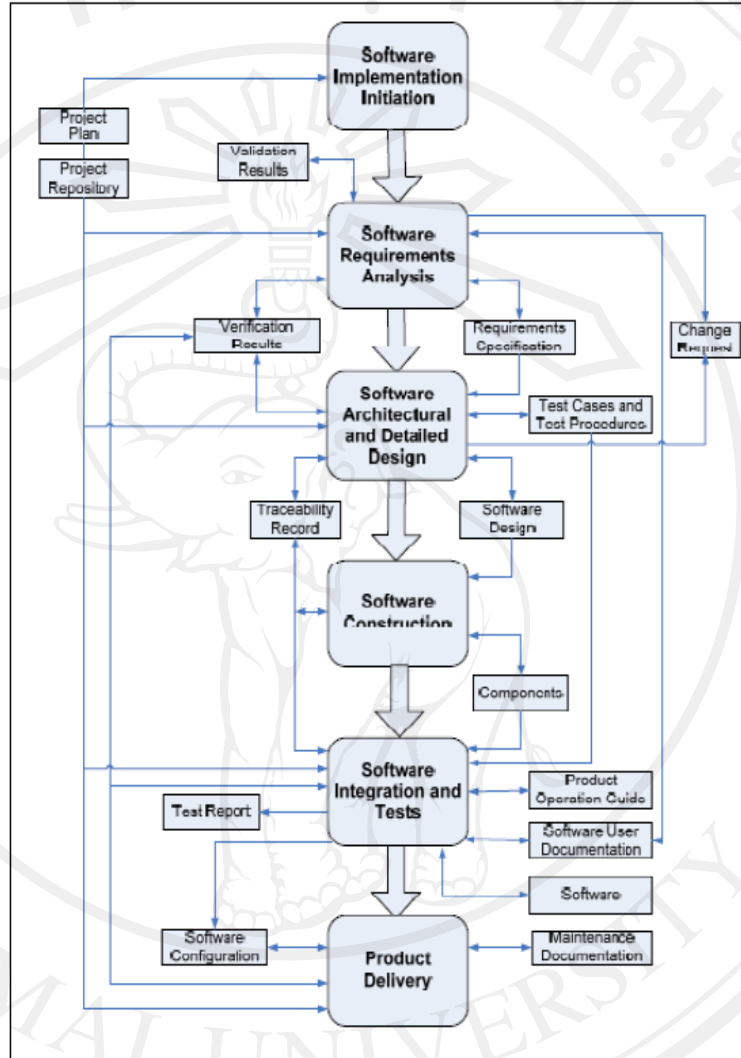
3.2.2 การติดตามแผนโครงการ (Project Planning Execution) การเก็บรวบรวมข้อมูลการปฏิบัติงานตามแผนงานที่กำหนดไว้ เพื่อตัดสินใจแก้ไขปรับปรุงวิธีการปฏิบัติให้ผลงานเป็นไปตามแผนงานอย่างมีประสิทธิภาพมากที่สุด โดยมุ่งที่จะตอบคำถามหลักว่า ในการดำเนินงานนั้นได้รับทรัพยากรครบถ้วนหรือไม่ ได้ปฏิบัติงานตามแผนงานที่กำหนดหรือไม่ ได้ผลตรงตามที่กำหนดหรือไม่ ซึ่งข้อมูลจากการติดตามจะเป็นประโยชน์อย่างยิ่งต่อการเร่งรัดให้งาน/โครงการ ดำเนินไปตรงตามเป้าหมายของแผนงาน และแล้วเสร็จภายในกำหนดเวลาตามที่ระบุในเอกสาร Progress Status Record และ Meeting Record ดังที่แสดงในภาคผนวก

3.2.3 การตรวจสอบและควบคุมโครงการ (Project Assessment and Control) การควบคุมโครงการเป็นขั้นตอนสำคัญขั้นตอนหนึ่ง ซึ่งเป็นการบังคับให้ดำเนินกิจกรรมต่างๆ เป็นไปตามกิจกรรมที่วางไว้ระหว่างดำเนินโครงการ เพื่อให้สามารถดำเนินกิจกรรมได้ตามแผนที่วางไว้

3.2.4 การปิดโครงการ (Project Closure) ตามที่ระบุในเอกสาร Acceptance Record ที่แสดงในภาคผนวก

3.3 กระบวนการพัฒนาซอฟต์แวร์ (Software Implementation Process)

ประกอบด้วยกระบวนการย่อย 6 กระบวนการดังนี้



รูปที่ 3.2 กระบวนการพัฒนาซอฟต์แวร์ (Software Implementation Process)

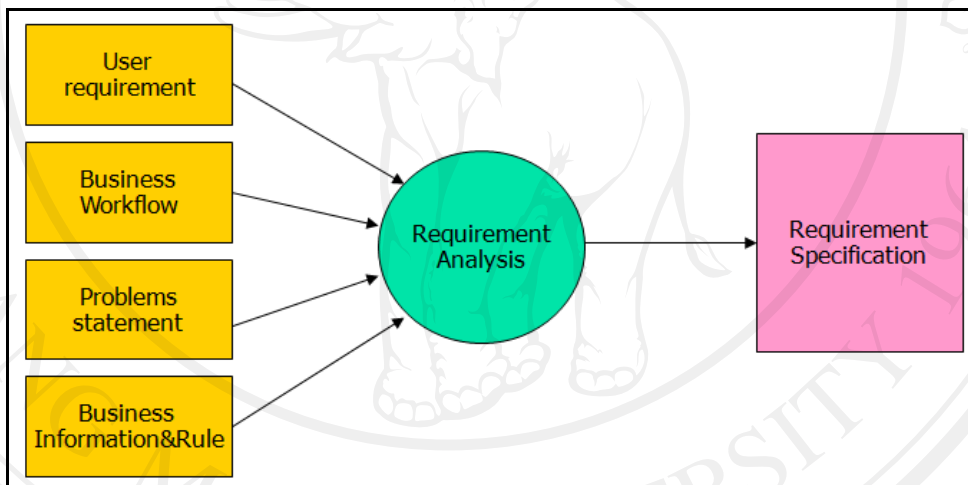
ของ ISO 29110 VSE

3.3.1 การเริ่มต้นการพัฒนาซอฟต์แวร์ (Software Implementation Initiation) สิ่งสำคัญในการเริ่มต้นการพัฒนาซอฟต์แวร์ ก็คือการศึกษาวิสัยทัศน์และขอบเขตของงาน เพื่อให้เราสามารถรวบรวมความต้องการในเชิงธุรกิจและของผู้ใช้งานได้ จากนั้นวิเคราะห์ระบบซอฟต์แวร์ที่ใช้อยู่ในปัจจุบัน รวมไปถึงการประเมินความเสี่ยง สิ่งที่ได้รับทราบถึงความต้องการที่เฉพาะเจาะจงตามจุดประสงค์ของโครงการ รวมไปถึงอุปสรรคปัญหาของธุรกิจ ขอบเขตของการพัฒนา

ซอฟต์แวร์ และการประเมินความเสี่ยงในด้านต่าง ๆ ตามที่ระบุในเอกสาร Project Planning ที่แสดงในภาคผนวก

3.3.2 การวิเคราะห์ความต้องการซอฟต์แวร์ (Software Requirements Analysis) คือกระบวนการวิเคราะห์เพื่อหาข้อกำหนดความต้องการของผู้ใช้ โดยจะต้องอาศัยข้อมูลในด้านต่างๆ ที่ได้รับมาจากผู้ใช้และองค์กรของผู้ใช้เพื่อทำการวิเคราะห์ ดังรูปที่ 3.3 โดยการศึกษาครั้งนี้ได้แบ่งแหล่งข้อมูลเพื่อนำมาวิเคราะห์หาความต้องการดังนี้

- ความต้องการของผู้ใช้ (User Requirement)
- กระบวนการทำงานขององค์กร (Business Workflow)
- ประเด็นปัญหาที่พบ (Problems Statement)
- ข้อมูลและกฎระเบียบขององค์กร (Business Information & Rule)



รูปที่ 3.3 แหล่งข้อมูลเพื่อนำมาวิเคราะห์หาความต้องการระบบ

กระบวนการวิเคราะห์ความต้องการมีขั้นตอนดังต่อไปนี้

- (1) เก็บรวบรวมข้อมูลที่เป็นข้อเท็จจริงต่างๆ จากผู้ที่เกี่ยวข้องกับการใช้งานระบบ อันได้แก่ตัวแทนผู้บริหาร และผู้ประกอบการ
- (2) วิเคราะห์เพื่อระบุถึงความต้องการต่างๆ
- (3) คัดเลือกส่วนที่เป็นสาระสำคัญและอยู่ในขอบเขตการพัฒนา
- (4) จัดจำแนกและจัดโครงสร้างของความต้องการ
- (5) จัดลำดับความสำคัญและตกลงเจรจา
- (6) ตรวจสอบความถูกต้อง
- (7) จัดทำเอกสาร Requirement specification ที่แสดงในภาคผนวก

3.3.3 สถาปัตยกรรมซอฟต์แวร์และรายละเอียดการออกแบบ (Software Architectural and Detailed Design) หลังจากการวิเคราะห์ความต้องการของระบบเรียบร้อยแล้ว ขั้นตอนต่อไปคือการออกแบบระบบ ในขั้นตอนนี้จะมีกิจกรรมที่สำคัญประกอบด้วย การทบทวนความต้องการทั้งในด้านสารสนเทศและหน้าที่ของระบบ การพัฒนารูปแบบของระบบใหม่ ซึ่งประกอบด้วยรายละเอียดเกี่ยวกับผลลัพธ์ข้อมูลนำเข้า การประมวลผล หน่วยจัดเก็บข้อมูล ซึ่งการศึกษาครั้งนี้ได้แบ่งการออกแบบเป็น 5 ส่วน ตามที่ระบุในเอกสาร Software Design ที่แสดงในภาคผนวก ดังนี้

- (1) การออกแบบสถาปัตยกรรมของระบบ (System Architecture)
- (2) การออกแบบรายละเอียดของระบบ (Detailed Design) โดยออกแบบดังนี้
 - การออกแบบระบบในระดับ Software System เป็นการแสดงกลไกที่ใช้อธิบายเพื่อให้เข้าใจตรงกันว่าระบบควรมีบริการอะไรบ้างเพื่อให้ผู้ใช้บรรลุวัตถุประสงค์หลังจากใช้ระบบ ตามการวิเคราะห์ความต้องการต่างๆ ที่ได้จาก System Requirement เพื่อช่วยเหลือให้บุคคลที่เกี่ยวข้องกับระบบ เข้าใจระบบได้ โดยเก็บคำอธิบายการปฏิสัมพันธ์กับระบบเอาไว้ใน Use Case Diagram
 - การออกแบบระบบในระดับ Subsystem หรือ Package ซึ่งเป็นการอธิบายรายละเอียดของระบบย่อยต่างๆ ควรมีบริการอะไรบ้าง ตามการวิเคราะห์จากผังงานระบบ (System Flowcharts) เพื่อช่วยเหลือให้บุคคลที่เกี่ยวข้องกับระบบ เข้าใจระบบได้ โดยเก็บคำอธิบายการปฏิสัมพันธ์กับระบบเอาไว้ใน Use Case Diagram
- (3) การออกแบบฐานข้อมูล เพื่อใช้เก็บรวบรวมข้อมูลก่อนนำมาแสดงผล
- (4) การออกแบบส่วนผู้ใช้งาน (User Interface) มีวัตถุประสงค์หลักเพื่อการรับคำสั่งจากผู้ใช้และแสดงผล โดยมีหลักการออกแบบส่วนผู้ใช้งานดังนี้
 - มีรูปแบบที่ต้องรองรับการแสดงผลในเว็บเบราว์เซอร์บนโทรศัพท์เคลื่อนที่
 - ความเรียบง่าย ได้แก่ มีรูปแบบที่เรียบง่าย ไม่ซับซ้อน และใช้งานได้สะดวก ไม่มีกราฟิกหรือตัวอักษรที่เคลื่อนไหวอยู่ตลอดเวลา ชนิดและสีของตัวอักษรไม่มากจนเกินไปทำให้วุ่นวาย
 - ความสม่ำเสมอ ได้แก่ ใช้รูปแบบเดียวกันตลอดทั้งระบบ

3.3.4 การสร้างซอฟต์แวร์ (Software Construction) เป็นขั้นตอนหลังจากการยอมรับแนวคิดที่ได้ออกแบบระบบ และประกอบเข้าด้วยกัน เพื่อให้ออกมาเป็นซอฟต์แวร์ที่สามารถใช้งานได้ ตรงตามข้อกำหนดที่ระบุไว้ ในขั้นตอนนี้ผู้ศึกษาค้นคว้า ทำการเริ่มเขียนและทดสอบโปรแกรมว่า ทำงานถูกต้องหรือไม่ ต้องมีการทดสอบกับข้อมูลจริงที่เลือกแล้ว ถ้าทุกอย่างเรียบร้อย จะได้โปรแกรมที่พร้อมนำไปใช้งานจริงต่อไป ตามที่ระบุในเอกสาร Test Cases and Test Procedures ที่แสดงในภาคผนวก

3.3.5 การบูรณาการและการทดสอบซอฟต์แวร์ (Software Integration and Tests) เป็นกระบวนการเพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามี ความถูกต้อง, ความสมบูรณ์, ปลอดภัย, และมีคุณภาพที่ดี การทดสอบ เป็นกระบวนการทดลองใช้ซอฟต์แวร์อย่างมีแนวทาง โดยใช้ความรู้ทางด้านเทคนิค เพื่อให้สามารถระบุหรือค้นหาความผิดพลาด (Error) ของซอฟต์แวร์ที่อาจจะซ่อนอยู่ให้ปรากฏออกมา และสามารถระบุถึงแนวทางการเกิดปัญหา พร้อมสมมติฐานของความผิดพลาดที่เกิดขึ้นได้ ซึ่งการศึกษาค้นคว้าครั้งนี้ได้แบ่งขั้นตอนการทดสอบดังนี้

- วิเคราะห์แผนการทดสอบจากความต้องการผู้ใช้
- จัดทำแผนงานการทดสอบ
- จัดทำแนวทางการทดสอบ
- ทดสอบจริง
- บันทึกผลการทดสอบพร้อมระบุข้อผิดพลาด
- ทำการแก้ไขปรับปรุง
- ทดสอบผลจากการแก้ไข

การศึกษาค้นคว้าครั้งนี้ได้แบ่งวิธีการทดสอบดังนี้

- (1) Unit Testing เป็นการทดสอบส่วนย่อยของระบบแบบแยกจากกัน คำว่า “ส่วนย่อย” สามารถสื่อความหมายได้หลายแบบ เช่น Functions, Procedures, หรือ Methods ตามความเหมาะสมของงาน
- (2) Integration Testing เป็นการทดสอบจุดประสงค์ เพื่อตรวจสอบความถูกต้องของ Function การทำงานต่างๆ เมื่อมีการ Integrate Unit / Module เข้าร่วมกัน โดยจะให้ความสำคัญในส่วนของการ Interface ระหว่างกันว่าสามารถใช้งานร่วมกันได้หรือไม่

(3) User Acceptance Testing เป็นการทดสอบโดยผู้ใช้งาน (End User) คือกลุ่มผู้บริหาร และกลุ่มผู้ประกอบการ เพื่อดูว่าซอฟต์แวร์ทำงานถูกต้องตามต้องการหรือไม่

3.3.6 การส่งมอบผลิตภัณฑ์ (Product Delivery) ประกอบด้วย ขั้นตอนการปรับเปลี่ยนระบบ การจัดทำเอกสารคู่มือผู้ใช้งาน ตลอดจนแผนการบำรุงรักษาระบบ