

บทที่ 3

ระเบียบวิธีการศึกษา

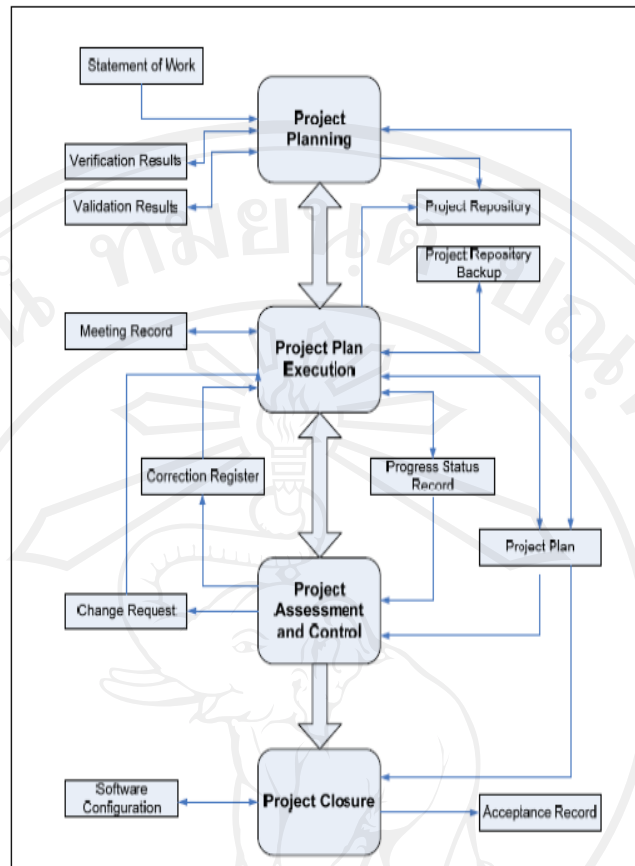
3.1 แนวทางการค้นคว้าและพัฒนา

การศึกษาค้นคว้าและการพัฒนาระบบแปลหนังสืออิเล็กทรอนิกส์โดยโปรแกรมแปลของ ไมโครซอฟต์บนโทรศัพท์เคลื่อนที่แบบสมาร์ต: ข้อความภาษาอังกฤษเป็นข้อความภาษาอื่น 36 ภาษา พร้อมการสังเคราะห์เสียงพูดเพื่อให้การค้นคว้าเป็นไปอย่างมีระเบียบหลักการ ผู้ค้นคว้าจึงเลือกพัฒนาซอฟต์แวร์ตามหลักการมาตรฐานอุตสาหกรรมซอฟต์แวร์ ISO 29110 VSE มีกระบวนการในการพัฒนาซอฟต์แวร์ที่เป็นระบบและเข้าสู่กระบวนการสากล โดยจะเป็นการเริ่มต้นในเชิงกิจกรรมของการปรับปรุงกระบวนการหรือ SPI (Software Process Improvement) ได้ให้ความสำคัญในกระบวนการที่จะต้องทำการปรับปรุงให้เป็นระบบและเป็นสากล 2 กระบวนการหลักคือกระบวนการด้านการบริหารโครงการ (Project Management) ดังรูปที่ 3.1 และกระบวนการด้านการสร้างซอฟต์แวร์ (Software Implementation) ดังรูปที่ 3.2 ซึ่งจะประกอบด้วยกระบวนการย่อยๆ ภายในอีกทั้งสองกระบวนการได้ถูกออกแบบให้เหมาะสมกับองค์กรขนาดเล็กจึงมีความเหมาะสมในการประยุกต์ใช้ได้ทันที

3.2 กระบวนการบริหารโครงการ (Project Management Process)

ประกอบด้วยกระบวนการย่อย 4 กระบวนการดังนี้

3.2.1 การวางแผนโครงการ (Project Planning) เป็นขั้นตอนแรกที่มีความเกี่ยวข้องกับความเป็นมาของโครงการ โดยมีแนวความคิดจากแหล่งต่างๆ ที่ต้องการให้มีโครงการเกิดขึ้นและเป็นเรื่องเกี่ยวข้องกับการเปลี่ยนปัจจัยทางการบริหาร การวางแผนโครงการจะเริ่มตั้งแต่การกำหนดวัตถุประสงค์ของโครงการวิธีการจัดการและวิธีทำงานต่างๆ เพื่อให้บรรลุตามเป้าหมายโดยการแยกโครงการออกเป็นกลุ่มงานและงานแล้วเสร็จจึงกำหนดเวลาแต่ละงานตลอดจนความสัมพันธ์ระหว่างงานตามที่ระบุในเอกสาร Project Planning ที่แสดงในภาคผนวก



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

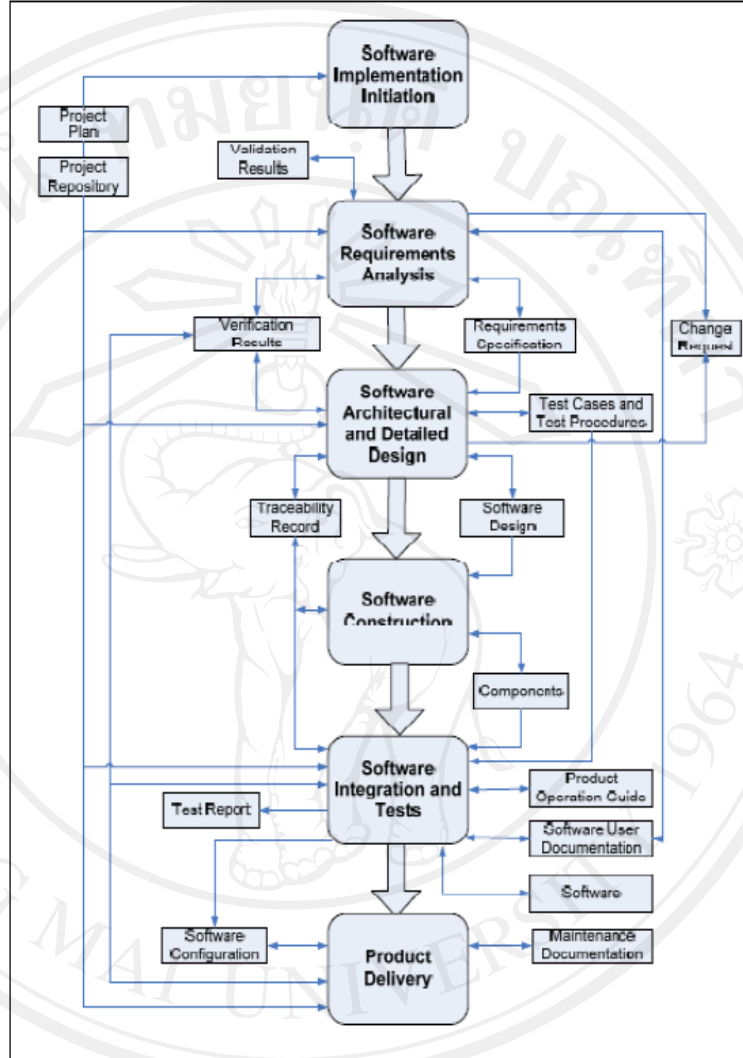
3.2.2 การติดตามแผนโครงการ (Project Planning Execution) การเก็บรวบรวมข้อมูล การปฏิบัติงานตามแผนงานที่กำหนดไว้เพื่อตัดสินใจแก้ไขปรับปรุงวิธีการปฏิบัติให้ผลงานเป็นไปตามแผนงานอย่างมีประสิทธิภาพมากที่สุด โดยมุ่งที่จะตอบคำถามหลักว่าในการดำเนินงานนั้น ได้รับทรัพยากรครบถ้วนหรือไม่ได้ปฏิบัติงานตามแผนงานที่กำหนดหรือไม่ได้ผลตรงตามที่กำหนดหรือไม่ ข้อมูลจากการติดตามจะเป็นประโยชน์อย่างยิ่งต่อการเร่งรัดให้งาน/โครงการดำเนินไปตรงตามเป้าหมายของแผนงานและแล้วเสร็จภายในกำหนดเวลาตามที่ระบุในเอกสาร Progress Status Record และ Meeting Record ที่แสดงในภาคผนวก

3.2.3 การตรวจสอบและควบคุมโครงการ (Project Assessment and Control) การควบคุมโครงการเป็นขั้นตอนสำคัญขั้นตอนหนึ่งซึ่งเป็นการบังคับให้ดำเนินกิจกรรมต่างๆเป็นไปตามกิจกรรมที่วางไว้ระหว่างดำเนินโครงการเพื่อให้สามารถดำเนินกิจกรรมได้ตามแผนที่วางไว้

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)

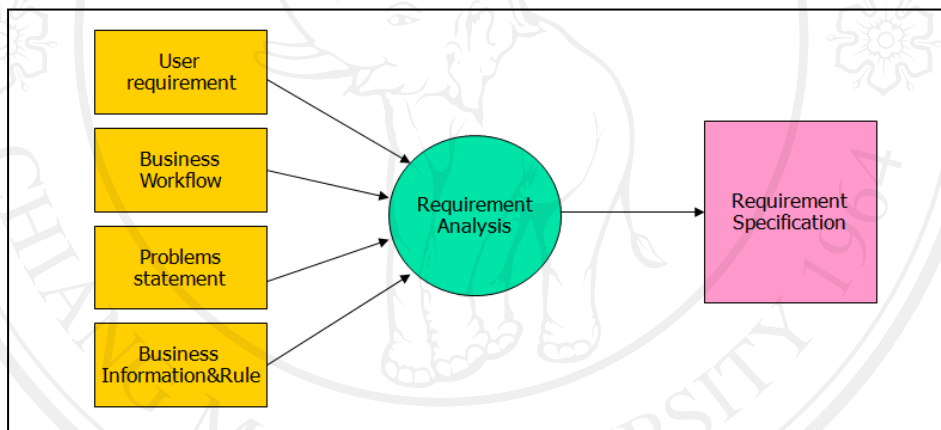
ของ ISO 29110 VSE

3.3.1 การเริ่มต้นการพัฒนาซอฟต์แวร์(Software Implementation Initiation)สิ่งสำคัญในการเริ่มต้นการพัฒนาซอฟต์แวร์ก็คือ การศึกษาวิสัยทัศน์และขอบเขตของงาน เพื่อให้เราสามารถรวบรวมความต้องการในเชิงธุรกิจ และของผู้ใช้งาน จากนั้นวิเคราะห์ระบบซอฟต์แวร์ที่ใช้อยู่ในปัจจุบัน รวมไปถึงการประเมินความเสี่ยง สิ่งที่ได้รับทราบถึงความต้องการที่เฉพาะเจาะจงตามจุดประสงค์ของโครงการรวมถึงอุปสรรคปัญหาของธุรกิจ ขอบข่ายของการพัฒนาระบบ

ซอฟต์แวร์และการประเมินความเสี่ยงในด้านต่าง ๆ ตามที่ระบุในเอกสารProject Planningที่แสดงในภาคผนวก

3.3.2 การวิเคราะห์ความต้องการซอฟต์แวร์(Software Requirements Analysis)คือกระบวนการวิเคราะห์เพื่อหาข้อกำหนดความต้องการของผู้ใช้ โดยจะต้องอาศัยข้อมูลในด้านต่างๆ ที่ได้รับมาจากผู้ใช้และองค์กรของผู้ใช้เพื่อทำการวิเคราะห์ดังรูปที่ 3.3 โดยการศึกษารูปนี้ได้แบ่งแหล่งข้อมูลเพื่อนำมาวิเคราะห์หาความต้องการดังนี้

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



รูปที่ 3.3 แหล่งข้อมูลเพื่อนำมาวิเคราะห์หาความต้องการระบบ

กระบวนการวิเคราะห์ความต้องการมีขั้นตอนดังต่อไปนี้

3.3.2.1 เก็บรวบรวมข้อมูลจากแบบสอบถามออนไลน์

3.3.2.2 วิเคราะห์เพื่อระบุถึงความต้องการต่างๆ

3.3.2.3 คัดเลือกส่วนที่เป็นสาระสำคัญและอยู่ในขอบเขตการพัฒนา

3.3.2.4 จัดจำแนกและจัดโครงสร้างของความต้องการ

3.3.2.5 จัดลำดับความสำคัญและตกลงเจรจา

3.3.2.6 ตรวจสอบความถูกต้อง

3.3.2.7 จัดทำเอกสารRequirement specificationที่แสดงในภาคผนวก

3.3.3 สถาปัตยกรรมซอฟต์แวร์และรายละเอียดการออกแบบ(Software Architectural and Detailed Design)หลังจากการวิเคราะห์ความต้องการของระบบเรียบร้อยแล้ว ขั้นตอนต่อไป คือ

การออกแบบระบบ ในขั้นตอนนี้จะมีกิจกรรมที่สำคัญประกอบด้วย การทบทวนความต้องการทั้งในด้านสารสนเทศและหน้าที่ของระบบ การพัฒนารูปแบบของระบบใหม่ ซึ่งประกอบด้วยรายละเอียดเกี่ยวกับผลลัพธ์ข้อมูลนำเข้า การประมวลผล หน่วยจัดเก็บข้อมูล ระเบียบการปฏิบัติ และบุคลากร ซึ่งการศึกษาครั้งนี้ได้แบ่งการออกแบบเป็น 5 ส่วนตามที่ระบุในเอกสาร Software Design ที่แสดงในภาคผนวกดังนี้

3.3.3.1 การออกแบบสถาปัตยกรรมของระบบ(System Architecture)

3.3.3.2 การออกแบบรายละเอียดของระบบ (Detailed Design) โดยออกแบบระบบในระดับ Software System เป็นการแสดงกลไกที่ใช้อธิบายเพื่อให้เข้าใจตรงกันว่าระบบควรมีบริการอะไรบ้างเพื่อให้ผู้ใช้บรรลุวัตถุประสงค์หลังจากใช้ระบบ ตามการวิเคราะห์ความต้องการต่างๆ ที่ได้จาก System Requirement เพื่อช่วยเหลือให้บุคคลที่เกี่ยวข้องกับระบบเข้าใจระบบได้ โดยเก็บคำอธิบายการปฏิสัมพันธ์กับระบบเอาไว้ใน Use case

3.3.3.3 การออกแบบฐานข้อมูล เพื่อใช้เก็บรวบรวมข้อมูลก่อนนำมาแสดงผล

3.3.3.4 การออกแบบส่วนผู้ใช้งาน(User Interface) มีวัตถุประสงค์หลัก เพื่อการรับคำสั่งจากผู้ใช้และแสดงผล โดยมีหลักการออกแบบส่วนผู้ใช้งานดังนี้

- ความเรียบง่าย ได้แก่ มีรูปแบบที่เรียบง่าย ไม่ซับซ้อน และใช้งานได้สะดวก ชนิตและสีของตัวอักษร
- ความสม่ำเสมอ ได้แก่ ใช้รูปแบบเดียวกันตลอดทั้งระบบ
- ความเป็นเอกลักษณ์

3.3.4 การสร้างซอฟต์แวร์(Software Construction)เป็นขั้นตอนนี้หลังจากการยอมรับแนวคิดที่ได้ออกแบบระบบ และประกอบเข้าด้วยกัน เพื่อให้ออกมาเป็นซอฟต์แวร์ที่สามารถใช้งานได้ ตรงตามข้อกำหนดที่ระบุไว้ในขั้นตอนนี้ผู้ศึกษาค้นคว้า ทำการเริ่มเขียนและทดสอบโปรแกรมว่า ทำงานถูกต้องหรือไม่ต้องมีการทดสอบกับข้อมูลจริงที่เลือกแล้ว ถ้าทุกอย่างเรียบร้อยจะได้โปรแกรมที่พร้อมนำไปใช้งานจริงต่อไปหลังจากนั้นต้องเตรียมคู่มือการใช้และการฝึกอบรมผู้ใช้งานจริงของระบบตามที่ระบุในเอกสาร Test Cases and Test Procedures ที่แสดงในภาคผนวก

3.3.5 การประกอบและการทดสอบซอฟต์แวร์(Software Integration and Tests)เป็นกระบวนการเพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามี ความถูกต้อง, ความสมบูรณ์, ปลอดภัย, และมีคุณภาพที่ดีการทดสอบ เป็นกระบวนการทดลองใช้ซอฟต์แวร์อย่างมีแนวทางโดยใช้ความรู้ทางด้านเทคนิค

เพื่อให้สามารถระบุหรือค้นหาความผิดพลาด(error)ของซอฟต์แวร์ที่อาจจะซ่อนอยู่ให้ปรากฏออกมา และสามารถระบุถึงแนวทางการเกิดปัญหาพร้อมสมมุติฐานของความผิดพลาดที่เกิดขึ้นได้ซึ่ง การศึกษาค้นคว้าครั้งนี้ได้แบ่งขั้นตอนการทดสอบดังนี้

- วิเคราะห์ความต้องการ
- จัดทำแผนงานการทดสอบ
- จัดทำแนวทางการทดสอบ
- ทดสอบจริง
- รายงานผลการทดสอบ
- ทดสอบผลจากการแก้ไข

การศึกษาค้นคว้าครั้งนี้ได้แบ่งวิธีการทดสอบดังนี้

3.3.5.1 Unit test เป็นการทดสอบส่วนย่อยของระบบแบบแยกจากกัน คำว่า “ส่วนย่อย” สามารถสื่อความหมายได้หลายแบบ เช่น functions, procedures, หรือ methods ตามความเหมาะสมของงาน

3.3.5.2 Integration test เป็นการทดสอบจุดประสงค์ เพื่อตรวจสอบความถูกต้องของ Function การทำงานต่างๆ เมื่อมีการ Integrate unit / module เข้าร่วมกัน โดยจะให้ความสำคัญในส่วนของการ Interface ระหว่างกันว่าสามารถใช้งานร่วมกันได้หรือไม่

3.3.5.3 User Acceptance Testing เป็นการทดสอบโดยผู้ใช้ (End user) หรือ โดยลูกค้าเพื่อดูว่าซอฟต์แวร์ทำงานถูกต้องตามต้องการหรือไม่

3.3.6 การส่งมอบ(Product Delivery)ประกอบด้วย ขั้นตอนการปรับเปลี่ยนระบบ การจัดทำเอกสารประกอบการฝึกอบรมผู้ใช้ ตลอดจนการบำรุงรักษาระบบ