

## บทที่ 3

### ระเบียบวิธีการศึกษา

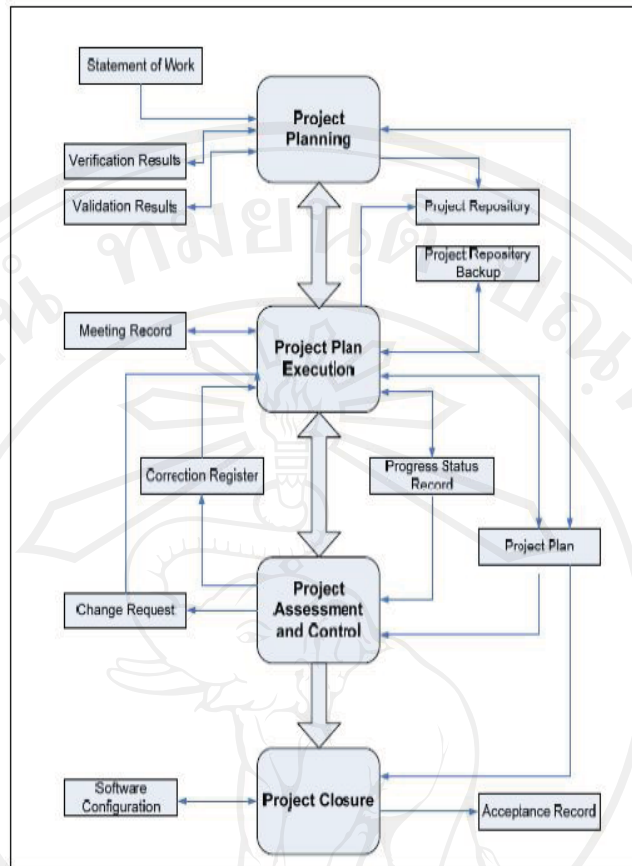
#### 3.1 แนวทางการค้นคว้าและพัฒนา

การศึกษาค้นคว้าและการพัฒนาออนไลน์สำหรับระบบให้คำแนะนำ อาจารย์ผู้เชี่ยวชาญตามหัวข้องานวิจัย เพื่อให้การค้นคว้าเป็นไปอย่างมีระเบียบหลักการ ผู้ค้นคว้าจึงเลือกพัฒนาซอฟต์แวร์ตามหลักการมาตรฐานอุตสาหกรรมซอฟต์แวร์ ISO 29110 VSE มีกระบวนการในการพัฒนาซอฟต์แวร์ที่เป็นระบบ และเข้าสู่กระบวนการสากล โดยจะเป็นการเริ่มต้นในเชิงกิจกรรมของการปรับปรุงกระบวนการ หรือ SPI (Software Process Improvement) ได้ให้ความสำคัญในกระบวนการที่จะต้องทำการปรับปรุงให้เป็นระบบและเป็นสากล 2 กระบวนการหลัก คือ กระบวนการด้านการบริหาร โครงการ (Project Management) ดังรูปที่ 3.1 และกระบวนการด้านการสร้างซอฟต์แวร์ (Software Implementation) ดังรูปที่ 3.2 ซึ่งจะประกอบด้วยกระบวนการย่อยๆภายในอีก ทั้งสองกระบวนการได้ถูกออกแบบให้เหมาะสมกับองค์กรขนาดเล็ก จึงมีความเหมาะสมในการประยุกต์ใช้ได้ทันที

#### 3.2 กระบวนการบริหารโครงการ (Project Management Process)

ประกอบด้วยกระบวนการย่อย 4 กระบวนการดังนี้

3.2.1 การวางแผนโครงการ (Project Planning) เป็นขั้นตอนแรกที่มีความเกี่ยวข้องกับความเป็นมาของ โครงการ โดยมีแนวความคิดจากแหล่งต่างๆ ที่ต้องการให้มีโครงการเกิดขึ้นและเป็นเรื่องเกี่ยวข้องกับ การเปลี่ยนแปลงจัดการบริหาร การวางแผนโครงการจะเริ่มตั้งแต่การกำหนดวัตถุประสงค์ของโครงการ วิธีการจัดการและวิธีทำงานต่างๆ เพื่อให้บรรลุตามเป้าหมาย โดยการแยก โครงการออกเป็นกลุ่มงานและงานแล้วเสร็จ จึงกำหนดเวลาแต่ละงาน ตลอดจนความสัมพันธ์ระหว่างงาน ตามที่ระบุในเอกสาร Project Planning ที่แสดงในภาคผนวก



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

3.2.2 การติดตามแผนโครงการ (Project Planning Execution) การเก็บรวบรวมข้อมูล การปฏิบัติงานตาม แผนงานที่กำหนดไว้ เพื่อตัดสินใจแก้ไข ปรับปรุงวิธีการปฏิบัติให้ผลงาน เป็นไปตาม แผนงานอย่างมีประสิทธิภาพมากที่สุด โดยมุ่งที่ จะตอบคำถามหลักว่า ในการ ดำเนินงานนั้น ได้รับทรัพยากรครบถ้วนหรือไม่ ได้ปฏิบัติงาน ตามแผนงานที่กำหนดหรือไม่ ได้ผล ตรงตามที่ กำหนดหรือไม่ ข้อมูลจากการติดตามจะเป็น ประโยชน์อย่างยิ่งต่อการเร่งรัดให้งาน/ โครงการ ดำเนินไปตรงตามเป้าหมายของแผนงานและแล้ว เสร็จภายในกำหนดเวลา ตามที่ระบุใน เอกสาร Progress Status Record และ Meeting Record ที่แสดงในภาคผนวก

3.2.3 การตรวจสอบและควบคุมโครงการ (Project Assessment and Control) การ ควบคุมโครงการเป็นขั้นตอนสำคัญขั้นตอนหนึ่ง ซึ่งเป็นการบังคับให้ดำเนินกิจกรรมต่างๆ เป็นไป ตามกิจกรรมที่วางไว้ ระหว่างดำเนินโครงการ เพื่อให้สามารถดำเนินกิจกรรมได้ตามแผนที่วางไว้

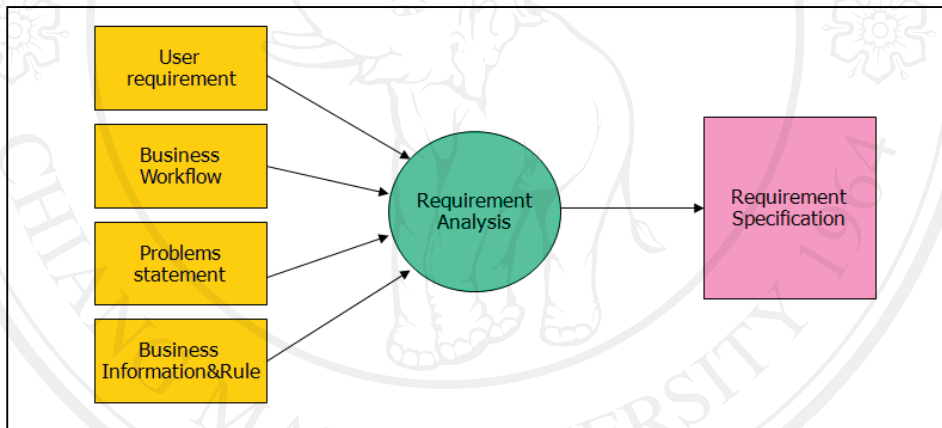
3.2.4 การปิดโครงการ (Project Closure) ตามที่ระบุในเอกสาร Acceptance Record ที่ แสดงในภาคผนวก



พัฒนาระบบซอฟต์แวร์ และการประเมินความเสี่ยงในด้านต่าง ๆ ตามที่ระบุในเอกสาร Project Planning ที่แสดงในภาคผนวก

3.3.2 การวิเคราะห์ความต้องการซอฟต์แวร์ (Software Requirements Analysis) คือ กระบวนการวิเคราะห์เพื่อหาข้อกำหนดความต้องการของผู้ใช้ โดยจะต้องอาศัยข้อมูลในด้านต่างๆ ที่ได้รับมาจากผู้ใช้และองค์กรของผู้ใช้เพื่อทำการวิเคราะห์ ดังรูปที่ 3.3 โดยการศึกษาค้างนี้ได้แบ่ง แหล่งข้อมูลเพื่อนำมาวิเคราะห์หาความต้องการดังนี้

- (1.) ความต้องการของผู้ใช้ (User Requirement)
- (2.) กระบวนการทำงานขององค์กร(Business Workflow)
- (3.) ประเด็นปัญหาที่พบ (Problems Statement )
- (4.) ข้อมูลและกฎระเบียบขององค์กร (Business Information&Rule)



รูปที่ 3.3 แหล่งข้อมูลเพื่อนำมาวิเคราะห์หาความต้องการระบบ

กระบวนการวิเคราะห์ความต้องการมีขั้นตอนดังต่อไปนี้

3.3.2.1 เก็บรวบรวมข้อมูลที่เป็นข้อเท็จจริงต่างๆ จากผู้ที่เกี่ยวข้องกับการใช้งานระบบ ได้แก่ อาจารย์ผู้เชี่ยวชาญ, เจ้าหน้าที่ดูแลระบบ, นักศึกษา, บุคคลภายนอกองค์กร แต่สามารถเข้าถึงบริการของระบบในองค์กรได้

3.3.2.2 วิเคราะห์เพื่อระบุถึงความต้องการต่างๆ

3.3.2.3 คัดเลือกส่วนที่เป็นสาระสำคัญและอยู่ในขอบเขตการพัฒนา

3.3.2.4 จัดจำแนกและจัด โครงสร้างของความต้องการ

3.3.2.5 จัดลำดับความสำคัญและตกลงเจรจา

3.3.2.6 ตรวจสอบความถูกต้อง

3.3.2.7 จัดทำเอกสาร Requirement specification ที่แสดงในภาคผนวก

3.3.3 สถาปัตยกรรมซอฟต์แวร์และรายละเอียดการออกแบบ (Software Architectural and Detailed Design) หลังจากการวิเคราะห์ความต้องการของระบบเรียบร้อยแล้ว ขั้นตอนต่อไป คือ การออกแบบระบบ ในขั้นตอนนี้จะมีกิจกรรมที่สำคัญประกอบด้วย การทบทวนความต้องการทั้ง ในด้านสารสนเทศและหน้าที่ของระบบ การพัฒนารูปแบบของระบบใหม่ ซึ่งประกอบด้วย รายละเอียดเกี่ยวกับผลลัพธ์ข้อมูลนำเข้า การประมวลผล หน่วยจัดเก็บข้อมูล ระเบียบการปฏิบัติ และบุคลากร ซึ่งการศึกษาค้นคว้าครั้งนี้ได้แบ่งการออกแบบเป็น 5 ส่วน ตามที่ระบุในเอกสาร Software Design ที่แสดงในภาคผนวก ดังนี้

3.3.3.1 การออกแบบสถาปัตยกรรมของระบบ (System Architecture) ให้คำแนะนำ อาจารย์ผู้เชี่ยวชาญตามหัวข้องานวิจัย โดยการเข้าถึงฐานข้อมูลบน พื้นฐานของออนโทโลยี (Ontology)

3.3.3.2 การออกแบบรายละเอียดของระบบ (Detailed Design) โดยออกแบบ ดังนี้

- (1) การออกแบบระบบในระดับ Software System เป็นการแสดงกลไก ที่ใช้อธิบายเพื่อให้เข้าใจตรงกันว่าระบบควรมีบริการอะไรบ้าง เพื่อให้ผู้ใช้บรรลุวัตถุประสงค์หลังจากใช้ระบบ ตามการวิเคราะห์ ความต้องการต่างๆ ที่ได้จาก System Requirement เพื่อช่วยเหลือ ให้บุคคลที่เกี่ยวข้องกับระบบ เข้าใจระบบได้ โดยเก็บคำอธิบายการ ปฏิสัมพันธ์กับระบบเอาไว้ใน Use case
- (2) การออกแบบระบบในระดับ Subsystem หรือ package ซึ่งเป็นการ อธิบายรายละเอียดของระบบย่อยต่างๆ ควรมีบริการอะไรบ้าง ตาม การวิเคราะห์จากผังงานระบบ (System Flowcharts) เพื่อช่วยเหลือ ให้บุคคลที่เกี่ยวข้องกับระบบ เข้าใจระบบได้ โดยเก็บคำอธิบายการ ปฏิสัมพันธ์กับระบบเอาไว้ใน Use case

3.3.3.3 การออกแบบฐานข้อมูล เพื่อให้เก็บรวบรวมข้อมูลก่อนนำมาแสดงผล

3.3.3.4 การออกแบบออนโทโลยี (Ontology) ใช้สำหรับเก็บคำศัพท์ที่ใช้งาน ร่วมกันการเข้าถึงฐานข้อมูลบนพื้นฐานของออนโทโลยีด้วยวิธีการปรับเปลี่ยน คำสั่งการสืบค้น เพื่อมุ่งเน้นการเข้าถึงฐานข้อมูลแบบสื่อความหมาย โดยใช้วิธีการ แปลงจากฐานข้อมูลเชิงสัมพันธ์ไปเป็นออนโทโลยี (Ontology) โดยมี กระบวนการออกแบบออนโทโลยีดังนี้

- (1) ระบุขอบเขตของแนวคิดของออนโทโลยี (Determine scope)
- (2) พิจารณาเลือกใช้ตัวแบบออนโทโลยีที่มีอยู่แล้ว (Consider reuse)
- (3) กำหนดคำศัพท์หรือนิยามสำคัญของออนโทโลยี (Enumerate terms)
- (4) การระบุคลาสและคุณสมบัติของคลาส (Define classes)
- (5) การระบุความสัมพันธ์ระหว่างแนวคิด (Define Relations)
- (6) สร้างตัวอย่างของข้อมูล (Create instances) การกำหนดตัวอย่างของข้อมูลในลำดับชั้นของคลาส

3.3.3.5 การออกแบบส่วนผู้ใช้งาน (User Interface) มีวัตถุประสงค์หลักเพื่อการรับคำสั่งจากผู้ใช้และแสดงผล โดยมีหลักการออกแบบส่วนผู้ใช้งานดังนี้

- (1) ความเรียบง่าย ได้แก่ มีรูปแบบที่เรียบง่าย ไม่ซับซ้อน และใช้งานได้สะดวก ไม่มีกราฟิกหรือตัวอักษรที่เคลื่อนไหวอยู่ตลอดเวลา ชนิดและสีของตัวอักษรไม่มากจนเกินไปทำให้วุ่นวาย
- (2) ความสม่ำเสมอ ได้แก่ ใช้รูปแบบเดียวกันตลอดทั้งระบบ
- (3) ความเป็นเอกลักษณ์ คำนึงถึงเอกลักษณ์และลักษณะขององค์กรนั้น

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

3.3.5 การประกอบและการทดสอบซอฟต์แวร์ (Software Integration and Tests) เป็นกระบวนการเพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามี ความถูกต้อง, ความสมบูรณ์, ปลอดภัย, และมีคุณภาพที่ดี การทดสอบ เป็นกระบวนการทดลองใช้ซอฟต์แวร์อย่างมีแนวทาง โดยใช้ความรู้ทางด้านเทคนิค เพื่อให้สามารถระบุหรือค้นหาความผิดพลาด (error) ของซอฟต์แวร์ที่อาจจะซ่อนอยู่ให้ปรากฏออกมา และสามารถระบุถึงแนวทางการเกิดปัญหา พร้อมสมมุติฐานของความผิดพลาดที่เกิดขึ้นได้ ซึ่งการศึกษาค้นคว้าครั้งนี้ได้แบ่งขั้นตอนการทดสอบดังนี้



- (1.) วิเคราะห์ความต้องการ
- (2.) จัดทำแผนงานการทดสอบ
- (3.) จัดทำแนวทางการทดสอบ
- (4.) ทดสอบจริง
- (5.) รายงานผลการทดสอบ
- (6.) ทดสอบผลจากการแก้ไข

การศึกษาค้นคว้าครั้งนี้ได้แบ่งวิธีการทดสอบดังนี้

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

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

3.3.5.3 User Acceptance Testing เป็นการทดสอบโดยผู้ใช้ (End user) หรือ โดยลูกค้า เพื่อคว่าซอฟต์แวร์ทำงานถูกต้องตามต้องการหรือไม่

3.3.6 การส่งมอบ (Product Delivery) ประกอบด้วย ขั้นตอนการปรับเปลี่ยนระบบ การจัดทำเอกสารประกอบการฝึกอบรมผู้ใช้ ตลอดจนการบำรุงรักษาระบบ