

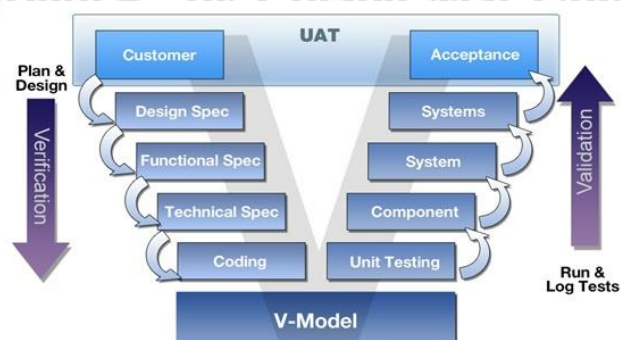
## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาครั้งนี้ ทางผู้ศึกษาได้ค้นคว้าหาข้อมูล ทฤษฎี จากแหล่งความรู้ต่าง ๆ ที่เกี่ยวข้องกับ การทดสอบซอฟต์แวร์ (Software Testing) ทั้งในเรื่องของประเภทของการทดสอบซอฟต์แวร์ กิจกรรม ในการทดสอบ และความเชื่อมโยงของกระบวนการทดสอบซอฟต์แวร์กับกระบวนการพัฒนา ซอฟต์แวร์ เพื่อให้การพัฒนาระบบสนับสนุนกระบวนการทดสอบซอฟต์แวร์ สำหรับบริษัท โปร ซอฟท์ คอมเทค จำกัด ออกมามีประสิทธิภาพ และบรรลุวัตถุประสงค์ที่ได้อย่างแท้จริง

#### 2.1 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ คือ กระบวนการในการตรวจสอบ ควบคุม และยืนยันว่าซอฟต์แวร์ที่ พัฒนาขึ้นมาเป็นไปตามข้อกำหนดความต้องการที่ได้ตกลงไว้กับผู้ใช้งานระบบ หรือลูกค้า ซึ่งการ ทดสอบซอฟต์แวร์จะประกอบไปด้วยเวริฟิเคชัน (Verification) และวาลิเดชัน (Validation) โดยที่ เวริฟิเคชัน คือการตรวจสอบกระบวนการพัฒนาซอฟต์แวร์ว่าเป็นไปตามมาตรฐาน และจะ รับประกันได้ว่าจะได้ผลลัพธ์ที่ถูกต้อง ส่วนวาลิเดชัน คือการตรวจสอบผลลัพธ์ของการพัฒนา เป็นไปตามความต้องการหรือไม่ มีข้อบกพร่องใด ๆ ที่ก่อให้เกิดผลกระทบกับการส่งมอบหรือไม่ (Mustafa K. and Khan, R.A. 2007)



รูปที่ 2.1 แผนภาพประกอบเพื่อทำความเข้าใจความแตกต่างระหว่าง การทดสอบและทวนสอบระบบ

ที่มา: (www.collegepals.org)

จากรูปที่ 2.1 จะเห็นได้ว่าการทำเวริฟิเคชันนั้น เกี่ยวข้องกับกระบวนการพัฒนาซอฟต์แวร์เริ่มตั้งแต่การวิเคราะห์ความต้องการกับลูกค้า การออกแบบซอฟต์แวร์ จนไปถึงการเขียนโปรแกรม ซึ่งกระบวนการข้างต้นยังไม่ปรากฏเป็นซอฟต์แวร์ ส่วนการทำวาไลเดชันจะเกี่ยวข้องกับการตรวจสอบซอฟต์แวร์ซึ่งเป็นการตรวจสอบผลลัพธ์ที่ได้ออกมาแล้ว โดยทำตั้งแต่ในระดับยูนิตไปจนถึงการทดสอบเมื่อรับมอบกับลูกค้า

### 2.1.1 ประเภทของการทดสอบซอฟต์แวร์

Mustafa K. และ Khan, R.A. (2007) ได้จำแนกประเภทของการทดสอบซอฟต์แวร์ออกเป็นสองประเภทหลัก ๆ ดังนี้

- การทดสอบแบบกล่องดำ (Black Box Testing) คือการทดสอบระบบโดยที่ไม่เห็นโครงสร้าง หรือการทำงานภายในของระบบ โดยจะเน้นไปที่ส่วนของ Input และ Output เท่านั้น
- การทดสอบแบบกล่องขาว (White Box Testing) คือ การทดสอบระบบโดยที่ดูในเรื่องของโครงสร้าง และการทำงานภายในระบบด้วย โดยจะลงไปถึงในระดับซอร์สโค้ดหรือ อัลกอริทึมซึ่งหมายความว่าผู้ทำการทดสอบจะต้องมีความรู้ทางด้าน การเขียนโปรแกรมในระดับหนึ่ง

### 2.1.2 วิธีการทดสอบซอฟต์แวร์

Jorgensen, Paul C. (2008) ได้กล่าวถึงวิธีการทดสอบที่เกิดขึ้นในกระบวนการทดสอบซอฟต์แวร์ไว้ดังต่อไปนี้

- การทดสอบระดับฟังก์ชัน (Functional Testing) คือ การทดสอบระบบในระดับฟังก์ชันการทำงาน โดยเน้นการตรวจสอบการทำงานว่าเป็นไปตามที่ได้กำหนดไว้ในข้อกำหนดความต้องการหรือไม่
- การทดสอบซ้ำ (Regression Testing) คือ การทดสอบระบบทั้งหมดอีกครั้งหลังจากที่มีการแก้ไขฟังก์ชัน หรือแก้ไขระบบส่วนใดส่วนหนึ่ง เพื่อตรวจสอบผลกระทบที่อาจจะเกิดขึ้นจากการแก้ไข

### 2.1.3 ระดับของการทดสอบซอฟต์แวร์

Jorgensen, Paul C. (2008) ได้อธิบายถึงขั้นตอนระดับของการทดสอบซอฟต์แวร์ โดยประกอบไปด้วยระดับการทดสอบซอฟต์แวร์ ดังต่อไปนี้

- การทดสอบระดับยูนิต (Unit Testing) คือ การทดสอบระบบในระดับฟังก์ชันการทำงาน ซึ่งจะเป็นส่วนที่เล็กที่สุดของการทดสอบระบบ ส่วนใหญ่จะทำการทดสอบโดยนักพัฒนาระบบเอง
- การทดสอบการทำงานร่วมกัน (Integration Testing) คือ การทดสอบระบบหลังจากมีการนำเอาแต่ละฟังก์ชัน หรือแต่ละระบบย่อยมารวมเข้าด้วยกัน จะเป็นการทดสอบการทำงานร่วมกันของแต่ละฟังก์ชัน หรือระบบย่อย
- การทดสอบทั้งระบบ (System Testing) คือ การทดสอบระบบทั้งหมดก่อนที่จะส่งมอบ หรือตรวจสอบเพื่อการอนุมัติการปิดโครงการ จะเป็นการทดสอบการทำงานที่จะใช้ข้อมูลที่ใกล้เคียงการทำงานจริงทั้งหมด
- การทดสอบเพื่อการรับมอบ (User Acceptance Testing) คือ การทดสอบระบบเพื่อรับมอบซอฟต์แวร์ หรืออนุมัติการปิดโครงการ ซึ่งจะเป็นการทดสอบโดยลูกค้าเจ้าของโครงการ

### 2.1.4 ประเภทของจุดบกพร่อง (Bug)

Jorgensen, Paul C. (2008) ได้กล่าวถึงจุดบกพร่อง โดยมีการแบ่งประเภทออกเป็น

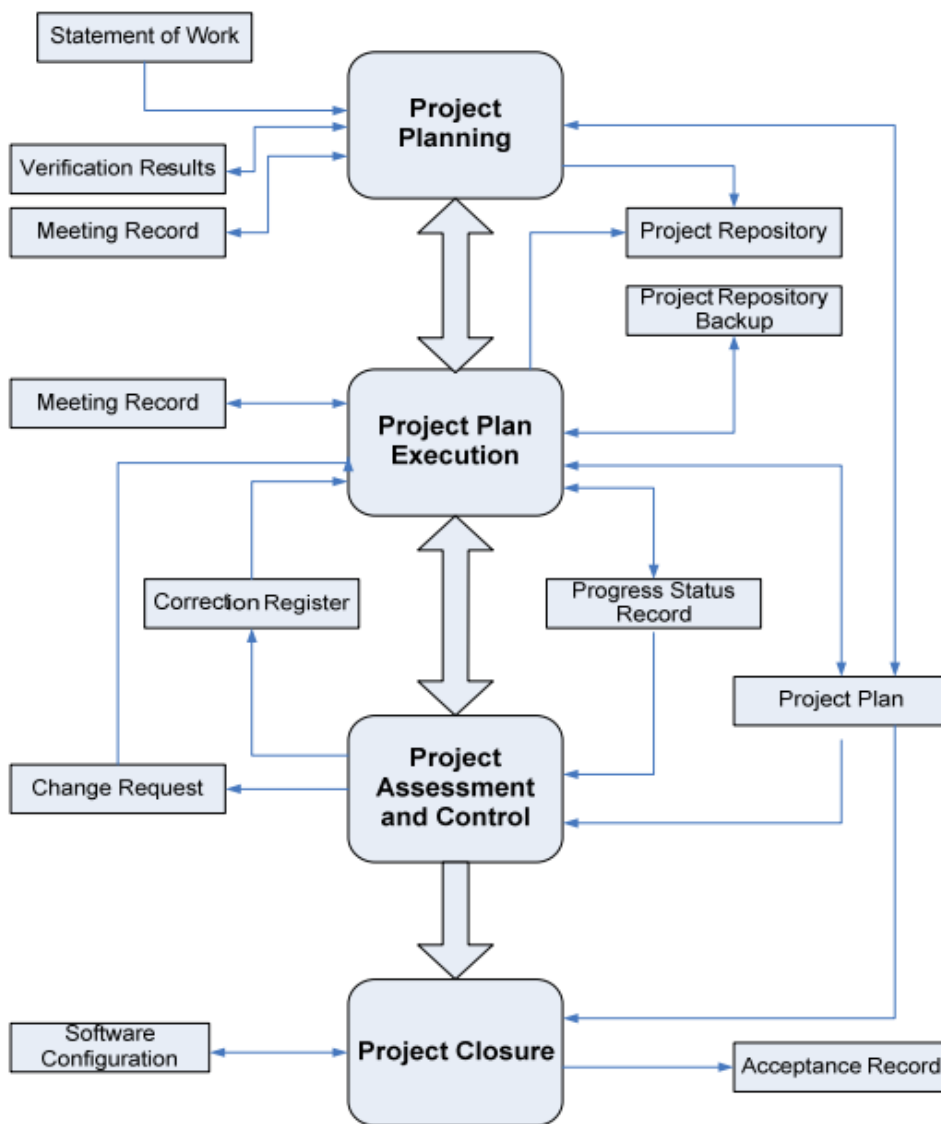
- ข้อบกพร่อง (Defect) คือ การกระทำ หรือกระบวนการที่เป็นสาเหตุทำให้เกิดข้อผิดพลาด
- ข้อผิดพลาด (Error) คือ ข้อบกพร่องหรือข้อผิดพลาดของระบบ ซึ่งอาจจะกลายเป็นสาเหตุทำให้ระบบล้มเหลว หรือไม่สามารถทำงานต่อไปได้
- ความล้มเหลว (Failure) คือ ความล้มเหลวของระบบ การที่ระบบไม่สามารถทำงานได้ตามที่กำหนดไว้

## 2.2 มาตรฐานไอเอสโอ/ไออีซี 29110

มาตรฐานไอเอสโอ/ไออีซี 29110 (ISO/IEC 29110) เป็นแนวคิดยุคใหม่ของไอเอสโอ ที่จะเน้นการเติบโตของอุตสาหกรรมขนาดกลางและขนาดเล็ก รวมทั้งผู้ประกอบการรายใหม่ให้มีโอกาสในการแข่งขัน ที่ผ่านมามาตรฐานวิศวกรรมซอฟต์แวร์ได้ถูกทำให้เป็นเรื่องที่เข้าใจยากและมีความ

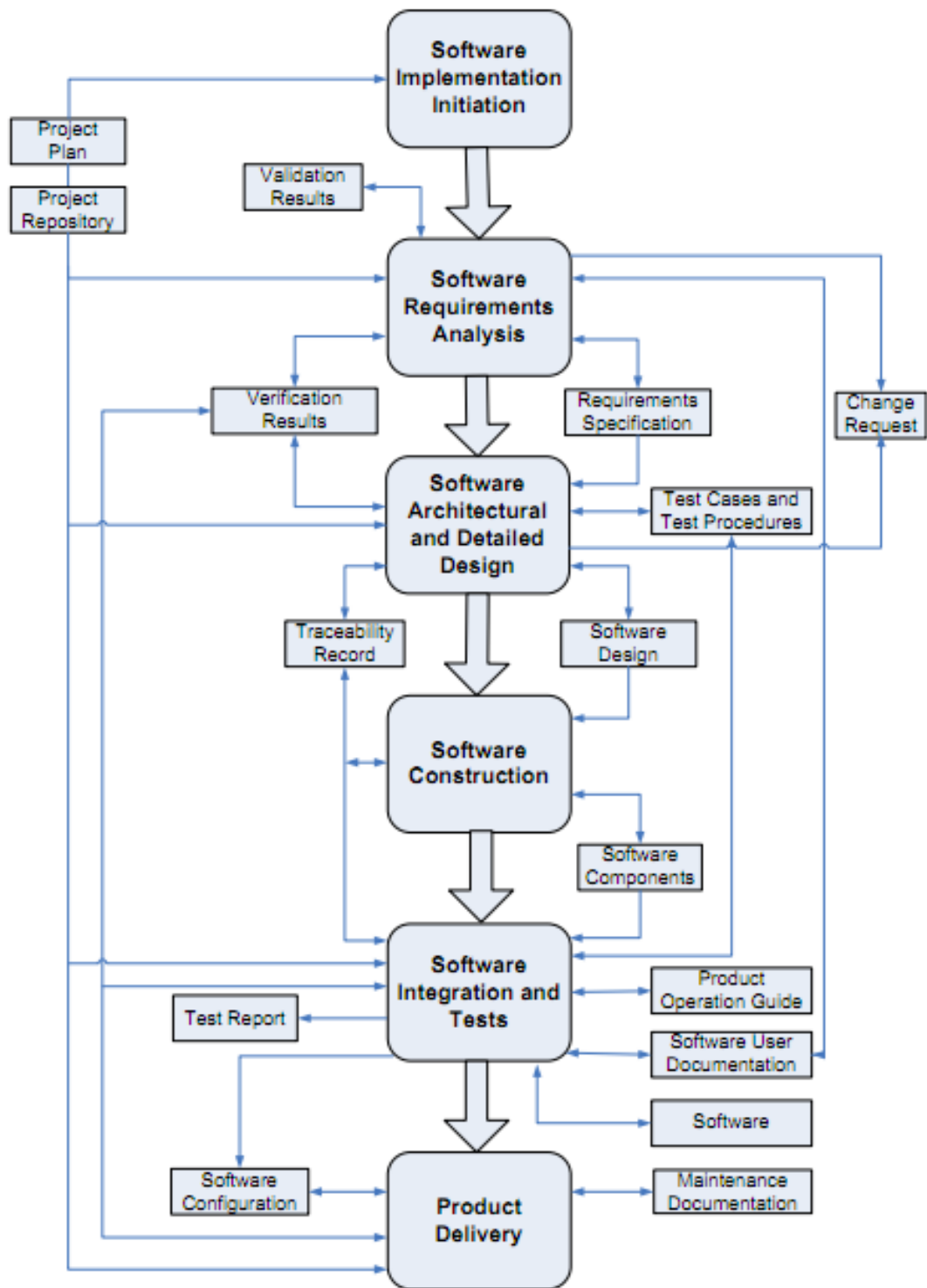
สลับซับซ้อนยุ่งยากในการที่จะปฏิบัติตาม ประกอบกับมาตรฐานซอฟต์แวร์ระดับสากลที่มีอยู่ในปัจจุบันจะเหมาะสมกับการปฏิบัติงานขององค์กรขนาดใหญ่ ไอเอสโอ 29110 จึงถูกพัฒนาด้วยแนวคิดพื้นฐานเพื่อสนับสนุนองค์กรขนาดเล็กให้มีโอกาสในการปรับปรุงกระบวนการและรับรองคุณภาพในระดับสากล (มูลนิธิสถาบันเพื่อพัฒนานวัตกรรม, 2555)

มาตรฐานไอเอสโอ 29110 ให้ความสำคัญในกระบวนการที่จะต้องทำการปรับปรุงระบบ อยู่ 2 กระบวนการหลัก คือ กระบวนการด้านการบริหาร โครงการ (Project Management) และกระบวนการด้านการสร้างซอฟต์แวร์ (Software Implement) ดังที่แสดงในรูปที่ 2.2 และ 2.3



รูปที่ 2.2 แผนภาพแสดงกระบวนการทางด้านการบริหารโครงการ

ที่มา: (www.innova.or.th)



รูปที่ 2.3 แผนภาพแสดงกระบวนการด้านการสร้างซอฟต์แวร์  
ที่มา: (www.innova.or.th)

### 2.2.1 ข้อกำหนดการประเมินไอเอสโอ 29110

หัวข้อที่ 1) – 4) จะเป็นการอธิบายรายละเอียดและข้อกำหนดของกระบวนการด้านการบริหารโครงการจากในรูปที่ 2.2 และหัวข้อที่ 5) – 8) จะเป็นการอธิบายรายละเอียดและข้อกำหนดของกระบวนการด้านการสร้างซอฟต์แวร์จากในรูปที่ 2.3

#### 1) การวางแผนโครงการ

- โครงการมีข้อกำหนดอ้างอิงสัญญา หรือที่โออาร์ที่ระบุถึงเนื้อหาของงานและขอบเขตการทำงานที่ชัดเจน
- กิจกรรมที่จะต้องดำเนินการในโครงการและทรัพยากรที่ใช้มีการประมาณการอย่างเป็นระบบสามารถระบุถึงค่าความพยายาม (Effort) ที่ชัดเจน
- มีนโยบายหรือกลยุทธ์ในการควบคุมด้านเวอร์ชันของซอฟต์แวร์และนำไปใช้ในการปฏิบัติงานจริง
- แผนโครงการมีการกำหนดการจัดส่งมอบชิ้นงานที่ชัดเจนและมีรายละเอียดของชิ้นงานที่ส่งมอบ
- มีการวางแผนโครงการและสร้างตารางแผนโครงการในลักษณะกำหนดการ

#### 2) การบริหารแผนโครงการ

- มีการกระจายแผนโครงการแจ้งแผนโครงการแก่ทีมงานและลูกค้า
- มีการประชุมและบันทึกผลการประชุมระหว่างทีมงานและการประชุมกับลูกค้า
- มีการจัดการด้านการควบคุมการเปลี่ยนแปลงต่างๆการจดทะเบียนร้องขอการเปลี่ยนแปลง
- มีกระบวนการตรวจสอบคุณภาพการทำงานในขั้นตอนต่างๆ
- มีกระบวนการในการตรวจสอบร่วมกันกับลูกค้าในการยืนยันถึงลักษณะของซอฟต์แวร์ที่ตรงตามความต้องการ

#### 3) การประเมินและการติดตามโครงการ

- มีการติดตามความคืบหน้าในการดำเนินโครงการและบันทึกผลออกเป็นรายงานที่แสดงเป็นลำดับขั้นตามการติดตามก่อนและหลัง
- มีการบันทึกผลถึงปัญหาอุปสรรคในการดำเนินโครงการและกิจกรรมที่ทำให้แผนโครงการที่ได้ออกไว้มีการปรับเปลี่ยนรวมทั้งสามารถติดตามการเปลี่ยนแปลงของเวอร์ชันของแผนโครงการได้อย่างเป็นระบบ
- ปัญหาของโครงการที่ทำให้แผนโครงการปรับเปลี่ยนได้รับการแก้ไขและบันทึกผลการทำงาน
- ความเสี่ยงได้ถูกระบุและทีมงานได้รับทราบถึงปัจจัยของความเสี่ยงและการป้องกัน
- รายการชิ้นงานที่จะต้องควบคุมในการจัดการองค์ประกอบในโครงการซอฟต์แวร์ (Software Configuration) ได้รับการระบุและกำหนดไว้อย่างเหมาะสมรวมทั้งรายละเอียดต่างๆที่เกี่ยวข้อง

#### 4) การปิดโครงการ

- โครงการมีการบันทึกการตรวจรับซอฟต์แวร์และการจัดส่งการเป็นเอกสารที่ชัดเจน
- มีการสร้างระบบเซิร์ฟเวอร์ที่ควบคุมดูแลระบบเอกสารและซอร์สโค้ดระหว่างการดำเนินโครงการและมีการจัดเก็บที่เป็นระบบตรงตามการจัดการองค์ประกอบในโครงการซอฟต์แวร์ที่ได้กำหนดไว้
- มีการจัดการด้านการจัดการองค์ประกอบในโครงการซอฟต์แวร์ก่อนส่งมอบงาน
  - สามารถสร้างคลังเอกสารเพื่อการดำเนินการด้านดูแลแก้ไขอย่างต่อเนื่องได้
- มีระบบในการสำรองและกู้คืนข้อมูล เอกสารต่างๆ และซอฟต์แวร์

#### 5) การพัฒนาความต้องการ

- มีการพัฒนาความต้องการอย่างเป็นระบบ และบันทึกผลไว้
- ความต้องการที่บันทึกไว้มีรายละเอียดที่ชัดเจน ถูกต้อง สามารถนำไปพัฒนาและไม่มีปัญหาในการสร้างความเข้าใจร่วมกันในการทดสอบและตรวจรับ

- ความต้องการ ได้ถูกอนุมัติโดยลูกค้า
  - เอกสารความต้องการได้รับการจัดเก็บอย่างเป็นระบบ และมีเบสไลน์ (Baseline) ตามหลักการจัดการองค์ประกอบในโครงการซอฟต์แวร์
  - ความต้องการได้รับการสื่อสารให้ทีมงานเข้าใจตรงกันและได้รับการยอมรับ
- 6) การออกแบบสถาปัตยกรรมระบบ
- เอกสารการออกแบบสถาปัตยกรรมระบบได้รับการจัดเก็บอย่างเป็นระบบ และมีเบสไลน์ตามหลักการจัดการองค์ประกอบในโครงการซอฟต์แวร์
  - เอกสารการออกแบบระบบโดยลงรายละเอียด (Detail Design) ได้รับการจัดเก็บอย่างเป็นระบบ และมีเบสไลน์ตามหลักการจัดการองค์ประกอบในโครงการซอฟต์แวร์
  - การออกแบบระบบมีการระบุถึงการติดต่อกับส่วนประสานภายนอกระบบ และข้อกำหนดของลักษณะการเชื่อมต่อ
  - การออกแบบระบบมีการระบุถึงการติดต่อกับส่วนประสานภายในระบบ และข้อกำหนดของลักษณะการเชื่อมต่อ
  - การออกแบบระบบสามารถสร้างการเชื่อมโยงตรวจสอบย้อนกลับไปที่ความต้องการได้
- 7) การสร้างการพัฒนาซอฟต์แวร์
- ซอฟต์แวร์คอมโพเนนต์ (Software Components) ได้รับการกำหนดและระบุลงรายละเอียด
  - ซอฟต์แวร์คอมโพเนนต์ที่ได้รับการพัฒนา
  - มีการทดสอบซอฟต์แวร์รายโปรแกรมหรือโมดูล
  - ข้อบกพร่องที่เกิดระหว่างการทดสอบโปรแกรมได้รับการแก้ไข
  - การทดสอบระบบสามารถสร้างการเชื่อมโยงตรวจสอบย้อนกลับไปที่ความต้องการและการออกแบบได้
- 8) การบูรณาการและการทดสอบระบบ
- มีกรณีการทดสอบและกระบวนการทดสอบ



- มีการบันทึกผลการทดสอบลงในรายงานการทดสอบ
- ข้อบกพร่องจากการทดสอบได้รับการแก้ไข
- เอกสารสำหรับผู้ใช้งานระบบได้รับการพัฒนา
- เอกสารแนวทางการปฏิบัติงานได้รับการพัฒนา

### 2.3 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องกับการบริหารจัดการโครงการพัฒนา การควบคุมคุณภาพซอฟต์แวร์ สามารถสรุปงานวิจัยที่เกี่ยวข้องได้ดังต่อไปนี้

โสภิตา ทองคู่ (2551) ได้ทำการศึกษาวิจัยเรื่อง การพัฒนาระบบการบริหารโครงการซอฟต์แวร์ของบริษัท นอร์เทิร์น พิคส์ เทคดิง ไพรเวท จำกัด โดยมีวัตถุประสงค์เพื่อช่วยในการบริหารเวลาในการพัฒนาซอฟต์แวร์ ซึ่งจะประกอบไปด้วยการจัดการข้อมูลลูกค้า การจัดการข้อมูลโครงการและตารางเวลาของโครงการ และการรายงานความคืบหน้าของโครงการต่อผู้บริหาร โดยผลสรุปการใช้งานระบบพบว่าสามารถช่วยอำนวยความสะดวกในการบริหารโครงการซอฟต์แวร์ อีกทั้งยังช่วยให้ลูกค้าสามารถดูรายงานความคืบหน้าได้สะดวกมากยิ่งขึ้น หลังจากศึกษาแล้วพบว่าระบบนี้จะมีความใกล้เคียงกับระบบการพัฒนาระบบสนับสนุนกระบวนการทดสอบซอฟต์แวร์ สำหรับบริษัท โปรซอฟท์ คอมเทค จำกัด ในส่วนของการจัดการข้อมูลโครงการและผู้รับผิดชอบ

ภัทรกร โพธิแก้ว (2552) ได้ทำการศึกษาวิจัยเรื่อง การพัฒนาระบบบริหารโครงการซอฟต์แวร์ บริษัท อินคอน จำกัด โดยมีวัตถุประสงค์เพื่อพัฒนาระบบสำหรับช่วยในการแก้ไขปัญหาการทับซ้อนของตารางการทำงาน การส่งงานล่าช้ากว่ากำหนดการ และช่วยอำนวยความสะดวกในการติดตามความคืบหน้าของโครงการของผู้บริหารและลูกค้า ซึ่งในตัวระบบจะประกอบไปด้วยการจัดการโครงการ การจัดการโครงการ โดย PERT/CPM การจัดการข้อมูลพนักงาน การออกรายงานและติดตามโครงการ และการจัดการข้อมูลผู้ใช้งานระบบ งานวิจัยนี้ได้ผลสรุปผลการศึกษาว่าระบบสามารถช่วยเพิ่มประสิทธิภาพในการบริหารโครงการและทรัพยากรที่ใช้ในแต่ละโครงการให้ดียิ่งขึ้นสามารถช่วยในการควบคุมคุณภาพซอฟต์แวร์ ให้เป็นไปตามมาตรฐานการพัฒนาซอฟต์แวร์ รวมทั้งยังสามารถตอบสนองความต้องการของผู้ใช้งานแต่ละกลุ่มได้อย่างเหมาะสม

ภาณุพงศ์ ประภัสระกุล (2554) ได้ทำการศึกษาวิจัยเรื่อง ระบบพัฒนาคุณภาพของวิศวกรซอฟต์แวร์โดยกระบวนการซอฟต์แวร์ระดับบุคคลบนโปรแกรมประยุกต์เว็บ การค้นคว้าแบบอิสระนี้มีวัตถุประสงค์เพื่อพัฒนาคุณภาพของวิศวกรซอฟต์แวร์ โดยใช้โปรแกรมประยุกต์เว็บมาเป็นเครื่องมือช่วยในการบริหารจัดการข้อมูลให้สามารถเข้าถึงได้ง่ายมากยิ่งขึ้น เริ่มต้นจากปัญหาเกี่ยวกับความไม่

เข้าในในทฤษฎีกระบวนการพัฒนาซอฟต์แวร์ระดับบุคคล ข้อจำกัดในการเข้าถึงข้อมูลการทำงานเมื่อ  
อยู่ภายนอกสถานที่ทำงาน และการรายงานผลการทำงานที่ยังไม่เหมาะสมกับความต้องการของ  
ผู้บริหารหรือหัวหน้างาน จึงได้มีแนวคิดในการที่จะพัฒนาระบบขึ้นมาเพื่อแก้ไขปัญหาดังกล่าว โดย  
ตัวระบบสามารถใช้ช่วยในส่วนของจัดการโครงการ การสรุปรายงานของโครงการ การบันทึกประวัติ  
ของข้อบกพร่อง การบันทึกผลการทดสอบ ซึ่งจากผลการประเมินพบว่าระบบสามารถช่วยเพิ่ม  
ศักยภาพของวิศวกรซอฟต์แวร์ให้มีการวางแผนงานที่ดี และส่งมอบงานที่มีคุณภาพมากขึ้น



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่  
Copyright© by Chiang Mai University  
All rights reserved