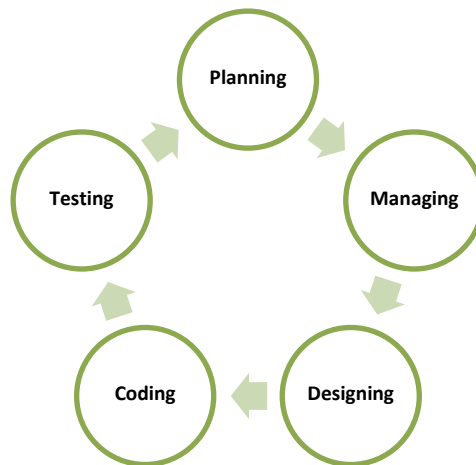


Extreme Programming



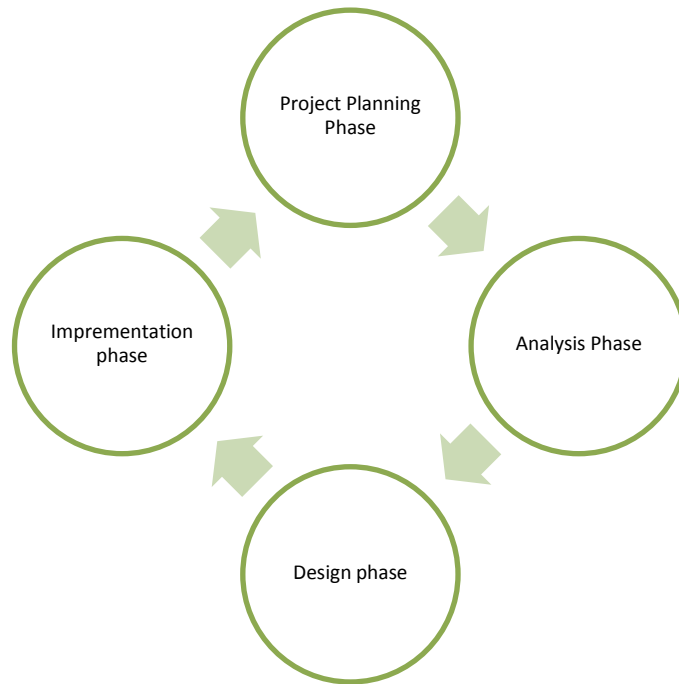
การพัฒนาซอฟต์แวร์ให้เสร็จอย่างรวดเร็ว ไม่เน้นการทำเอกสาร และสามารถตอบสนองต่อการเปลี่ยนแปลงความต้องการของลูกค้าหรือผู้ใช้ระบบได้บ่อยและรวดเร็ว ถึงแม้ว่าการเปลี่ยนแปลงนั้นจะเกิดขึ้นในกระบวนการทำๆ ของการพัฒนาซอฟต์แวร์ก็ตาม ซึ่งเป็นการเพิ่มความพึงพอใจของลูกค้าและทำให้โครงการประสบความสำเร็จ

หลักการของ XP

1. **เน้นการมีส่วนร่วมของลูกค้าและผู้ที่เกี่ยวข้อง** ผู้ที่จะสามารถระบุคุณลักษณะของระบบและฟังก์ชันการทำงานของระบบที่พัฒนาได้ว่าตรงกับวัตถุประสงค์การใช้งานหรือไม่
2. **เน้นการสื่อสารแบบพบปะพูดคุย** วิธีการของ XP จะเน้นการสื่อสารแบบเห็นหน้า (face to face) ได้พบปะพูดคุย ระหว่างบุคคลที่เกี่ยวข้องในโครงการ
3. **เน้นการออกแบบระบบที่เรียบง่าย** ไม่ซับซ้อน เนื่องจากวิธีแบบ XP ไม่เน้นการจัดทำเอกสารมากนัก จึงจำเป็นที่ทีมพัฒนาจะต้องออกแบบระบบและเขียนโปรแกรมให้เรียบง่าย ไม่ซับซ้อน
4. **เน้นการส่งมอบซอฟต์แวร์ที่ทำงานได้ให้กับลูกค้าบ่อยๆ และต่อเนื่อง** ผู้พัฒนาซอฟต์แวร์ควรจะนำเสนอซอฟต์แวร์ที่พัฒนาเสร็จและทำงานได้ให้ กับผู้ใช้หรือลูกค้าตั้งแต่นั่นๆ และส่งให้บ่อยๆ เท่าที่จะทำได้

Software Development Life Cycle (SDLC)

วิธีการแบบทำน้ยได้ มีชื่อเรียกอีกชื่อว่า ตัวแบบน้ำตก (Waterfall model)



ควรรออกแบบ

Flowchart	
Data Model	The ER diagram illustrates the relationships between five main entities: นักศึกษา (Student), อาจารย์ (Teacher), การลงทะเบียน (Registration), รายวิชา (Course), and อาจารย์ผู้สอน (Instructor). Each entity is represented by a rectangle. Attributes are shown in ovals connected to the entities. Relationships are indicated by diamonds with cardinalities. Entity: นักศึกษา - Attributes: ชื่อ (Name), ภาควิชา (Department), คณะ (Faculty), รหัสนักศึกษา (Student ID). Entity: อาจารย์ - Attributes: ชื่อ (Name), ภาควิชา (Department), คณะ (Faculty), รหัสนักเรียน (Teacher ID). Entity: การลงทะเบียน - Attributes: ภาควิชา (Department), ปีที่เข้าศึกษา (Year of Study), รายวิชาที่ลง (Registered Course), วันลงทะเบียน (Registration Date). Entity: รายวิชา - Attributes: ชื่อวิชา (Course Name), ค่าเรียน (Fee), สาขา (Branch), คณะ (Faculty), หน่วยกิต (Credit), รหัสวิชา (Course Code), ชื่อ (Name). Entity: อาจารย์ผู้สอน - Attributes: รหัส (Code), ชื่อ (Name). Relationships: นักศึกษา (1) ลงทะเบียน (ม) - Relationship diamond labeled "มี" (Has). อาจารย์ (1) สอน (ม) - Relationship diamond labeled "สอน" (Teaches). การลงทะเบียน (ม) รายวิชา (ม) - Relationship diamond labeled "มี" (Has). รายวิชา (ม) อาจารย์ผู้สอน (ม) - Relationship diamond labeled "มี" (Has).
หน้าจอ	
รายงาน	

องค์ความรู้การบริหารโครงการ หรือ PMBOK (Project Management Body of Knowledge) ที่มีการ

นิยามโดยสถาบันการบริหารโครงการ หรือ PMI (Project Management Institute) ประกอบด้วย 9 สาขาความรู้ ดังนี้

1. การบริหารขอบเขตโครงการ เป็นการกำหนดและการควบคุมฟังก์ชันของระบบ และขอบเขตของงานของ ทีมพัฒนาโครงการ
2. การบริหารเวลาโครงการ เป็นการสร้างรายละเอียดกำหนดการของงานของโครงการทั้งหมดแล้วติดตาม ความก้าวหน้าของโครงการโดยเทียบกับขั้นแสดงความสำเร็จ ที่กำหนดไว้แล้ว
3. การบริหารต้นทุนโครงการ เป็นการคำนวณการวิเคราะห์ต้นทุนและค่าใช้จ่ายเมื่อเริ่มโครงการ และ ปรับปรุงพร้อมทั้งติดตามค่าใช้จ่ายเมื่อโครงการดำเนินไป
4. การบริหารคุณภาพโครงการ เป็นการสร้างแผนรวมเพื่อประกันคุณภาพของโครงการ ซึ่งประกอบด้วย กิจกรรมการควบคุมคุณภาพของทุกระยะของโครงการ
5. บริหารทรัพยากรมนุษย์ของโครงการ เป็นการสรรหาและการจ้างทีมงาน การฝึกอบรม สร้างแรงจูงใจ การ สร้างทีม และการสร้างบรรยากาศการทำงานที่มีความสุขและมีประสิทธิผล
6. การบริหารการสื่อสารโครงการ เป็นการระบุผู้มีส่วนได้ส่วนเสียทั้งหมดและการสื่อสาร พร้อมทั้งสร้างกลไก และกำหนดการสื่อสารทั้งหมด
7. การบริหารความเสี่ยงโครงการ เป็นการระบุและทบทวนความเสี่ยงต่อความล้มเหลวทั้งหมดของโครงการ และ การพัฒนาแผนลดความเสี่ยง
8. การบริหารการจัดซื้อจัดหาของโครงการ เป็นการพัฒนาคำร้องขอข้อเสนอ หรือ RFP (Request for Proposal) การประเมินผลการเสนอราคา การร่างสัญญา และ การติดตามประสิทธิภาพการทำงานของผู้รับงาน
9. การบริหารการบูรณาการโครงการ เป็นส่วนการบริหารโครงการที่ประสานงานทุกด้านของโครงการเข้าด้วยกันเพื่อให้กระบวนการต่างๆ

สำหรับภาษา UML เวอร์ชัน 2.0 จะประกอบด้วยไดอะแกรมทั้งหมด 13 ไดอะแกรมและแบ่งออกเป็น 3 ประเภทให้เลือกใช้ได้ตามความเหมาะสมดังนี้

1) แผนภาพประเภทโครงสร้าง

- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram

2) แผนภาพประเภทพฤติกรรม

- Activity diagram
- State machine diagram
- Use case diagram

3) แผนภาพประเภทปฏิสัมพันธ์

- Communication diagram
- Interaction overview diagram
- Sequence diagram
- UML timing diagram

โครงสร้างของเอกสารการออกแบบ เอกสารการออกแบบจะประกอบด้วย 5 ส่วนที่สำคัญ คือ

1. วัตถุประสงค์ (purpose) เป็นส่วนที่อธิบายว่าเอกสารนี้เขียนขึ้นเพื่อแสดงรายละเอียดของระบบการทำงาน หรือระบบการทำงานย่อยใด รวมถึงระบุวัตถุประสงค์ของการทำงานในขั้นตอนต่างๆ ทั้งนี้เนื้อหาที่ปรากฏ จะต้องตรงตามความต้องการของระบบที่กำหนดขึ้นจากขั้นตอนการวิเคราะห์ความต้องการ
2. ลำดับความสำคัญ (general priorities) เป็นส่วนที่อธิบายลำดับความสำคัญที่เป็นแนวทางในการออกแบบ เนื่องจากระบบการทำงานแต่ละระบบอาจมีการให้ความสำคัญในการทำงานที่แตกต่างกัน เช่น การทำงานในโมดูลหนึ่งอาจให้ความสำคัญกับการใช้ประโยชน์ของข้อมูลหรือองค์ประกอบบางอย่างในระบบ อย่างเต็มประสิทธิภาพมากกว่าการบำรุงรักษาดูแลระบบ หรือให้ความสำคัญกับด้านการป้องกันรักษา ปลอดภัยของข้อมูลหรือระบบการทำงานอื่นที่เกี่ยวข้องมากที่สุด
3. คำร่างการออกแบบ (outline of the design) เป็นส่วนที่ให้คำอธิบายกว้างๆ เกี่ยวกับการออกแบบที่ผู้อ่าน สามารถมองเห็นภาพรวมในการทำงานของระบบที่จะพัฒนาขึ้นได้อย่างรวดเร็ว ประเด็นการออกแบบ หลัก (major design issues) เป็นส่วนที่อธิบายประเด็นสำคัญที่เป็นปัญหา มีการสรุปรายละเอียดทางเลือกในการพัฒนาระบบการทำงานที่เป็นไปได้
4. รายละเอียดการออกแบบ (other details of the design) เป็นส่วนที่ให้รายละเอียดเกี่ยวกับการออกแบบทั้งหมด ที่ยังไม่ได้ระบุใน

การทดสอบระบบ

การทดสอบระบบ คือ กระบวนการค้นหาข้อผิดพลาดของระบบและประเมินผลการทำงานของระบบเทียบกับข้อกำหนด คุณสมบัติ คุณลักษณะต่างๆ ตามที่ได้กำหนดไว้ในทุกๆ ส่วนการทำงานซึ่งจะทำให้การทำงานในส่วนต่างๆ สามารถทำงานได้ตามวัตถุประสงค์และลดปัญหาข้อผิดพลาดที่อาจเกิดขึ้น อีกทั้งยังช่วยให้สามารถวิเคราะห์ถึงปัญหาข้อผิดพลาดต่างๆ ซึ่งจะช่วยลดความเสียหายที่อาจเกิดขึ้นให้น้อยที่สุด

โดยทั่วไปแล้วในกระบวนการทดสอบระบบนั้นจะมีการดำเนินการ 2 สิ่งควบคู่กันไปตลอดขั้นตอนของการทดสอบระบบ คือ

- การทวนสอบ (verification) เป็นกระบวนการทวนสอบว่าในแต่ละขั้นตอนที่เสร็จสิ้นเป็นไปตามความต้องการที่กำหนดไว้ในแต่ละขั้นตอนหรือไม่
- การทดสอบเพื่อยืนยันผล (validation) เป็นการทดสอบเพื่อยืนยันผลรวมถึงตรวจสอบความสมเหตุสมผลและประเมินผลระบบเมื่อเสร็จสิ้นการดำเนินการในแต่ละขั้นตอนเพื่อให้มั่นใจว่าไม่มีข้อบกพร่องเกิดขึ้น

ขั้นตอนการทดสอบระบบ

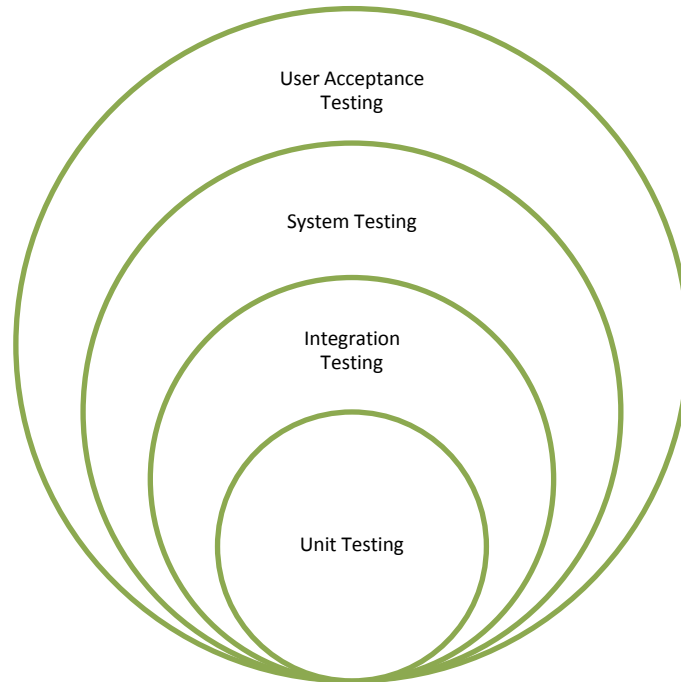


1. การวางแผนการทดสอบ (test planning) เป็นส่วนที่มีความสำคัญที่สุดก่อนที่กระบวนการทดสอบระบบจะเริ่มขึ้น ซึ่งการวางแผนการทดสอบระบบเป็นเสมือนกรอบแนวทางที่ทำให้การดำเนินการกระบวนการทดสอบระบบมีเป้าหมายที่ชัดเจนและสามารถที่จะใช้บอกว่าการดำเนินการกระบวนการทดสอบระบบจะเกิดขึ้นในลักษณะไหน อะไรจะถูกนำมาทดสอบบ้าง รวมถึงใครบ้างที่เป็นผู้ดำเนินการทดสอบและมีขั้นตอนการดำเนินการอย่างไร
2. การวิเคราะห์และการออกแบบ (test analysis and design) เป็นการนำเอารูปแบบกระบวนการ การทำงานและการกระทำต่าง ๆ มาสร้างเป็นเงื่อนไขเพื่อใช้ในการทดสอบระบบรวมไปจนถึงการสร้างกรณีทดสอบซึ่งประกอบด้วยขั้นตอนดังนี้
 - ตรวจสอบสิ่งสำคัญที่ต้องการทดสอบ เช่น ความต้องการของผู้ใช้งาน ความเสี่ยง คุณลักษณะด้านการเชื่อมต่อ เป็นต้น

- วิเคราะห์และประเมินสิ่งที่สามารถทำการทดสอบได้และสิ่งที่ไม่สามารถทำการทดสอบได้จากความต้องการและวัตถุประสงค์
 - กำหนดรูปแบบ ลำดับความสำคัญ วิธีการดำเนินการและเงื่อนไขที่จะใช้ในการทดสอบในแต่ละกรณีออกแบบกรณีทดสอบ
 - กำหนดข้อมูลและสิ่งที่จำเป็นเพื่อใช้สนับสนุนกระบวนการทดสอบ ออกแบบสภาพแวดล้อมที่จะใช้ในกระบวนการทดสอบรวมถึงเครื่องมือและความต้องการด้านต่างๆ
3. การสร้างและการดำเนินการทดสอบ (implementation and execution) คือ การลงมือทดสอบระบบตามระดับชั้นของการทดสอบโดยการนำกรณีทดสอบและข้อมูลต่าง ๆ ที่จำเป็นสำหรับการทดสอบระบบมาสร้างเป็นกระบวนการทดสอบระบบหรือสร้างกลุ่มคำสั่งเพื่อทำการทดสอบภายใต้สภาพแวดล้อมที่กำหนดโดยทั่วไปแล้วระดับชั้นของการทดสอบระบบจะแบ่งออกเป็น 4 ระดับ คือ
- การทดสอบระดับหน่วย
 - การทดสอบระดับบูรณาการ
 - การทดสอบระดับระบบ
 - การทดสอบการยอมรับ
4. การรายงานและประเมินผล (evaluation and report) เป็นขั้นตอนการดำเนินการหลังจากที่ได้มีการดำเนินการกระบวนการทดสอบได้ตามเกณฑ์ที่กำหนดไว้ในแต่ละระดับการทดสอบแล้ว ซึ่งมีรายละเอียดขั้นตอนต่างๆ ดังนี้
- ตรวจสอบและบันทึกผลการทดสอบว่าเป็นไปตามเกณฑ์ที่ระบุไว้ในแผนการทดสอบหรือไม่ - ประเมินผลการทดสอบที่ได้ว่าอยู่ในระดับใดและเป็นไปตามเกณฑ์ที่กำหนดไว้ในแผนการทดสอบระบบหรือไม่
 - จัดทำรายงานสรุปผลการทดสอบระบบสำหรับผู้ที่มีส่วนเกี่ยวข้อง
5. การสิ้นสุดกระบวนการทดสอบ (test closure activities) เป็นขั้นตอนที่เก็บรวบรวมข้อมูลจากการดำเนินการทดสอบที่เสร็จสมบูรณ์แล้วตามช่วงระยะเวลาที่ได้มีการวางแผนไว้ ซึ่งในขั้นตอนนี้จะประกอบด้วยขั้นตอนย่อยๆ ดังนี้
- ตรวจสอบรายการที่จะต้องมีการส่งมอบตามแผน
 - รายงานผลการทดสอบที่เสร็จสมบูรณ์รวมถึงในส่วนการทดสอบที่ยังมีการดำเนินการอยู่
 - จัดทำเอกสารต่าง ๆ ตามที่ได้กำหนดไว้กระบวนการต่างๆ ในขั้นตอนการทดสอบนี้ในบางขั้นตอนหรือบางกิจกรรมสามารถที่จะดำเนินการควบคู่กันไปได้ตามความเหมาะสม

เนื่องจากการวางแผนการทดสอบระบบและการสร้างเอกสารประกอบการทดสอบทั้งก่อนและหลังการทดสอบถือเป็นขั้นตอนที่มีความสำคัญ ดังนั้นในหน่วยนี้จะเน้นรายละเอียดปัจจัยต่างๆ ที่จะนำมาวิเคราะห์เพื่อใช้ประกอบการวางแผนการทดสอบระบบรวมถึงตัวอย่างเอกสารบางส่วนที่ใช้ประกอบการทดสอบระบบ

ระดับชั้นการทดสอบระบบ



1. **การทดสอบระดับหน่วย (unit testing)** อาจเรียกได้หลายชื่อ อาทิ ส่วนประกอบ (component) โมดูล (module) โปรแกรม (program) และฟังก์ชัน (function) โดยมีวัตถุประสงค์เพื่อค้นหา และตรวจสอบการทำงานของโมดูลซอฟต์แวร์ โปรแกรม วัตถุ (objects) คลาส (classes) และอื่นๆ
2. **การทดสอบระดับบูรณาการ (integration testing)** เป็นการทดสอบโดยรวมเอาแต่ละหน่วยย่อยที่ได้ผ่านการทดสอบมาแล้วรวมเข้าด้วยกันทั้ง ฮาร์ดแวร์และซอฟต์แวร์
3. **การทดสอบระดับระบบ (system testing)** เป็นการทดสอบการทำงานของผู้ที่มีหน้าที่เกี่ยวข้อง โดยที่ขอบเขตในการทดสอบต้องมีความชัดเจนในแผนแม่บท (master plan) หรือ แผนการทดสอบระดับ (level test plan) ของแต่ละการทดสอบ ซึ่งในการทดสอบระดับระบบนั้นสภาพแวดล้อมในการทดสอบควรจะสอดคล้องกับสภาพของการใช้งานจริงให้มากที่สุด
4. **การทดสอบการยอมรับจากผู้ใช้งาน (user acceptance testing)** โดยปกติแล้วจะเป็นการทดสอบการทำงานของระบบร่วมกันกับผู้ใช้งานโดยที่วัตถุประสงค์ของการทดสอบการยอมรับของผู้ใช้นั้นเพื่อสร้างความเชื่อมั่นและความคุ้นเคย