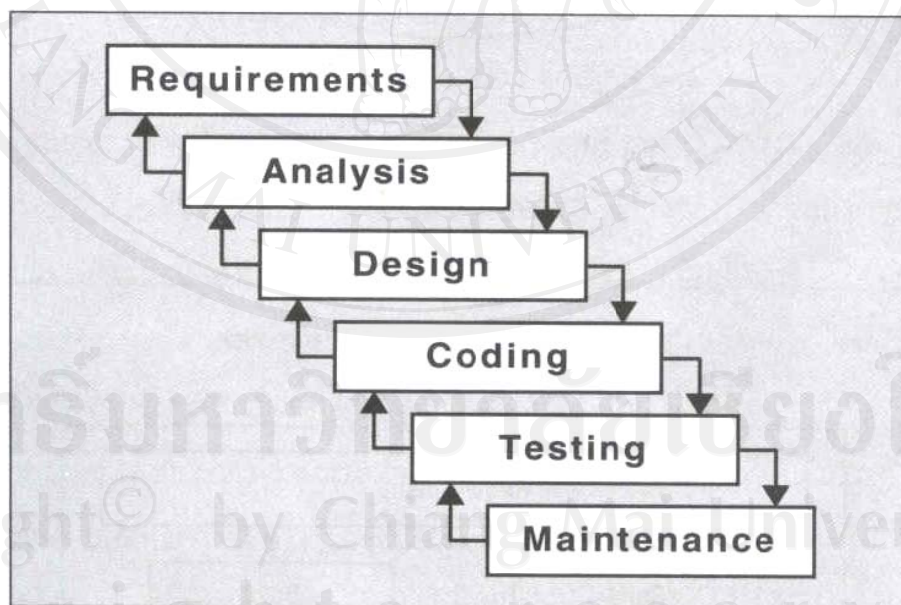


บทที่ 3

ระเบียบวิธีการศึกษา

วิธีการศึกษาวิจัย การพัฒนาแบบจำลองเพื่อการศึกษาโดยใช้เทคโนโลยีความจริงเสริม กรณีศึกษาแบบหลายมาร์คเกอร์ โดยอาศัยขั้นตอนตามกระบวนการพัฒนาซอฟต์แวร์แบบน้ำตก (Waterfall model) แบ่งออกเป็น 6 ขั้นตอนดังนี้คือ

1. ศึกษาความต้องการของระบบ
2. วิเคราะห์และวางแผน
3. การออกแบบระบบ
4. การพัฒนาระบบ
5. การทดสอบระบบ
6. การส่งมอบและการบำรุงรักษา



รูปที่ 3.1 แสดงขั้นตอนการพัฒนาซอฟต์แวร์ตามแบบจำลองน้ำตก

3.1 ศึกษาความต้องการของระบบ

การศึกษาความต้องการของระบบเป็นขั้นตอนแรกที่ต้องทำอย่างรอบคอบและชัดเจนที่สุด ดังนั้น ในการพัฒนาแบบจำลองเพื่อการศึกษาโดยใช้เทคโนโลยีความจริงเสริม กรณีศึกษาแบบหลายมาร์คเกอร์ จึงมีขั้นตอนการหาความต้องการดังนี้

Requirements Definition นิยามความต้องการ เนื่องจากความต้องการที่ได้มาอาจยังไม่ครบถ้วนจึงต้องมีการตีความความต้องการให้ออกมาเป็นรูปธรรมสามารถเข้าใจได้ง่าย ไม่คลุมเครือ โดยการศึกษาหาความต้องการของโปรแกรมการพัฒนาแบบจำลองเพื่อการศึกษาโดยใช้เทคโนโลยีความจริงเสริม กรณีศึกษาแบบหลายมาร์คเกอร์ โดยได้เข้าไปศึกษาข้อมูลเหล่านี้ คือ

- สัมภาษณ์ผู้เชี่ยวชาญทางด้านแมลง, นักศึกษาสาขาวิชาชีววิทยา, ครูผู้สอนระดับประถมศึกษา และบุคคลทั่วไป
- ศึกษาเทคโนโลยีใหม่ที่สามารถนำมาประยุกต์เข้ากับการเรียน
- ศึกษาเทคโนโลยีและเทคนิคต่าง ๆ ที่เกี่ยวข้องเช่น เทคโนโลยีความจริงเสริม, ชุดควบคุมการแสดงผล 3 มิติ และระบบฐานข้อมูล Microsoft Access
- ศึกษาพฤติกรรมของผู้ใช้งาน
- ศึกษาข้อกำหนดและความเป็นไปได้ในการนำระบบไปใช้งานจริง

Assessment การประเมินต้องการและเงินทุนที่มีอยู่ในโครงการ เช่นเงินทุนในโครงการนี้มีมากน้อยเพียงใด และมีระยะเวลาเท่าใดจึงจะเหมาะสม

- ประเมินการทางด้าน งบประมาณ เวลา และ บุคลากรผู้เชี่ยวชาญในระบบ
- ผลลัพธ์ที่ต้องการ จากกระบวนการนี้ คือ เอกสารข้อกำหนดความต้องการของระบบ ซึ่งจำเป็นต้องนำไปใช้ในกระบวนการออกแบบ

3.2 วิเคราะห์และวางแผน

หลังจากที่ระบุความต้องการของลูกค้าได้แล้ว ทางทีมงานจะเริ่มออกแบบระบบให้มีความสอดคล้องกับความต้องการที่มี โดยจัดหาเทคโนโลยีและการออกแบบที่สัมพันธ์กับความต้องการของลูกค้า

Project Planning การออกแบบเอกสารจัดการโครงการ ระบุถึง ขั้นตอนการทำงาน และโครงสร้างในองค์กร ผู้ที่มีส่วนร่วมทั้งหมดที่เกี่ยวข้องกับการทำโครงการนี้

Solution Analysis การหาแนวทางการแก้ไขปัญหา การวิเคราะห์ความเสี่ยงที่อาจเกิดขึ้นกับโครงการ โดยระบุทางเลือกแก้ไขปัญหาที่เหมาะสมที่สุด

Conference Room Pilot การประชุมเพื่อวางแผน ในการประชุมจะมีการวางแผนทุก ๆ สัปดาห์และการส่งมอบงานอย่างน้อยเดือนละ 1 ครั้ง

ผลลัพธ์ที่ต้องการ จากกระบวนการนี้ คือ เอกสารข้อกำหนดวางแผนการทำงานของระบบ ซึ่งจำเป็นต้องนำไปใช้ในกระบวนการทำงานทั้งหมดในโครงการ

3.3 ออกแบบระบบ

เมื่อศึกษาความต้องการ การวิเคราะห์และวางแผน เป็นที่เรียบร้อยแล้ว จึงเริ่มการทำงานในส่วนของการออกแบบ และการสร้างตัวต้นแบบที่ตอบสนองต่อความต้องการ โดยมีการทำงานตามแผนที่วางเอาไว้

การออกแบบระบบทั้งในส่วนการออกแบบอัลกอริทึมที่ใช้ในการทำงาน และทำการออกแบบระบบตามการออกแบบซอฟต์แวร์เชิงวัตถุด้วยยูเอ็มแอล (UML) ซึ่งประกอบด้วย

1. ยูสเคสไดอะแกรม (Use Case Diagram) ใช้สำหรับออกแบบส่วนของการทำงานหลัก และผู้ที่มีส่วนเกี่ยวข้องกับระบบ รวมถึงความสัมพันธ์กับระบบภายนอก
2. คลาสไดอะแกรม (Class Diagram) ใช้สำหรับออกแบบคลาส โครงสร้างของคลาส และความสัมพันธ์ระหว่างคลาสที่มีในระบบ
3. สเตทไดอะแกรม (State Diagram) ใช้สำหรับออกแบบสถานะ ของการทำงานหลักที่เกิดขึ้นในระบบ
4. ซีควเ็นซ์ไดอะแกรม (Sequence Diagram) ใช้สำหรับออกแบบและอธิบายการทำงานในส่วนที่สำคัญต่าง ๆ ของแต่ละคลาสที่มีความเชื่อมโยงกัน

ผลลัพธ์ที่ต้องการ จากกระบวนการนี้ คือ เอกสารข้อกำหนดการออกแบบระบบ ที่จำเป็นสำหรับการพัฒนาระบบ และผลตอบกลับไปยังการกำหนดความต้องการ คือ ข้อมูลการออกแบบที่สัมพันธ์กับความต้องการ โดยจะนำไปตรวจสอบว่าการออกแบบนั้น ตอบสนองกับความต้องการหรือไม่ หากไม่เป็นไปตามความต้องการก็จะต้องแก้ไขการออกแบบนั้น หรือหากไม่สามารถแก้ไขการออกแบบได้ อาจจะต้องปรับปรุงข้อกำหนดความต้องการให้สอดคล้องกันมากที่สุด

3.4 พัฒนาระบบ

เป็นกระบวนการในการ พัฒนาระบบให้ได้ตามข้อกำหนดที่ได้ออกแบบเอาไว้และประกอบเข้าด้วยกัน เพื่อให้ออกมาเป็นซอฟต์แวร์ที่สามารถใช้งานได้ ซึ่งการดำเนินงานในกระบวนการนี้ มีรายละเอียดดังนี้

เครื่องมือสำหรับการพัฒนา

1. ชุดพัฒนาซอฟต์แวร์ เออาร์ทูลคิต (ARToolKit 2.5.7)
2. ชุดแสดงผลกราฟิก 3มิติ ไอริช เอนจิน (Irrlicht Engine 1.4.1)
3. ซอฟต์แวร์สำหรับการพัฒนาแบบเบ็ดเสร็จ ไมโครซอฟต์สตูดิโอ 2005 (Microsoft Studio 2005)
4. คอมพิวเตอร์พร้อมเว็บแคม

ขั้นตอนการพัฒนา

1. ติดตั้งเครื่องมือสำหรับการพัฒนาระบบ บนคอมพิวเตอร์
2. ศึกษาเอกสารการออกแบบระบบ
3. แบ่งการพัฒนาออกเป็น ส่วน ๆ ตามเอกสารการออกแบบ
4. ออกแบบคำสั่งของโปรแกรมทีละส่วน
5. สร้างไฟล์คำสั่งและเขียนคำสั่งทีละส่วน
6. ทำการคอมไพล์ ทดสอบการทำงานของแต่ละส่วน
7. นำแต่ละส่วนมาเชื่อมกัน แล้วคอมไพล์ตามลำดับ อีกครั้ง
8. จัดทำเอกสารประกอบโปรแกรม

ผลลัพธ์ที่ต้องการ จากกระบวนการนี้ คือ ระบบที่ถูกพัฒนาแล้ว พร้อมเอกสารประกอบโปรแกรมเพื่ออธิบายรายละเอียดการทำงานในแต่ละส่วน ซึ่งการทำงานของระบบอาจจะยังไม่

สมบูรณ์ โดยจะนำไปใช้ในกระบวนการทดสอบต่อไป ส่วนผลตอบกลับไปยังการออกแบบ คือ ข้อมูลการพัฒนาที่สัมพันธ์กับการออกแบบ โดยจะนำไปตรวจสอบว่าการพัฒนา ดำเนินไปตามการ ออกแบบหรือไม่ หากไม่เป็นตามการออกแบบ ก็จะต้องแก้ไขให้สอดคล้องกัน ก่อนที่จะเข้าสู่การ ทดสอบต่อไป

3.5 ทดสอบระบบ

เป็นกระบวนการในการทดสอบระบบซอฟต์แวร์ที่พัฒนาได้เพื่อให้ทำงานได้อย่างถูกต้อง ตรงตามความต้องการ โดยนำซอฟต์แวร์มาทดสอบการทำงานทั้ง การทดสอบระดับหน่วยย่อย การ ทดสอบการทำงานร่วมกันของระบบ และการทดสอบความสามารถในการใช้งานตาม สภาพแวดล้อมจริง แล้วนำปัญหาจากการทดสอบมาปรับปรุงแก้ไขเพื่อให้การทำงานสมบูรณ์ ซึ่ง วิธีการดำเนินงานในกระบวนการนี้ มีรายละเอียดดังนี้

1. กำหนดขั้นตอนในการทดสอบ
2. จัดเตรียมกรณีทดสอบและข้อมูลทดสอบ
3. เตรียมหน่วยย่อยของระบบที่ต้องการทดสอบ
4. ทำการทดสอบในระดับหน่วยย่อย พร้อมสรุปผลการทดสอบ
5. ทำการทดสอบในระดับของการผสานหน่วยย่อย พร้อมสรุปผลการทดสอบ
6. ทำการทดสอบระบบภายใต้สภาพแวดล้อมการใช้งานจริง พร้อมสรุปผลการทดสอบ
7. จัดทำเอกสารรายงานการทดสอบระบบ

ผลลัพธ์ที่ต้องการ จากกระบวนการนี้ คือ เอกสารรายงานการทดสอบระบบ ส่วนผลตอบ กลับไปยังกระบวนการพัฒนา จะเกิดขึ้นเมื่อพบปัญหาที่ทำให้ไม่สามารถผ่านการทดสอบตามผลที่ คาดหวัง ซึ่งต้องนำผลลัพธ์กลับไปหาสาเหตุและแก้ไขให้สมบูรณ์ แล้วทำการทดสอบซ้ำอีกครั้งจน ได้ผลลัพธ์ตามที่คาดหวัง

3.6 สังมอบและบำรุงรักษาระบบ

ในการนำระบบไปใช้งาน มักจะเกิดการเปลี่ยนแปลงหรือปัญหาขึ้นกับระบบ ซึ่งเป็น ไป ตามสภาพแวดล้อมการทำงานที่แตกต่างกัน ดังนั้นกระบวนการนี้ จะเป็นการวางแผนบำรุงรักษา ระบบ โดยจัดเตรียมแผนรองรับการเปลี่ยนแปลงและปัญหาที่อาจเกิดขึ้นกับระบบหลังจากที่

นำไปใช้งานแล้ว อย่างเป็นระบบและมีประสิทธิภาพ ซึ่งสิ่งต่าง ๆ ที่เกิดขึ้นจะถูกนำมาแก้ไข ปรับปรุงทันทีหรือบันทึกไว้เพื่อแก้ไขปรับปรุงในซอฟต์แวร์รุ่นถัดไป ตามการอนุมัติของผู้มีอำนาจของทีมผู้พัฒนา ซึ่งจะถูกรวบรวมไว้ในส่วนต่าง ๆ ของระบบที่เกี่ยวข้องกันและพิจารณาถึงความเหมาะสมในการเปลี่ยนแปลง ตามแผนงานที่วางไว้

ผลลัพธ์ที่ต้องการ จากกระบวนการนี้ คือ แผนการบำรุงรักษาระบบ โดยจะต้องสอดคล้องกับการนำไปใช้งานจริง และผลตอบกลับจากกระบวนการนี้ จะนำไปใช้ในการตรวจสอบกระบวนการก่อนหน้า เพื่อค้นหาสาเหตุของปัญหา แล้วแก้ไขให้สมบูรณ์ในซอฟต์แวร์รุ่นปัจจุบัน หรือเก็บไว้แก้ไขให้สมบูรณ์ในซอฟต์แวร์รุ่นถัดไป

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved