



1.1 ความเป็นมาและความสำคัญของปัญหา

ขั้นตอนหนึ่งที่สำคัญในการพัฒนาซอฟต์แวร์ เพื่อให้ซอฟต์แวร์ที่พัฒนาขึ้นนั้นมีประสิทธิภาพ น่าเชื่อถือ และถูกต้องตามความต้องการของผู้ใช้งาน คือ ขั้นตอนการทดสอบ (Testing) การทดสอบซอฟต์แวร์มีได้หลายระดับตั้งแต่ การทดสอบระดับหน่วย (Unit testing) การทดสอบการรวม (Integration testing) การทดสอบย้อนกลับ (Regression testing) การทดสอบการใช้ได้ (Validation testing) และการทดสอบระบบ (System testing) ในการทดสอบทุกระดับสิ่งที่จำเป็นสำหรับให้ผู้ทดสอบ (Tester) ใช้ในการทดสอบคือกรณีทดสอบ กรณีทดสอบที่ดีนั้นจะต้องสามารถค้นหาข้อผิดพลาดที่ยังไม่ถูกค้นพบในซอฟต์แวร์ที่พัฒนานั้นได้ การออกแบบกรณีทดสอบในแต่ละระดับแบ่งออกเป็นสองประเภท คือแบบแบล็กบ็อกซ์ (Black-box testing) และแบบไวท์บ็อกซ์ (White-box testing)

การออกแบบกรณีทดสอบโดยวิธีการของแบล็กบ็อกซ์นั้นนิยมใช้ในทุกะดับของการทดสอบ ซึ่งในการสร้างกรณีทดสอบในแบบแบล็กบ็อกซ์ ผู้ทดสอบต้องทำขั้นตอนของการแยกชั้นสมมูล (Equivalence class partitioning) คือการแยกข้อมูลตามประเภท และเงื่อนไขของข้อมูลเข้า (Input) เพื่อให้ได้คลาสของข้อมูลเข้า (Input class) ที่ถูกต้อง (Valid) และ คลาสของข้อมูลเข้าที่ไม่ถูกต้อง (Invalid) และต้องทำการวิเคราะห์ค่าขอบเขต (Boundary value analysis) ของข้อมูลเข้าเพื่อให้กรณีทดสอบที่สร้างขึ้นนั้นครอบคลุมค่าบริเวณขอบเขต สำหรับทุกคลาสของข้อมูลเข้าที่ได้ทำการแยกชั้นสมมูลไว้ ซึ่งการแยกชั้นสมมูล และการสร้างกรณีทดสอบให้ครอบคลุมทุกอย่างชั้นสมมูลและทุก ๆ ค่าบริเวณขอบเขตเป็นขั้นตอนที่ใช้เวลาเป็นอย่างมากสำหรับผู้ทดสอบ โดยเฉพาะสำหรับการทดสอบที่มีข้อมูลเข้าเป็นจำนวนมาก

ในปัจจุบันยังไม่มีเครื่องมือช่วยแยกชั้นสมมูล และสร้างกรณีทดสอบเพื่อให้ครอบคลุมทุกชั้นสมมูล ดังนั้นงานวิจัยนี้จึงพัฒนาเครื่องมือที่ช่วยแยกชั้นสมมูล สร้างกรณีทดสอบเพื่อให้ครอบคลุมทุกชั้นสมมูล และจัดการกรณีทดสอบ เพื่อให้การเก็บข้อมูลกรณีทดสอบ การติดตาม ตรวจสอบ และแก้ไขข้อผิดพลาดต่าง ๆ ในขั้นตอนของการทดสอบซอฟต์แวร์สามารถทำได้สะดวกยิ่งขึ้น และเป็นมาตรฐานเดียวกัน

1.2 วัตถุประสงค์

เพื่อพัฒนาระบบจัดการกรณีทดสอบซอฟต์แวร์ โดยระบบจะประกอบด้วยเครื่องมือช่วยในการสร้างกรณีทดสอบ และฐานข้อมูลสำหรับเก็บข้อมูลกรณีทดสอบ

1.3 ขอบเขตของการวิจัย

- 1) ผู้ทดสอบสามารถบันทึกรายละเอียดของข้อมูลกรณีทดสอบเข้าสู่ระบบ โดยระบบจะรับข้อมูลดังกล่าวไปทำการออกรายงานสรุปผลการทดสอบ รวมทั้งรายงานแสดงความก้าวหน้าในการทดสอบ และการแก้ไขข้อผิดพลาดของโปรแกรม
- 2) กรณีทดสอบที่พัฒนาขึ้น และใช้งานภายในระบบ มีรูปแบบที่ปรับปรุงจากมาตรฐานของ IEEE Std 829-1998 [1] (ดูรายละเอียดใน ภาคผนวก ก)
- 3) ในส่วนการสร้างกรณีทดสอบ ระบบจะรับประเภทข้อมูล และลักษณะเฉพาะ หรือเงื่อนไขของข้อมูลจากผู้ทดสอบมาใช้ในการสร้างกรณีทดสอบ
- 4) ผู้ใช้จะสามารถจัดการ เพิ่ม ลด แก้ไข หรือปรับปรุงกรณีทดสอบที่สร้างขึ้นมาได้
- 5) เครื่องมือดังกล่าวทำงานภายใต้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 98 หรือสูงกว่า (Microsoft Windows 98 or higher)

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

- 1) ศึกษากระบวนการจัดการกรณีทดสอบ และเครื่องมือ ที่มีอยู่ในปัจจุบัน
- 2) ศึกษาวิธีการทดสอบซอฟต์แวร์ วิธีการสร้างกรณีทดสอบ และมาตรฐานสำหรับกรณีทดสอบ
- 3) ออกแบบฐานข้อมูล สำหรับระบบจัดการกรณีทดสอบ
- 4) พัฒนาส่วนรับข้อมูลกรณีทดสอบ และส่วนจัดการข้อมูลกรณีทดสอบ
- 5) พัฒนาตัวสร้างชั้นสมมูล
- 6) พัฒนาตัวสร้างกรณีทดสอบ
- 7) รวบรวมข้อมูลกรณีทดสอบ ตัวสร้างชั้นสมมูล และตัวสร้างกรณีทดสอบ
- 8) ทดสอบผลการรวม และแก้ไขข้อผิดพลาด
- 9) จัดทำวิทยานิพนธ์และคู่มือการใช้เครื่องมือ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ช่วยลดภาระผู้ทดสอบ เนื่องจากระบบสามารถสร้างชั้นสมมูล และกรณีทดสอบที่ครอบคลุมทุกชั้นสมมูล สำหรับวิธีการแบบแบล็กบ็อกซ์ได้โดยอัตโนมัติ
- 2) ช่วยเก็บบันทึก รวบรวมข้อมูลเกี่ยวกับกรณีทดสอบ และผลการทดสอบ ซึ่งสามารถนำมาใช้ติดตามผลการทดสอบ ความก้าวหน้าของการทดสอบ คุณภาพของซอฟต์แวร์ที่พัฒนา และการทำงานของผู้ทดสอบแต่ละคน

- 3) มีการเก็บข้อมูลกรณีทดสอบที่ เป็นมาตรฐานเดียวกันทำให้ผู้ทดสอบสามารถใช้งานหรือแก้ไขปรับปรุงกรณีทดสอบได้ง่ายขึ้น
 - 4) กรณีทดสอบที่บันทึกไว้สามารถนำไปใช้ในการทดสอบซอฟต์แวร์ที่จะพัฒนาขึ้นมาใหม่ในอนาคตได้
- สามารถนำเครื่องมือที่พัฒนาขึ้นไปใช้งานร่วมกับเครื่องมือที่ทำงานเกี่ยวข้องกันได้ เช่น เครื่องมือที่ช่วยวางแผนการทดสอบ