



1.1 ความเป็นมาและความสำคัญของปัญหา

การทดสอบซอฟต์แวร์เป็นกระบวนการที่สำคัญในการพัฒนาซอฟต์แวร์ให้มีคุณภาพ เนื่องจากการค้นหาข้อผิดพลาดที่เกิดขึ้นเพื่อนำไปแก้ไขก่อนที่จะถูกนำไปใช้งาน การทดสอบที่ดีนั้นจำเป็นจะต้องมีการวางแผนการทดสอบในทุก ๆ ขั้นตอน ไม่ว่าจะเป็นการทดสอบระดับหน่วย (Unit Testing) การทดสอบการรวม (Integration Testing) การทดสอบระบบ (System Testing) และการทดสอบการยอมรับ (Acceptance Testing) เพื่อให้สามารถติดตามและควบคุมได้ง่าย อีกทั้งยังสามารถทำการปรับแต่งได้หากการทดสอบมีแนวโน้มว่าจะเสร็จไม่ทันตามกำหนดเวลา อันจะทำให้สามารถรับประกันได้ว่าซอฟต์แวร์ที่ผลิตออกมา ได้ผ่านการทดสอบมาอย่างครบถ้วนสมบูรณ์ และสามารถส่งมอบให้กับลูกค้าได้ทันตามกำหนด แต่องค์กรส่วนใหญ่มักจะละเลยกระบวนการวางแผนการทดสอบนี้เนื่องจากการไม่ต้องการสูญเสียเวลา แรงงาน และค่าใช้จ่ายกับกระบวนการดังกล่าว องค์กรมักจะทำให้ผู้พัฒนาทดสอบควบคู่ไปกับการเขียนโปรแกรมเพื่อเป็นการประหยัดค่าใช้จ่าย ซึ่งอาจทำให้ซอฟต์แวร์ที่ได้มีคุณภาพไม่เพียงพอ เนื่องจากข้อผิดพลาดยังไม่ถูกค้นพบเพราะขาดการควบคุมที่ดี

การวางแผนการทดสอบซอฟต์แวร์ เริ่มต้นด้วยการกำหนดจุดมุ่งหมายและขอบเขตของการทดสอบ เพื่อให้กลุ่มผู้ทดสอบทราบว่าควรจะไปทดสอบไปในทิศทางใด หลังจากนั้นจะเริ่มทำการทดสอบที่ระดับต่าง ๆ โดยเริ่มจากการทดสอบที่ระดับหน่วยก่อน ซึ่งองค์กรจะต้องกำหนดกิจกรรมที่จะเกิดขึ้นแล้วมอบหมายให้กับกลุ่มผู้ทดสอบเพื่อทำการออกแบบกรณีทดสอบ (Test case) และดำเนินการทดสอบตามที่ได้ออกแบบไว้ เมื่อสิ้นสุดการทดสอบระดับหน่วยก็จะเริ่มการทดสอบที่ระดับสูงขึ้นจนถึงการทดสอบระบบโดยเน้นไปที่ความต้องการของผู้ใช้ก่อนที่จะส่งมอบซอฟต์แวร์ให้กับลูกค้า ซึ่งจะสังเกตได้ว่าในแต่ละระดับของการทดสอบมีขั้นตอนการทำงานที่ค่อนข้างซับซ้อน และต้องจัดการกับข้อมูลเป็นจำนวนมาก จึงจำเป็นต้องมีการวางแผนการทดสอบเป็นอย่างดีเพื่อให้การทำกิจกรรมต่าง ๆ สามารถบรรลุเป้าหมายตามที่กำหนดไว้ได้ ดังนั้น งานวิจัยนี้จึงมีแนวความคิดในการออกแบบและพัฒนาเครื่องมือเพื่อรองรับกระบวนการวางแผนการทดสอบ เพื่อให้เป็นกระบวนการที่สามารถควบคุมได้ มีประสิทธิภาพและ ไม่สูญเสียค่าใช้จ่ายมาก อันจะทำให้ซอฟต์แวร์ที่ได้มีคุณภาพดียิ่งขึ้น

1.2 วัตถุประสงค์

เพื่อพัฒนาเครื่องมือที่มีความสามารถในการจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์

1.3 ขอบเขตการวิจัย

- 1) เครื่องมือนี้จะใช้วางแผนการทดสอบเฉพาะการทดสอบระดับหน่วย การทดสอบการรวม และการทดสอบระบบเท่านั้น
- 2) ผู้ใช้เครื่องมือนี้จะต้องวางกำหนดการในการทำกิจกรรมเกี่ยวกับการวางแผนการทดสอบโดยระบุวันที่เริ่มต้น ระยะเวลา ความเกี่ยวข้องกับกิจกรรมอื่น¹ และทรัพยากรที่ใช้ ซึ่งเครื่องมือจะไม่จัดการกับข้อขัดแย้งในการจัดสรรทรัพยากรสำหรับแต่ละกิจกรรม
- 3) เครื่องมือนี้ช่วยในการติดตามสถานะการทำงานของขั้นตอนต่าง ๆ ที่เกิดจากการวางแผนการทดสอบจากเอกสารที่เกิดขึ้นจากการทดสอบ
- 4) เครื่องมือนี้สามารถเชื่อมโยงไปยังเอกสารเกี่ยวกับการทำกิจกรรมต่าง ๆ ในการทดสอบที่บันทึกไว้ในฐานข้อมูลได้เช่น แผนการทดสอบ เอกสารที่ระบุกรณีทดสอบ เป็นต้น
- 5) เครื่องมือสามารถออกรายงานที่เป็นแผนการทดสอบซอฟต์แวร์โดยอิงกับมาตรฐาน IEEE Std.829-1998 โดยให้ผู้ใช้สามารถปรับเปลี่ยนรูปแบบของเอกสารได้ตามที่เครื่องมือได้จัดสรรไว้
- 6) ในส่วนของการทดสอบระบบ เครื่องมือจะช่วยในการทดสอบตามหน้าที่เท่านั้น สำหรับการทดสอบอื่น ๆ เช่น การทดสอบประสิทธิภาพ การทดสอบความปลอดภัย เป็นต้น ผู้ใช้จะเป็นผู้กำหนดกิจกรรมและดำเนินการทดสอบเอง
- 7) เครื่องมื่อดังกล่าวทำงานภายใต้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 98 หรือสูงกว่า (Microsoft Windows 98 or higher)

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

- 1) ศึกษามาตรฐานของเอกสารที่ใช้ในการทดสอบที่มีอยู่ในปัจจุบัน
- 2) ศึกษาขั้นตอนที่ใช้ในการวางแผนการทดสอบ และรายละเอียดเกี่ยวกับระดับของการทดสอบ
- 3) ศึกษาเครื่องมือที่เกี่ยวข้องกับการวางแผนการทดสอบ
- 4) ออกแบบตัวจัดการสร้างแผนการทดสอบ ตัวจัดกำหนดการ และตัวจัดการเอกสารที่เกี่ยวข้องกับการทดสอบ
- 5) พัฒนาเครื่องมือตามที่ได้ออกแบบไว้ในข้อที่ 4
- 6) ทดสอบเครื่องมือที่พัฒนา
- 7) เขียนวิทยานิพนธ์ และคู่มือการใช้เครื่องมือ

¹ ประกอบด้วย เริ่มต้นพร้อมกัน (Start-to-Start) เริ่มต้นเมื่อสิ้นสุด (Start-to-Finish) สิ้นสุดเมื่อเริ่มต้น (Finish-to-Start) และสิ้นสุดพร้อมกัน (Finish-to-Finish)

1.5 ประโยชน์ที่ได้รับ

- 1) ทำให้องค์กรที่พัฒนาซอฟต์แวร์ได้นำเครื่องมือนี้ไปใช้งานเพื่อที่สามารถติดตามและควบคุมกระบวนการทดสอบได้ อันจะทำให้ซอฟต์แวร์ที่ได้มีคุณภาพดีขึ้น
- 2) เครื่องมือที่พัฒนานี้สามารถนำไปใช้กับระบบจัดการกรณีทดสอบ หรือเครื่องมือที่ใช้ในการทดสอบซอฟต์แวร์อื่น ๆ ได้
- 3) เป็นแนวทางในการพัฒนาการสร้างเครื่องมือเพื่อช่วยองค์กรในการจัดการขั้นตอนต่าง ๆ ในการพัฒนาซอฟต์แวร์ให้มีประสิทธิภาพมากขึ้น