

การทดสอบซอฟต์แวร์ตามกรอบการทำงาน สำหรับห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ

วรพรรณ พุกานันท์ อรุณดา แสงเนตร อติติยา กิตติศรีวรพันธุ์ และ วันทนา ศรีสมบูรณ์

สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ชลบุรี

Emails: se56160090@gmail.com, se56160209@gmail.com, se56160420@gmail.com, wantana.w58@gmail.com

บทคัดย่อ

ในปัจจุบันการทดสอบซอฟต์แวร์เป็นกระบวนการที่สำคัญในวงจรการพัฒนาซอฟต์แวร์ ซึ่งจะช่วยให้การประหยัดทรัพยากรที่ต้องใช้ในการดำเนินการโครงการ ทั้งด้านงบประมาณและเวลา ในกรณีที่สามารถพบข้อผิดพลาดหรือปัญหาที่เกิดขึ้นตั้งแต่ระยะแรก ๆ ของโครงการ โครงการนี้จึงนำเสนอการทำการทดสอบการทำงานการทดสอบซอฟต์แวร์สำหรับห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ (Information System Engineering Research Laboratory: ISERL) ในกระบวนการทดสอบแบบ V-Model ซึ่งระบบสารสนเทศที่ใช้เป็นเครื่องมือในการทดสอบ คือ ระบบจัดการการทดสอบซอฟต์แวร์ (Software Testing Management System: STMs) และระบบสารสนเทศที่ทดสอบ คือ ระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณ

ABSTRACT

Software Testing is one of the important filed in software engineering it has a very significance role. A defect free product will be need by customers. Exploratory manual software testing is essential to find defects early when they are less costly to remediate. This study utilize a framework and the Software Testing Management System: STMs developed by Software Engineering students, Burapha University that covers all of the major areas of software testing. In this experiment, there were 2 pilot systems was selected from the projects in Information System Engineering Research Laboratory: ISERL. The selected systems were 1) Project Plan and 2) Performance and Budget Monitoring system.

คำสำคัญ-- กรอบการทำงานการทดสอบซอฟต์แวร์สำหรับ ISERL; การทดสอบซอฟต์แวร์; การประกันคุณภาพ

1. บทนำ

การพัฒนาซอฟต์แวร์เพื่อใช้ในองค์กร มีเป้าหมายเพื่อให้ได้ซอฟต์แวร์ที่มีประสิทธิภาพ และตอบสนองความต้องการใช้งานของผู้ใช้ได้ แต่ด้วยปัจจัยแวดล้อมหลายประการ เช่น ระยะเวลาการพัฒนาซอฟต์แวร์ที่จำกัด งบประมาณที่ได้รับไม่เพียงพอ การสื่อสารระหว่างนักพัฒนาระบบและผู้ใช้ไม่สอดคล้องกัน เป็นต้น ล้วนมีผลต่อคุณภาพซอฟต์แวร์ที่พัฒนาขึ้นในภาพรวม เนื่องจากผู้พัฒนามุ่งเน้นที่จะพัฒนาซอฟต์แวร์ให้แล้วเสร็จตามกำหนดการ ภายใต้เงื่อนไขที่ไม่เอื้ออำนวย จึงมักตัดสินใจลดขั้นตอนสำคัญในวงจรการพัฒนาซอฟต์แวร์บางส่วนออกไป โดยเฉพาะอย่างยิ่งการทดสอบซอฟต์แวร์ จึงมองไม่เห็นความสำคัญของการทดสอบซอฟต์แวร์ ประกอบกับผู้ใช้ไม่มีความรู้และความเชี่ยวชาญในเรื่องวงจรการพัฒนาซอฟต์แวร์ที่ดีพอ ทำให้พบข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นระหว่างการใช้งาน ซึ่งส่งผลให้ประสิทธิภาพการทำงานและประสิทธิผลขององค์กรลดลง และอาจก่อให้เกิดปัญหารวมทั้งค่าใช้จ่ายอื่น ๆ ตามมาอีกมากมาย ดังนั้น การทดสอบซอฟต์แวร์ เป็นกระบวนการสำคัญที่สุดที่องค์กรจะต้องให้ความสำคัญ เพื่อให้เกิดความเชื่อมั่นในการใช้งานซอฟต์แวร์ และเพื่อให้การลงทุนทั้งหมดเป็นไปอย่างคุ้มค่ามากที่สุด

ผู้ดำเนินโครงการจึงนำเสนอกรอบการทำงานการทดสอบซอฟต์แวร์เพื่อใช้งานสำหรับห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ ซึ่งระบบสารสนเทศที่นำมาเป็นกรณีศึกษา คือ ระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณ ในโครงการให้บริการซอฟต์แวร์ผ่านอินเทอร์เน็ต วิทยาลัยพระจอมเกล้า จังหวัดเพชรบุรี

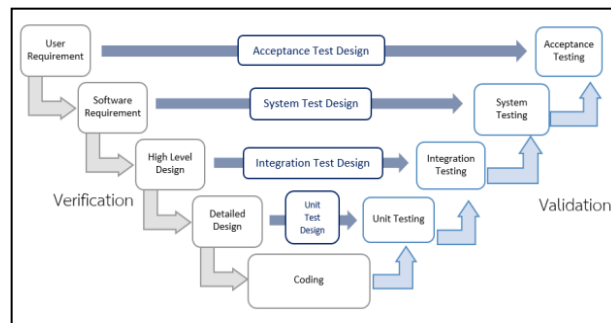
2. งานวิจัยหรือบทความที่เกี่ยวข้อง

ในการดำเนินการโครงการครั้งนี้ ผู้ดำเนินโครงการได้ทำการศึกษาทฤษฎีที่เกี่ยวข้องกับการทดสอบ เพื่อให้มีความเข้าใจ และก่อให้เกิดแนวทางในการทดสอบ โดยได้มีการนิยามศัพท์เฉพาะ ทฤษฎีที่เกี่ยวข้อง งานวิจัย และบทความที่เกี่ยวข้องกับกรอบการทำงานการทดสอบซอฟต์แวร์ กระบวนการทดสอบ

ในรูปแบบ V-Model ตลอดจนถึงข้อมูลของระบบสารสนเทศที่ทำการทดสอบ ซึ่งจะกล่าวถึงในลำดับถัดไป

2.1. กระบวนการทดสอบแบบ V-Model

กระบวนการพัฒนาซอฟต์แวร์เป็นลำดับจากบนลงล่าง และกระบวนการทดสอบซอฟต์แวร์เป็นลำดับจากล่างขึ้นบนตามลักษณะของตัวอักษร V ทุกกระบวนการสามารถทำย้อนกลับเพื่อทบทวน/ตรวจสอบที่ผ่านมาได้ (Validation and Verification) [2] ในรูปที่ 1



รูปที่ 1. กระบวนการทดสอบซอฟต์แวร์แบบ V-Model

2.2. การทดสอบซอฟต์แวร์

การจัดทำกรอบการทำงานทดสอบซอฟต์แวร์ เป็นการจัดทำขึ้นเพื่อทดสอบซอฟต์แวร์ภายในองค์กร ในการทดสอบซอฟต์แวร์จะแบ่งการทดสอบซอฟต์แวร์ออกเป็น 3 การทดสอบ ได้แก่ การทดสอบระดับบูรณาการ (Integration Test) การทดสอบระบบ (System Test) และการทดสอบใช้งานจริง (User Acceptance Test) [1]

2.3. การทดสอบระดับบูรณาการ (Integration Test)

การทดสอบการทำงานของกลุ่มโปรแกรมหรือส่วนประกอบย่อยที่ถูกประสานเข้าด้วยกัน โดยทำงานหน้าที่ใดหน้าที่หนึ่งร่วมกันสำหรับวิธีการทดสอบเพิ่มทีละหน่วยมีให้เลือก 2 วิธี ได้แก่ การทดสอบแบบเพิ่มโมดูลจากบนลงล่าง (Top - Down Approach) และการทดสอบเพิ่มโมดูลจากล่างขึ้นบน (Bottom - Up Approach)

2.4. การทดสอบระบบ (System Test)

การทดสอบร่วมกันของทั้งระบบ โดยเน้นที่การประสานเชื่อมโยงกันระหว่างระบบย่อย รวมทั้งการตรวจสอบในภาพรวมของระบบว่าระบบได้ตอบสนองความต้องการใช้งานทั้งในส่วนที่ฟังก์ชันการทำงาน และในส่วนของคุณภาพของซอฟต์แวร์โดยอ้างอิงจากเอกสารกำหนดความต้องการใช้งานซอฟต์แวร์เป็นหลัก

2.5. การทดสอบใช้งานจริง (User Acceptance Test)

การทดสอบในขั้นตอนสุดท้ายก่อนที่ระบบจะถูกยอมรับได้ว่าสามารถทำงานได้จริง เป็นการทดสอบร่วมกันระหว่างผู้ใช้งานและผู้ออกแบบพัฒนาระบบโดยใช้ข้อมูลจริงป้อนเข้าสู่ระบบ เพื่อจำลองสถานการณ์ใช้งานระบบในอนาคต ซึ่งอาจทำให้ค้นพบข้อผิดพลาดบางอย่างที่ไม่คาดคิดมาก่อน เนื่องจากการทดสอบในขั้นตอนที่ผ่านมาทั้งหมดล้วนแล้วแต่ใช้ข้อมูลจำลองทั้งสิ้น

ดังนั้นจึงอาจเป็นไปได้ว่ายังมีบางประเด็นที่ซอฟต์แวร์ไม่ได้พัฒนาเพื่อรองรับสถานการณ์ดังกล่าว

3. ระบบสารสนเทศที่เกี่ยวข้อง

ระบบสารสนเทศที่เชื่อมโยงกับการทดสอบซอฟต์แวร์ คือ ระบบจัดการการทดสอบซอฟต์แวร์ (Software Testing Management System: STMs) [3] ซึ่งเป็นระบบสำหรับการจัดการทดสอบซอฟต์แวร์มีขั้นตอนการทดสอบแบบ V-Model พัฒนาโดยนิสิตชั้นปีที่ 3 สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ซึ่งระบบสามารถช่วยในการสร้างโครงการทดสอบ ตั้งแต่ขั้นตอนการวางแผน จัดทำกรณีทดสอบของการทดสอบแต่ละระดับ จัดเก็บข้อมูลข้อบกพร่อง สถานะการจัดการข้อบกพร่องระหว่างผู้ทดสอบและนักพัฒนาซอฟต์แวร์ ตลอดจนรายงานสรุปผลการทดสอบในรูปแบบตารางและแผนภูมิ ซึ่งเป็นเครื่องมือหลักในการดำเนินโครงการครั้งนี้

โดยระบบจัดการการทดสอบซอฟต์แวร์ ประกอบด้วย 3 เมนู ได้แก่

- 1) ข้อมูลพื้นฐาน
- 2) โครงการทดสอบ
- 3) รายงานสถิติการทดสอบ

ในส่วนของเมนูการจัดการโครงการทดสอบ เป็นส่วนที่ใช้ในการสร้างโครงการทดสอบต่าง ๆ โดยหลังจากที่ได้สร้างโครงการทดสอบขึ้นมาแล้ว จะสามารถสร้าง

- 1) TEST PLAN
- 2) TEST LOG
- 3) DEFECT LOG
- 4) TEST REPORT

TEST PLAN จะเป็นส่วนที่ใช้ในการใส่รายละเอียดของแผนการทดสอบ เช่น วัตถุประสงค์การทดสอบ, ขอบเขตการทดสอบ, ความเสี่ยง และทีมทดสอบ เป็นต้น

TEST LOG จะเป็นส่วนของการเพิ่มชุดกรณีทดสอบ และแสดงรายการกรณีทดสอบในโครงการนั้น ๆ โดยสามารถทำการคัดลอกชุดข้อมูลของกรณีทดสอบได้

DEFECT LOG จะแสดงรายการ และเพิ่มข้อบกพร่องที่พบ

TEST REPORT จะแสดงรายงานสรุปผลการทดสอบ (TEST SUMMARY REPORT) ซึ่งประกอบด้วย

- 1) รายงานจำนวนกรณีทดสอบ (TEST CASE)
- 2) รายงานสรุปผล ซึ่งประกอบด้วย
 - รายงานสรุประดับความรุนแรง
 - รายงานสรุประดับความสำคัญ
 - รายงานสรุปผลการทดสอบ
 - รายงานสรุปผลประเภทข้อบกพร่อง

โดยสามารถเลือกได้ว่าจะให้แสดงเพียงอย่างใดอย่างหนึ่ง หรือแสดงทั้งหมด

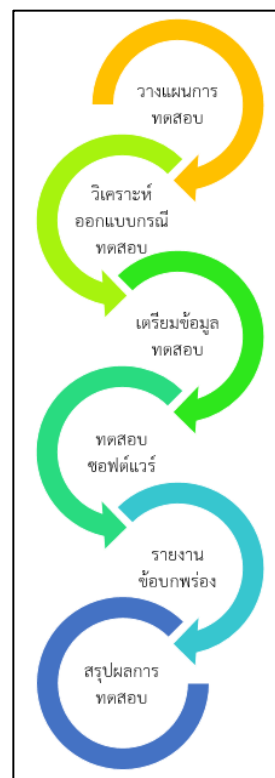
ในการดำเนินโครงการ ผู้ดำเนินโครงการได้ทำการทดสอบซอฟต์แวร์ทั้งหมด 4 โครงการ ได้แก่ โครงการให้บริการซอฟต์แวร์ผ่านอินเทอร์เน็ต วทบ.เพชรบุรี โครงการให้บริการซอฟต์แวร์ผ่านอินเทอร์เน็ต วทบ.สุพรรณบุรี โครงการให้บริการซอฟต์แวร์ผ่านอินเทอร์เน็ต วทบ.ยะลา โครงการให้บริการซอฟต์แวร์ผ่านอินเทอร์เน็ต วทบ.สุราษฎร์ธานี

ในแต่ละโครงการประกอบไปด้วย 7 ระบบ ดังต่อไปนี้

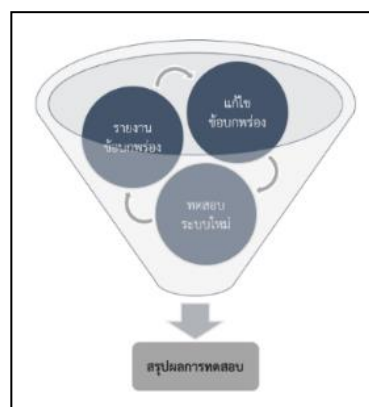
- 1) ระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณ
- 2) ระบบบุคลากรและส่งเสริมการพัฒนามูลฐาน
- 3) ระบบบริหารจัดการครุภัณฑ์
- 4) ระบบบริหารจัดการกรอบมาตรฐานคุณวุฒิระดับอุดมศึกษา
- 5) ระบบจัดการผลงานวิจัยและวิชาการ
- 6) ระบบงานบริการวิชาการ
- 7) ระบบงานทำนุบำรุงศิลปวัฒนธรรม

4. การดำเนินงาน

หลังจากการศึกษาหาความรู้เพิ่มเติมเพื่อนำมาวิเคราะห์และออกแบบกรอบการทำงานการทดสอบซอฟต์แวร์ได้ผลสรุปออกมาเป็นกรอบการทำงานดังรูปที่ 2 ซึ่งมีทั้งหมด 6 ขั้นตอน ในขั้นตอนของการรายงานข้อบกพร่องจะมีขั้นตอนย่อยลงไปอีก 2 ขั้นตอนคือ แก้ไขข้อบกพร่องและทดสอบระบบใหม่ ซึ่งการทำงานของขั้นตอนนี้จะทำซ้ำจนกว่าระบบจะไม่พบข้อบกพร่องอีกต่อไป ดังรูปที่ 3.



รูปที่ 2. กรอบการทำงานการทดสอบซอฟต์แวร์



รูปที่ 3. การทำงานของการรายงานข้อบกพร่อง

จากการที่ผู้ดำเนินโครงการได้ทำการทดสอบระบบต่าง ๆ ในแต่ละโครงการ จึงได้ยกตัวอย่างมา 1 ระบบที่ได้นำกรอบการทำงานการทดสอบซอฟต์แวร์มาใช้ นั่นคือ ระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณ ของโครงการให้บริการซอฟต์แวร์ผ่านอินเทอร์เน็ต วทบ.เพชรบุรี ซึ่งมีรายละเอียดการทดสอบดังต่อไปนี้

3.1. วางแผนการทดสอบ

ผู้ดำเนินโครงการได้กำหนดลำดับการทดสอบ ระยะเวลาที่ต้องทำการทดสอบ วันที่เริ่มต้น - วันที่สิ้นสุด ซึ่งแสดงรายละเอียดในตารางที่ 1

ตารางที่ 1 แผนการทดสอบ (Test Schedule)

	งาน	ระยะเวลา	วันที่ เริ่มต้น	วันที่ สิ้นสุด
1	วางแผนการ ทดสอบ	1 วัน	02/12/59	02/12/59
2	วิเคราะห์ ออกแบบกรณี ทดสอบ	1 สัปดาห์	05/12/59	09/12/59
3	เตรียมข้อมูล ทดสอบ	1 วัน	09/12/59	09/12/59
4	ทดสอบ ซอฟต์แวร์	1 สัปดาห์	12/12/59	16/12/59
5	รายงาน ข้อบกพร่อง	1 สัปดาห์	19/12/59	23/12/59
6	สรุปผลการ ทดสอบ	1 วัน	23/12/59	23/12/59

3.2. วิเคราะห์ออกแบบกรณีทดสอบ

การตรวจสอบความถูกต้องของโมเดลในการทำงานของระบบนั้น ต้องพยายามหาข้อผิดพลาดของตัวระบบให้ได้มากที่สุด ใช้เวลา และจำนวนของกรณีทดสอบให้น้อยที่สุด เพื่อให้ซอฟต์แวร์มีคุณภาพและสามารถใช้งานได้อย่างถูกต้อง

ซึ่งในการทดสอบของระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณนี้มีทั้งหมด 4 โมดูล ดังต่อไปนี้

- 1) โมดูลตั้งค่า
 - จัดการปีงบประมาณ
 - แสดงแหล่งงบประมาณ
 - เปลี่ยนสถานการณ์ใช้งานกลุ่มข้อมูลยุทธศาสตร์
 - แสดงโครงสร้างองค์กร
- 2) โมดูลข้อมูลยุทธศาสตร์
 - จัดการข้อมูลพันธกิจ
 - จัดการข้อมูลประเด็นยุทธศาสตร์
 - จัดการข้อมูลเป้าประสงค์
 - จัดการข้อมูลกลยุทธ์
- 3) โมดูลโครงการตามแผนยุทธศาสตร์
 - จัดการข้อมูลโครงการ
 - แสดงสรุปโครงการตามยุทธศาสตร์
- 4) โมดูลจัดการงบประมาณโครงการ
 - จัดการงบประมาณจัดสรร
 - จัดการงบประมาณใช้ไป

3.3. เตรียมข้อมูลทดสอบ

ก่อนการทดสอบซอฟต์แวร์ในส่วนต่าง ๆ นั้นต้องทำการเตรียมชุดข้อมูลที่ต้องใช้ทดสอบให้สอดคล้องกับกรณีทดสอบ ซึ่งตัวอย่างข้อมูลระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณ เช่น ข้อมูลพันธกิจ ข้อมูลโครงการ เป็นต้น

3.4. ทดสอบซอฟต์แวร์

หลังจากที่ผู้ดำเนินโครงการได้วางแผนการทดสอบ วิเคราะห์ออกแบบกรณีทดสอบ และเตรียมข้อมูลทดสอบเรียบร้อยแล้ว จึงได้ทำการทดสอบระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณต่อไป ในส่วนของขั้นตอนที่ 5 และ 6 ในกรอบการทำงานการทดสอบซอฟต์แวร์จะถูกกล่าวถึงในผลการดำเนินงานถัดไป

5. ผลการดำเนินโครงการ

หลังจากที่ทดสอบระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณเสร็จสิ้นแล้ว จะได้ผลของทดสอบออกมา ผู้ดำเนินโครงการจึงได้ทำตามขั้นตอนที่ 5 และ 6 ของกรอบการทำงานการทดสอบซอฟต์แวร์นั้นคือ รายงานข้อบกพร่องและสรุปผลการทดสอบ ผลการทดสอบในส่วนต่าง ๆ ของระบบมีรายละเอียดดังนี้

รายงานข้อบกพร่อง เป็นการแสดงรายการข้อบกพร่องที่พบในการทดสอบ ซึ่งระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณมีรายการข้อบกพร่องทั้งหมด 1 รายการ

สรุปผลการทดสอบเป็นการสรุปผลจำนวนที่พบข้อบกพร่องจากกรณีทดสอบทั้งหมด ระดับความรุนแรง ระดับความสำคัญทั้งหมด แสดงสรุปผลการทดสอบ)Test Summary Report (ของการทดสอบการทำงานของระบบแผนงานโครงการและติดตามการเบิกจ่ายงบประมาณ จากกรณีทดสอบทั้งหมด 27 กรณีทดสอบ พบว่า ผ่านการทดสอบ 20 กรณีทดสอบ ไม่ผ่านการทดสอบ 2 กรณีทดสอบ และถูกระงับการทดสอบ 5 กรณีทดสอบ ดังในรูปที่ 4. คิดเป็นเปอร์เซ็นต์ ผ่าน 74.07% ไม่ผ่าน 7.41% และถูกระงับการทดสอบ 18.52% ดังในรูปที่ 5.

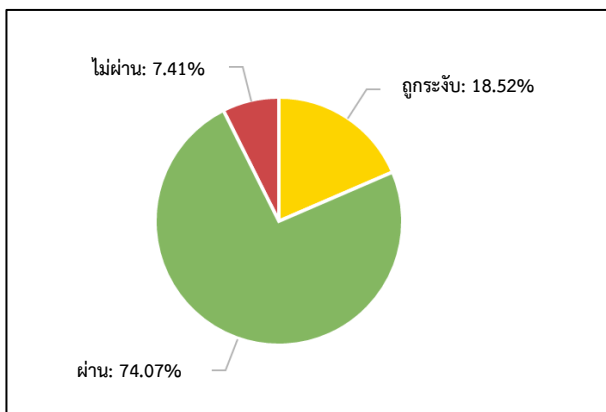
6. สรุปผล

บทความนี้เป็นนำเสนอกรอบการทำงานการทดสอบซอฟต์แวร์สำหรับห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ โดยโครงการที่ทำการทดสอบมีทั้งหมด 4 โครงการ เพื่อให้การทดสอบภายในห้องปฏิบัติการฯ มีแนวทางการทดสอบไปในทิศทางเดียวกัน และเพิ่มประสิทธิภาพการทำงานในองค์กรมากยิ่งขึ้น

ตารางจำนวนกรณีทดสอบ		
สถานะ	จำนวน	ร้อยละ
ทั้งหมด	27	
กำลังทดสอบ	0	0.00
ผ่าน	20	74.07
ไม่ผ่าน	2	7.41
ยังไม่ได้ทดสอบ	0	0.00
ระงับการทดสอบ	5	18.52

เกณฑ์	เป้าหมาย	ผลลัพธ์	สถานะ
RUN RATE	90.00	81.48	ไม่ผ่าน
PASS RATE	90.00	74.07	ไม่ผ่าน

รูปที่ 4. ตารางสรุปผลจำนวนกรณีทดสอบ



รูปที่ 5. แผนภูมิสรุปผลการทดสอบ

ในอนาคตได้นำระบบการจัดการทดสอบซอฟต์แวร์ มาช่วยในการจัดการ ปรับปรุง ติดตามกระบวนการทดสอบ และทำการสรุปผลการทดสอบผ่านระบบสารสนเทศ เพื่อให้ผู้ที่มีส่วนเกี่ยวข้องสามารถดูผลการทดสอบได้อย่างมีประสิทธิภาพมากยิ่งขึ้น รวมถึงติดตามสถานะข้อบกพร่องอย่างสม่ำเสมอ เพื่ออำนวยความสะดวกแก่ผู้ทดสอบ และนักพัฒนาซอฟต์แวร์

เอกสารอ้างอิง

- [1] Wikipedia (2006). Software Testing [ออนไลน์]. เข้าถึงได้จาก : https://en.wikipedia.org/wiki/Software_testing
- [2] Testnbug.com (2014). Software development life cycle Models [ออนไลน์]. เข้าถึงได้จาก : <http://www.testnbug.com/2014/12/software-development-life-cycle-models/>
- [3] TSP57 (2017). ระบบจัดการการทดสอบซอฟต์แวร์ [ออนไลน์]. เข้าถึงได้จาก: <https://10.51.4.17/TSP57/ISERL/>