การใช้ระบบอัตโนมัติทดสอบซอฟต์แวร์เว็บแอพพลิเคชั่น : กรณีศึกษาบริษัทด้านเทคโนโลยีสารสนเทศและการสื่อสาร

(Using Automation Testing Test the Software Web Application : Case Study of Information and Communication Technology Company)

เศรษฐพงษ์ อิ่มสุวรรณ์ *

ผู้ช่วยศาสตราจารย์ ดร.ศุภรัชชัย วรรัตน์

บทคัดย่อ

ปัจจุบันนี้องค์กรธุรกิจที่เกี่ยวกับการผลิตหรือการให้การบริการได้พัฒนาซอฟต์แวร์ขึ้นมาใช้ในการจัดการการปฏิบัติการ ขั้นตอนที่สำคัญของการนำซอฟต์แวร์มาใช้งานคือการตรวจสอบการทำงานของซอฟต์แวร์ให้ถูกต้องตามข้อกำหนดของผู้ใช้งาน ผู้วิจัยจึงทำการศึกษาเรื่องการตรวจสอบการทำงานของซอฟต์แวร์มีวัตถุประสงค์เพื่อลดเวลาของการตรวจสอบ และลดข้อผิดพลาดของการใช้งาน ผู้วิจัยดำเนินการใช้การทดสอบอัตโนมัติเข้ามาทดสอบซอฟต์แวร์เปรียบเทียบกับการทดสอบด้วยมือ เพื่อศึกษาและพัฒนาระบบการทดสอบอัตโนมัติมาประยุกต์ใช้ในการทดสอบซอฟต์แวร์เว็บแอพพลิเคชั่นทำให้ลดระยะเวลาในการทดสอบ ผลที่ตามมาคือ สามารถลดแรงงานคน ค่าใช้จ่าย เวลา ความผิดพลาดในการบันทึกข้อมูล ผู้วิจัยได้ทำการศึกษาโดยใช้ผู้ทดสอบ ทั้งหมด 9 คนภายในทีม ทำการทดสอบซอฟต์แวร์ด้วยการทดสอบด้วยมือ 20 กรณี จำนวน 5 ครั้ง แล้วหาค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบ และบันทึกเวลาที่ใช้ในการทดสอบแบบอัตโนมัติ 20 กรณีเดียวกันจำนวน 5 ครั้ง พบว่าเวลาเฉลี่ยที่ใช้ในการทดสอบแบบอัตโนมัติมีเวลาน้อยกว่าเวลาที่ใช้ในการทดสอบด้วยมือ ทุกครั้ง ค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบแบบอัตโนมัติคิดเป็น ร้อยละ 19.26 ของเวลาที่ใช้ในการทดสอบด้วยมือ ผลการศึกษาเป็นไปตามวัตถุประสงค์ของการนำการทดสอบแบบอัตโนมัติ ทำให้ลดระยะเวลาในการทดสอบ ซอฟต์แวร์เว็บแอพพลิเคชั่น และยังได้ผลการทดสอบที่ การทดสอบด้วยมือไม่สามารถทำได้ได้แก่ สามารถสั่งงานแบบตั้งเวลาได้ รายงานผลการทดสอบได้รวดเร็วและใช้เป็นเครื่องมือติดตามกรณีทดสอบว่าถูกต้องครบถ้วนตามข้อกำหนด

คำสำคัญ: การทดสอบซอฟต์แวร์ การทดสอบซอฟต์แวร์อัตโนมัติ

---

* นักศึกษาหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาการจัดการทางวิศวกรรม มหาวิทยาลัยธุรกิจบัณฑิตย์

** ที่ปรึกษาสารนิพนธ์

ABSTRACT

The objective of this research is to study the using Automation Testing test the operation of the software according to user requirements. It is an important step and is required to investigate the accomplishment of software. Researcher use Automation Testing to examine software web application compare to Manual Testing. Using Automation Testing to prove that time spent in testing software web application can reduce, with more advantages such as human labor costs, data recording errors. The study was conducted by using nine manual testers test twenty cases compare to automation testing The manual testing process takes a significant amount of time as compared to automation testing. In the experiment, automation testing took 19.26 percent of the time allocated for manual testing on the average. The results of the study were based on the purpose of Automation Testing, which reduced the testing time. The more test results are not available in the Manual Testing, such as : can be run on schedule run and report test results quickly. Test result Log is a test case that is accurate to meet the requirements.

1. บทนำ

การนำคอมพิวเตอร์มาใช้งานมีพัฒนาการมาตามลำดับ ยุคแรกของการเริ่มใช้ให้คอมพิวเตอร์ทำงานใช้การเขียนโปรแกรมเป็นโค้ด และนำไปทดลองใช้เพื่อตรวจสอบว่าโปรแกรมที่เขียนขึ้นทำงานได้ ต่อมาความต้องการของผู้ใช้งานคอมพิวเตอร์มากขึ้น และงานซับซ้อนมากขึ้น การเขียนโปรแกรมพัฒนาเป็นซอฟแวร์ ซอฟต์แวร์ได้ถูกพัฒนาขึ้นเพื่อใช้งานในเรื่องต่าง ๆ เช่น งานในสำนักงาน หรืองานเฉพาะเรื่องของธุรกิจต่าง ๆ การพัฒนาซอฟต์แวร์มีขั้นตอนที่สำคัญ คือ การทดสอบซอฟต์แวร์ที่พัฒนาขึ้นก่อนนำไปใช้งาน การทดสอบซอฟต์แวร์ มีพัฒนาการตามลำดับ การทดสอบเริ่มจากทดสอบเพื่อให้แน่ใจว่าซอฟต์แวร์ทำงานได้ตามข้อกำหนดทางเทคนิค ใช้สั่งงานให้คอมพิวเตอร์ทำงานได้ตามความต้องการของผู้ใช้งาน กรณีที่มีข้อผิดพลาดผู้เขียนโปรแกรมจะทดสอบเพื่อค้นหาข้อบกพร่องภายในโปรแกรมแล้วแก้ไขการทำงานของโปรแกรมให้สั่งงานได้ถูกต้อง การทดสอบซอฟต์แวร์จะช่วยค้นหาข้อบกพร่อง และลดข้อผิดพลาดจากการทำงานของซอฟต์แวร์ให้เหลือน้อยที่สุด เป็นสิ่งที่ช่วยเพิ่มคุณภาพให้กับซอฟต์แวร์

ในช่วงแรก ๆ ของการพัฒนาซอฟต์แวร์ที่เขียนขึ้นเพื่อใช้งานนั้น สามารถใช้การทดสอบแบบ Manual Testing เพราะซอฟต์แวร์มีองค์ประกอบน้อย และมีการทำงานที่ไม่ซับซ้อนมากนัก การทำงานทุกงานต้องมีการพัฒนาระบบงานและมีการทดสอบอย่างต่อเนื่อง การพัฒนาระบบงานทำให้องค์ประกอบเพิ่มขึ้นเมื่อซอฟต์แวร์นั้นได้รับการพัฒนาหรือเพิ่มฟังก์ชั่นการทำงานเข้ามาในระบบต้องทดสอบว่าการพัฒนาระบบกระทบกับระบบการทำงานที่ทำอยู่หรือไม่ เมื่อมีการพัฒนาหรือเปลี่ยนแปลงเพื่อยืนยันว่าฟีเจอร์ที่เพิ่มขึ้นมานั้นไม่กระทบกับระบบการทำงานเดิมที่ได้ทำการทดสอบไปแล้ว ทำให้มีงานบางส่วนนั้นต้องทำซ้ำ ๆ กับส่วนที่ได้ทดสอบไปแล้ว ผู้วิจัยจึงเห็นควรนำการทดสอบแบบอัตโนมัติเข้ามาใช้ทดสอบเมื่อมีการปรับปรุง

กระบวนการพัฒนาซอฟต์แวร์ เพื่อช่วยลดระยะเวลาการทดสอบแบบการทดสอบด้วยมือ และแรงงานที่ใช้ในการทดสอบ รวมถึงแก้ปัญหากระบวนการทดสอบความถดถอย

2. วัตถุประสงค์ของการศึกษาวิจัย

1) เพื่อลดระยะเวลาในการทดสอบซอฟต์แวร์เว็บแอพพลิเคชั่นด้วยการทดสอบแบบอัตโนมัติ

3. ขอบเขตของงานวิจัย

1) พัฒนาเครื่องมือสำหรับสร้างกรณีทดสอบแบบ Automation Testing

2) ใช้การทดสอบอัตโนมัติทดสอบการทำงานของซอฟต์แวร์เว็บแอพพลิเคชั่นของบริษัทที่เป็นกรณีศึกษา

3) ทำการทดสอบความถดถอย

4. ประโยชน์ที่จะได้รับ

การใช้การทดสอบอัตโนมัติทดสอบการทำงานของซอฟต์แวร์เว็บแอพพลิเคชั่น

1) ลดปัญหาการใช้แรงงานคนทำงานที่ต้องป้อนข้อมูลที่ซ้ำกันเมื่อทำการทดสอบด้วยมือ

2) สามารถทำงานแทนคนได้รวดเร็วและมีความแม่นยำมากกว่าคน

3) ช่วยในการทำการทดสอบความถดถอย

5. ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

5.1 ทฤษฎี

5.1.1 Lean Software Development (LSD)

Lean Software Development (LSD) เกิดและพัฒนามาจากโรงงานอุตสาหกรรม มุ่งเน้นการกำจัดความสิ้นเปลืองในการผลิต เพื่อให้การผลิตมีประสิทธิภาพสูงสุด การใช้ทรัพยากรเพื่อให้เกินประโยชน์สูงสุด ซึ่งต่อมาได้มีบทบาทในการนำ Lean มาใช้ใน การพัฒนา Software Lean มีแนวคิดอยู่ 7 ประการ และมี tools ในการปฏิบัติทั้งหมด 22 อย่าง

แนวคิดอยู่ 7 ประการ

1. Eliminate waste กำจัดความสิ้นเปลือง
2. Amplify Learning หลักการเบื้องต้นที่นำพาสู่การ เรียนรู้ความต้องการลูกค้า
3. Decide ad Late as Possible เป็นการคงตัวเลือก คงการตัดสินใจที่เด็ดขาดไว้
4. Deliver as fast as possible ส่งมอบงานให้ลูกค้าอย่างรวดเร็ว
5. Empower the Team องค์กรที่เติบโตแล้ว

6. Build Integrity In สร้างความสมบูรณ์ของซอฟต์แวร์

7. See the whole ถ้าเป็นระบบที่ซับซ้อนให้เริ่มจากการแบ่งส่วนของระบบให้เป็นระบบย่อย ๆ แล้วจัดการทีละส่วน

การพัฒนาซอฟต์แวร์ขึ้นใช้งานทุกงาน มีความมุ่งหมายให้ใช้งานได้ถูกต้องและตรงตามข้อกำหนดของผู้ใช้งาน ผู้วิจัยได้ประยุกต์ใช้แนวคิดของ Lean Software Development ตามกฎข้อที่ 1 มุ่งเน้นไปที่ Eliminate waste การกำจัดความสิ้นเปลืองในการพัฒนาซอฟต์แวร์ ความสิ้นเปลืองของงานพัฒนาซอฟต์แวร์ ได้แก่ การผลิตซอฟต์แวร์ ที่ไม่มีคุณภาพ เป็นการผลิตของเสียที่ต้องการแก้ไขอยู่บ่อยครั้งทำให้สิ้นเปลือง เวลา แรงงาน และค่าใช้จ่าย เพื่อลดความสูญเสีย เวลา แรงงาน ค่าใช้ จ่าย ทำได้โดยค้นหาข้อบกพร่องของซอฟต์แวร์ก่อนนำไปใช้งานด้วยโปรแกรมทดสอบแบบอัตโนมัติ เพื่อลดความสูญเสียเวลาในการติดตามตรวจสอบและแก้ไขข้อบกพร่องแต่ละข้ออย่างรวดเร็วเป็นการสร้างความสมบูรณ์ ให้กับซอฟต์แวร์ก่อนส่งมอบงานให้ผู้ใช้

5.1.2 The 7 Wastes

ขั้นตอนพื้นฐานในการพัฒนาซอฟต์แวร์นั้น ประกอบไปด้วยการวางแผน เขียนโค้ด และการทดสอบซอฟต์แวร์ โดย 7 Wastes จะมุ่งเน้นไปที่ความสิ้นเปลือง 7 ประการในการพัฒนาซอฟต์แวร์ประกอบไปด้วย

1. Over Production ขยะจากการผลิตมากเกินความจำเป็น
2. Over Processing ขยะจากการมีขั้นตอนการทำงานมากเกินไป
3. Transportation ขยะจากการเคลื่อนย้ายงาน
4. Inventory ขยะจากการมีของคงคลังมากเกินไป
5. Motion ขยะจากการเคลื่อนไหวของคนทำงาน
6. Waiting ขยะจากการรอคอย
7. Defects ขยะจากผลิตภัณฑ์ที่บกพร่อง

การกำจัดความสิ้นเปลือง ผู้วิจัยได้เลือกขั้นตอนที่ 7 ในการพัฒนาซอฟต์แวร์คือ Defects ข้อบกพร่องที่เกิดขึ้นขณะที่มีการพัฒนาซอฟต์แวร์เป็นสิ่งที่หลีกเลี่ยงไม่ได้แต่จำนวนข้อบกพร่องสามารถจัดการได้จากโดยใช้วิธีการที่เหมาะสม การกำจัดข้อบกพร่องทีมพัฒนาซอฟต์แวร์จะเน้นไปที่การป้องกันการเกิดขึ้นของข้อบกพร่องตั้งแต่ขั้นตอนแรก ๆของการพัฒนา เพื่อให้สามารถค้นหาข้อบกพร่องตั้งแต่ขั้นตอนแรก ๆของการพัฒนา การค้นพบข้อผิดบกพร่องได้เร็วจะลดผลกระทบที่อาจจะเกิดขึ้นในขั้นตอนการพัฒนาระบบในลำดับต่อไป ข้อมูลที่เกี่ยวข้องกับการวิเคราะห์ข้อบกพร่อง หรือรายงานผลการแก้ไขข้อบกพร่อง ควรเป็นการทำงานแบบอัตโนมัติและสื่อสารกันระหว่างทีมพัฒนา ทีมผู้ทดสอบ และผู้ใช้งาน

5.2 งานวิจัยที่เกี่ยวข้อง

รัชนก ไชยประเสริฐ (2013) ได้ทำการวิจัย เรื่องกรอบการทดสอบระบบอัตโนมัติในระบบ POCT (Point of Care Testing) งานวิจัยนี้เสนอการทดสอบระบบแบบอัตโนมัติในกระบวนการพัฒนาซอฟต์แวร์ ซึ่ง

ช่วยเพิ่มประสิทธิภาพในการทดสอบซอฟต์แวร์และช่วยลดต้นทุนในการพัฒนาซอฟต์แวร์ด้านทรัพยากร คน เวลา และสามารถรายงานผลการทดสอบได้อย่างรวดเร็ว

ธนพล สิชฌนุกฤษฎ์ (2554) ได้ทำการวิจัยเรื่อง การพัฒนาระบบจัดการกรณีทดสอบซอฟต์แวร์ งานวิจัยเรื่องนี้ได้พัฒนาเครื่องมือระบบจัดการกรณีทดสอบซอฟต์แวร์ เพื่อนำมาช่วย ผู้ทดสอบในการสร้างกรณีทดสอบ นอกจากนั้นยังช่วยในการเก็บข้อมูล กรณีทดสอบ การตรวจสอบ การติดตาม การแก้ไข เพิ่ม-ลด กรณีทดสอบที่ผู้ทดสอบใช้ในการทดสอบทั้งหมด เพื่อให้การทดสอบทำได้สะดวกยิ่งขึ้นและเป็นการลดภาระของผู้ทดสอบ

สกรณ์ บุษบง (2556) ได้ทำการวิจัยเรื่องการสร้างกรณีทดสอบ สำหรับการทดสอบระดับรวมหน่วยเพิ่มทีละหน่วยโดยอัตโนมัติ จากกรณีทดสอบระดับหน่วย งานวิจัยนี้ได้เสนอแนวทางแก้ไขปัญหาความล่าช้า และยุ่งยากของการสร้างกรณีทดสอบระดับรวมหน่วย โดยพัฒนาเครื่องมือที่ใช้ในการสร้างกรณีทดสอบระดับรวมหน่วยอัตโนมัติจากกรณีทดสอบระดับหน่วยโดยใช้วิธีเพิ่มทีละหน่วยจากกรณีทดสอบระดับหน่วย งานวิจัยเรื่องนี้อธิบายการสร้างความสัมพันธ์ระหว่างโมดูลเพื่อลดความยุ่งยากซับซ้อน และเวลาในการสร้างกรณีทดสอบ

ศิวิมล เย็นไสว (2558) ได้ทำการวิจัยเรื่องแนวทางแก้ไข ปัญหาการทดสอบซอฟต์แวร์ด้วยวิธี ATDD และการบริหารผลการปฏิบัติงาน งานวิจัยเรื่องนี้มีวัตถุประสงค์เพื่อศึกษาปัญหาที่ส่งผลต่อประสิทธิภาพการทดสอบ ซอฟต์แวร์และนำเสนอแนวทางการแก้ปัญหาที่คาดว่าจะส่งผลต่อประสิทธิภาพของซอฟต์แวร์ แนวทางแก้ไขโดยเลือกแบบ ATDD Acceptance Test Driven Development มาใช้ร่วมกับการบริหารผลการปฏิบัติงาน โดยที่ขั้นตอนการทำงานของ ATDD ช่วยในด้านของการออกแบบก่อนเริ่มการทดสอบซอฟต์แวร์ พบว่า การนำเครื่องมืออัตโนมัติมาใช้แก้ปัญหาด้านเครื่องมือช่วยลดเวลาในการทดสอบซอฟต์แวร์ ช่วยค้นพบจุดบกพร่องในการทดสอบทำให้แก้ไขได้รวดเร็ว

ณัฐรัตน์ หาญวรวงศ์ (2556) ได้ทำการวิจัยเรื่อง การออกแบบและพัฒนาการสร้างกรณีทดสอบสำหรับการทดสอบซอฟต์แวร์แบบอัตโนมัติ โดยใช้โครงสร้าง UI user interface ผู้วิจัยได้ศึกษาเรื่อง กรอบการทำงานที่ช่วยให้การพัฒนาซอฟต์แวร์รวดเร็วและสามารถวิเคราะห์กรณีทดสอบ เพื่อหาส่วนของซอฟต์แวร์ที่ควรได้รับการทดสอบ ผู้วิจัยจึงพัฒนาเครื่องมือสร้างโครงสร้าง UI ของหน้าจอของซอฟต์แวร์ที่ต้องการทดสอบ เครื่องมือที่พัฒนาจากโครงสร้างของ UI สามารถสร้างกรณีทดสอบสำหรับการทดสอบแบบอัตโนมัติได้จากข้อมูลของโครงสร้าง UI

Juraj Huska (2012) ได้ทำวิจัยเรื่อง Automated Testing of the Component-based Web Application User Interface เพื่อศึกษาแนวโน้ม การทดสอบ Web Application ขององค์กรธุรกิจ ผู้วิจัยสำรวจเครื่องมือของ Automated Testing เพื่อเลือกใช้กับ Web Application และเปรียบเทียบการใช้เครื่องมือในประเด็นการใช้งาน ซึ่งจะทำให้การทดสอบได้ผลสำเร็จ ผลการวิจัยพบว่าการประยุกต์ใช้ Application Programming Interface (API) สำหรับใช้ทดสอบองค์ประกอบของ Web Application ทำให้ผลการทดสอบเป็นผลสำเร็จ

Phat Chau Tan (2016) ได้ทำงานวิจัยเรื่อง Automation Testing With Robot Framework ผลงานวิจัยพบว่า Automation Testing ช่วยลดเวลาที่ใช้ในการทดสอบได้มาก ในขณะเดียวกันก็ทำ Regression Testing ให้ได้เพราะ Automation Test สามารถ Run Test ในเวลากลางคืนก็ได้ วันหยุดสุดสัปดาห์ก็ได้ ผลการทำวิจัยเรื่องนี้ Test Case ทุกกรณี run ได้ผลเสร็จสมบูรณ์ และประสบความสำเร็จ ถึงแม้ว่าจะมีปัญหาให้แก้ไขหรือมีสิ่งที่เข้าใจยาก และสลับซับซ้อนในบางขั้นตอน ทำให้ผลการทดสอบน่าเชื่อถือ

Alazar Seyoum Haile (2011) ทำงานวิจัยเรื่อง Automation of Test Cases for Web Application of CRM Test Cases (Customer Relationship Management) ผู้ดำเนินการวิจัยนำกรณีทดสอบ 10 Test Cases ผลการทดสอบด้วยแบบ Automation Testing ใช้เวลาในการทดสอบเปรียบเทียบกับเวลาที่ทำการด้วยแบบ Manual Testing ใน 10 Test Cases เดียวกัน กรณีต่อกรณี กระบวนการทดสอบด้วยแบบ Manual Testing ใช้เวลามากกว่าการทดสอบแบบ Automation Testing ทุกกรณี เมื่อเปรียบเทียบกันโดยเฉลี่ยของการทดสอบ 5 กรณี เวลาที่ใช้ในการทดสอบแบบ Automation Testing ใช้เวลาเพียง 16% โดยเฉลี่ยของเวลาที่ใช้ทดสอบแบบ Manual Testing ผลงานวิจัยที่ได้อีกอย่าง คือ การใช้ Automation ทดสอบ Web ของ Customer Service ลดจำนวน Error ที่อาจจะเกิดจากการป้อนข้อมูลที่มีความสลับซับซ้อน เมื่อต้องป้อนข้อมูลซ้ำ ๆ ในการทำการทดสอบถดถอย

6. วิธีการดำเนินงานวิจัย

1 ศึกษาการทดสอบซอฟต์แวร์ของบริษัทที่เป็นกรณีศึกษา

2 เครื่องมือที่ใช้ในการวิจัย

3 วิเคราะห์ปัญหาการทดสอบซอฟต์แวร์แบบ Manual Testing และเหตุผลที่ใช้แบบ Automation Testing

4 พัฒนา Framework สำหรับทำ Automation Testing

5 เปรียบเทียบเวลาทดสอบซอฟต์แวร์ด้วยแบบ Manual Testing และ แบบ Automation Testing วิเคราะห์ผลการทดสอบ

6 ทำ Regression Testing

6.1 อุปกรณ์และเครื่องมือที่ใช้ในการวิจัย

6.1.1 อุปกรณ์ฮาร์ดแวร์ที่จะนำมาใช้

เครื่องคอมพิวเตอร์โน๊ตบุ๊ค

- หน่วยประมวลผล : Intel Core i3
- ความเร็ว : 2.5 GHZ
- หน่วยความจำ (RAM) : 8GB
- ฮาร์ดดิสก์ความจุ : 500 GB

6.1.2 ซอฟแวร์ที่จะนำมาใช้

- ระบบปฏิบัติการ Microsoft Windows 10 Home 64 bit
- Robot Framework RIDE โปรแกรมสำหรับเขียน Test Scrip และ Keyword
- Firefox เว็บบราวเซอร์สำหรับรัน Robot Framework
- Visual Studio Code โปรแกรมสำหรับแก้ไข library Python
- Selenium เป็นซอฟต์แวร์ สำหรับ Automation Testing เพื่อทดสอบเว็บแอพพลิเคชั่น
- Robot Framework ซอฟต์แวร์ สำหรับการทำ Automation Testing โดยมีรูปแบบ Syntax เป็นภาษาเขียนทำความเข้าใจได้ง่าย
- Python เป็นภาษาพื้นฐานที่ใช้สำหรับเขียน Robot Framework

6.1.3 Robot Framework Library

- Selenium 21. Library ใช้สำหรับเชื่อมต่อกับเว็บแอพพลิเคชั่น
- Builtin ใช้สำหรับจัดการส่วนที่เป็นคำหลักทั่วไปที่ต้องการใช้เกี่ยวกับ Python
- Date Time ใช้สำหรับแปลงเวลาและวันที่
- Auto It Library ใช้สำหรับจัดการส่วนที่เกี่ยวข้องกับ Window Application
- Custom Excel XIS Library ใช้สำหรับอ่านเขียนและแก้ไขข้อมูลบน Excel
- Archive Library ใช้สำหรับจัดการ File ที่เป็น zip และ bar
- HttpLibrary.HTTP สำหรับทดสอบ HTTP
- XML ใช้สำหรับสร้างและแก้ไข XML ไฟล์
- Database Library ใช้สำหรับเชื่อมต่อ Database
- Ca Plections ใช้สำหรับสร้างข้อมูลชนิด Python lists and dictionaries

6.2 ดำเนินการทดสอบซอฟต์แวร์และเก็บเวลาทดสอบ

เปรียบเทียบเวลาทำการทดสอบแบบ Manual Testing และ Automation Testing

การทดสอบซอฟต์แวร์ของบริษัทที่เป็นกรณีศึกษา เป็นการทดสอบแบบ Functional Testing การทดสอบแบบนี้เป็นขั้นตอนที่ถูกสร้างขึ้นมา เพื่อใช้ยืนยันว่าแต่ละส่วนประกอบของระบบทำงานร่วมกันได้อย่างถูกต้อง ตามความต้องการที่ตกลงร่วมกันไว้ระหว่าง Developer และ Customer โดยเน้นที่การทดสอบจาก interface ของระบบงาน ถ้าเป็นระบบเว็บแอพพลิเคชั่น ก็เน้นไปที่ user interface บน browser

ขั้นตอนแรกของการทำการวิจัย ผู้วิจัยได้ทดสอบซอฟต์แวร์ของบริษัทที่เป็นกรณีศึกษา ใช้การทดสอบแบบ Functional Testing ตาม technical specification จำนวน 20 test cases

ขั้นตอนที่ 2 เลือกผู้ทดสอบซอฟต์แวร์ในทีมงาน 9 คน ทำการทดสอบ 20 test cases ด้วย Manual Testing ทุกคน บันทึกเวลาที่ใช้ในการทดสอบแบบ Manual Testing เวลาที่ทำการทดสอบ คิดเป็นนาที

(min.) และวินาที (sec.) ทำตารางเปรียบเทียบเวลา ของผู้ทดสอบทั้ง 9 คน บันทึกเวลาของแต่ละคน คนละ 5 ครั้ง แล้วหาค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบ วิเคราะห์เวลาที่ใช้ในการทดสอบแบบ manual testing จากค่าเฉลี่ย

ขั้นตอนที่ 3 ดำเนินการทดสอบซอฟต์แวร์ 20 test case เดียวกันแบบ Automation Testing จำนวน 5 ครั้ง บันทึกเวลาที่ใช้ในการทดสอบทุกครั้ง หาค่าเฉลี่ย เวลาที่ได้จากการทดสอบแบบ Automation Testing ทำตารางเปรียบเทียบ ค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบทั้ง 2 แบบ วิเคราะห์ค่าเฉลี่ยของเวลาทุกตาราง และสรุปผลการศึกษา

7. ผลการศึกษา

7.1 ตารางเปรียบเทียบค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบวิเคราะห์จากตารางทุกตารางทั้ง 2 แบบ

ครั้งที่	กรณีทดสอบ	ค่าเฉลี่ยของเวลาที่ใช้ทดสอบแบบอัตโนมัติ ชม. : นาที. : วินาที.	ค่าเฉลี่ยของเวลาที่ใช้การทดสอบด้วยมือ ชม. : นาที. : วินาที.	ผลต่างของเวลาที่ใช้ในการทดสอบ
1	กรณีทดสอบ 20 กรณี	0 : 10 : 53	1 : 01 : 25	0:50:32
2	กรณีทดสอบ 20 กรณี	0 : 10 : 46	1 : 01 : 56	0:51:10
3	กรณีทดสอบ 20 กรณี	0 : 12 : 41	1 : 01 : 01	0:48:20
4	กรณีทดสอบ 20 กรณี	0 : 12 : 50	1 : 00 : 35	0:47:45
5	กรณีทดสอบ 20 กรณี	0 : 11 : 28	0 : 59 : 20	0:47:57

ค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบซอฟต์แวร์แอพพลิเคชั่นแบบการทดสอบด้วยมือมากกว่าเวลาในการทดสอบแบบอัตโนมัติทุกครั้งตามที่แสดงในช่องผลต่างของเวลาที่ใช้ในการทดสอบ ค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบแบบอัตโนมัติคิดเป็นร้อยละ 19.26 ของเวลาที่ใช้ในการทดสอบแบบด้วยมือเป็นไปตามวัตถุประสงค์ การใช้การทดสอบอัตโนมัติทำให้ลดเวลาที่ใช้ในการทดสอบซอฟต์แวร์แอพพลิเคชั่น

7.2 จากผลของการทดสอบด้วยวิธี Automation Testing

ผู้วิจัย พบว่า เป็นเครื่องมือที่ช่วยในการทดสอบระบบ โดยมีความสามารถต่าง ๆ ที่การทดสอบแบบ Manual Testing ทำไม่ได้ ดังนี้

1. ใช้ในการเก็บข้อมูลความต้องการของระบบ (Requirement)
2. เก็บวิธีการทดสอบ
3. เก็บเงื่อนไขของการทดสอบ (Test case)

4. เก็บข้อมูลการวางแผนกิจกรรม การทดสอบ (Test Plan and Activity Test Plan)

5. ทดแทนการรัน Test Script ด้วยคน สามารถตั้งเวลา Execute หรือสั่งการทำงานแบบ Schedule Run ได้

6. รายงานผลการทดสอบ สถานะ หรือคุณภาพของระบบที่ถูกทดสอบ (Test Result Log)

7. ใช้เป็นเครื่องมือในการติดตามว่า Test Case ครบถ้วนกับข้อกำหนดหรือไม่ โดยตรวจสอบจาก Test Coverage Matrix

8. การทำ Defect Management เพื่อใช้ติดตามข้อบกพร่อง และใช้สื่อสารกันระหว่าง Tester Team และ Developer Team สามารถส่ง Defect ในรูปแบบ Excel ที่เกิดขึ้นไปให้ Programmer ทำให้ลดความเข้าใจคลาดเคลื่อน และทำงานได้สะดวกรวดเร็ว

9. ช่วยในการทำ Regression Testing ได้เร็วกว่าทำด้วยแบบ Manual

7.3 ผลการทดสอบความถดถอยบรรลุวัตถุประสงค์ทุกข้อ

1. ผลการทดสอบความถดถอยพบว่าข้อบกพร่องที่พบได้ถูกแก้ไขแล้วทำให้ซอฟต์แวร์ทำงานได้ตามที่ระบุไว้ในข้อกำหนด

2. ไม่มีข้อผิดพลาดในการทำงานของซอฟต์แวร์หลังการแก้ไขเพิ่มเติมโค้ดภายในระบบหรือเปลี่ยนแปลงระบบปฏิบัติการ

3. รักษาคุณภาพของซอฟต์แวร์คือคุณสมบัติและฟังชั่นการทำงานของซอฟแวร์ที่ทำงานได้ตรงตามความต้องการของผู้รับบริการและตามข้อกำหนดที่ระบุไว้ในซอฟต์แวร์

7.4 ผลจากการที่ผู้วิจัยได้ พัฒนาโปรแกรมการใช้งานของ Automation Testing Tool ระหว่างการทดสอบพบว่าบรรลุวัตถุประสงค์ของการใช้งานทดสอบซอฟต์แวร์และมีข้อดีดังนี้

1. เพิ่มความรวดเร็วในการพัฒนาระบบงาน ทดสอบ

2. ทำให้เกิดความร่วมมือที่ดีระหว่าง Tester Team และ Developer Team จากการทำงานร่วมกันระหว่างที่ ดำเนินทดสอบ

3. ช่วยในการบริหารจัดการ Test Resource และการเชื่อมความสัมพันธ์กันระหว่าง Test Case กับข้อกำหนดที่ได้รับมาจากผู้ใช้งานระบบ

4. เพิ่มความมั่นใจว่า Test Asset ต่าง ๆ ถูกนำมาใช้งานเป็น version ล่าสุด ช่วยวางแผนการทำงานล่วงหน้าตั้งเวลา Execute หรือรัน Test Script ได้

8.สรุปผลการวิจัย

8.1 ผลการศึกษา

ผลจากการศึกษาแสดงว่า การทดสอบซอฟต์แวร์แอพพลิเคชั่น โดยใช้วิธีการทดสอบแบบอัตโนมัติใช้เวลาเพียงร้อยละ 19.26 ของเวลาที่ทดสอบด้วยด้วยมือ การลดเวลาที่ใช้ในการทดสอบทำให้

เกิดผลที่ตามมาคือ ลดแรงงานคน ลดค่าใช้จ่าย และลดเวลาการทำงานทดสอบ และผลลัพธ์ที่ดีอีกข้อหนึ่งคือ ลดความผิดพลาดในการบันทึก ข้อมูลด้วยแรงงานคน เมื่อต้องบันทึกข้อมูลเพื่อการทดสอบเป็นจำนวนมาก หรือบันทึกข้อมูลเดิมซ้ำหลายครั้ง

8.2 ข้อเสนอแนะ

1. เมื่อมีกรณีทดสอบ (Test Case) เป็นจำนวนมากที่ต้องใช้ในการบริหารงาน
2. ในการบริหารงานต้องทำกรณีทดสอบเดียวกันซ้ำหลายครั้ง ในรอบการทดสอบครั้งเดียว การใช้ Automation Testing จะช่วยให้ไม่ต้องบันทึก ข้อมูลซ้ำ ลดเวลา แรงงาน และค่าใช้จ่าย
3. การใช้ Automation Testing จะช่วยเรื่องการแก้ไขกรณีทดสอบตามสถานการณ์การทดสอบที่ต่างกัน
4. เมื่อต้องทำรายงานเป็นเอกสารหรือการวิเคราะห์งาน การใช้ Automation Testing จะทำได้สะดวกรวดเร็วและถูกต้อง
5. จะได้รายงานผลการทดสอบเป็นการบันทึกข้อมูล รายละเอียดผลการทดสอบว่า Pass หรือ Fail วันที่ เวลาทดสอบ ผู้ทดสอบ
6. รูปแบบการใช้งานที่ยังมีบางส่วนเป็น code อยู่บ้างทำให้ผู้ที่ไม่มีพื้นฐานทางการเขียนโปรแกรมใช้งานค่อนข้างยาก
7. Keyword ยังไม่ครอบคลุมการทำงานทั้งหมด ในบ้างครั้งยังจำเป็นต้องทำ Keyword เฉพาะทางขึ้นมาใช้งาน

ข้อเสนอแนะในการทำวิจัยครั้งต่อไป

1 รูปแบบการใช้งานที่ยังมีบางส่วนเป็น Code อยู่บ้างทำให้ผู้ที่ไม่มีพื้นฐานทางการเขียนโปรแกรมใช้งานค่อนข้างยาก จึงควรทำวิจัยครั้งต่อไปศึกษาการใช้งานที่เกี่ยวข้องกับ Code น้อยลง

2 Keyword ยังไม่ครอบคลุมการทำงานทั้งหมด ในบ้างครั้งยังจำเป็นต้องทำ Keyword เฉพาะทางขึ้นมาใช้งาน จึงควรทำวิจัยการพัฒนาการเขียน Keyword ที่ใช้ในการทดสอบ

3 ศึกษา Automation Testing Tool ที่มีการพัฒนาขึ้นใหม่ที่สามารถใช้ทดสอบซอฟต์แวร์ที่มีประสิทธิภาพในการค้นหาข้อบกพร่อง เพื่อเพิ่มความสามารถในการทดสอบที่เพิ่มขึ้น

บรรณานุกรม

ภาษาไทย

วิทยานิพนธ์และสารนิพนธ์

การเขียน Test Case IEEE Standard for Software Test Documentations, IEE Std829-1998(Sep 16), IEEE press, Newyork, Ny, 1998.

ธนพล สิชฌนุกฤษฎ์. (2554). A Development of a software test case management system (วิทยานิพนธ์ปริญญามหาบัณฑิต). กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย.

ณัฐรัตน์ หาญวรวงศ์. (2556). การออกแบบและพัฒนาการสร้างกรณีทดสอบซอฟต์แวร์แบบอัตโนมัติโดยใช้ดครงสร้าง UI (วิทยานิพนธ์มหาบัณฑิต). กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย.

รังสิต ศิริรังษี. (2557). การทดสอบซอฟต์แวร์ (Software Testing) (พิมพ์ครั้งที่ 1). เชียงใหม่: โรงพิมพ์เชียงใหม่ นพบุรีการพิมพ์.

รัชนก ชัยประเสริฐ. (2013). A test automation framework in POCT system using TDD techniques (วิทยานิพนธ์ปริญญามหาบัณฑิต). กรุงเทพฯ: มหาวิทยาลัยมหิดล.

วิบูลย์ ชัยจิราภรณ์. (2556). Why Automated Testing. วารสารจีแม๊กแซต, 35(10-12).

วิบูลย์ ชัยจิราภรณ์. (2557). การทำ Function Test ด้วย Automated Test Tool. วารสารจีแม๊กแซต, 37(12-14).

ศศิวิมล เย็นไสว. (2558). แนวทางการแก้ไขปัญหาการทดสอบซอฟต์แวร์ด้วยวิธี ATDD และการบริหารผลการปฏิบัติงาน (สารนิพนธ์วิทยาศาสตร์มหาบัณฑิต). กรุงเทพฯ: มหาวิทยาลัยเทคโลโลยีพระจอมเกล้าพระนครเหนือ.

สกรณ์ บุษบง. (2556). การสร้างกรณีทดสอบแบบเพิ่มทีละหน่วยโดยอัตโนมัติจากรณีทดสอบระดับหน่วย (วิทยานิพนธ์ปริญญามหาบัณฑิต). นครราชสีมา: มหาวิทยาลัยเทคโนโลยีสุรนารี.

สุรัส ตั้งไพบูลย์. (2547) . เทคนิคการลดความสูญเสียในโรงงานอุตสาหกรรม. กรุงเทพฯ: ส.เสริมมิตรการพิมพ์

Jay Heizer' S Barry Rendr. (2005). Operation Management. จินตนัย ไพรสณฑ์ และคณะ. การจัดการการผลิตและการปฏิบัติการ (พิมพ์ครั้งที่ 10). กรุงเทพฯ: เพียร์สัน เอ็ดดูเคชั่น อินโดไชน่า

ภาษาต่างประเทศ

Acceptance Test Driven Development. สืบค้นวันที่ 1 กุมภาพันธ์ 2560, จาก http://agilethailand,wordpress.com/2012

Alazar Seydum Hail. (2011). Automation of test cases for web applications automation of CRM test cases. Helsinki Metropolia University of Applied Sciences..

Automation Testing. สืบค้นวันที่ 1 กุมภาพันธ์ 2560, จาก https://www.testing-whiz.com/blog

Automation Testing. สืบค้นวันที่ 1 กุมภาพันธ์ 2560, จาก http://red.badger.com/blog/

Automation Testing. สืบค้นวันที่ 1 กุมภาพันธ์ 2560, จาก https://www.medium.com//Automation

Elisabeth Hendrickson. (2008). Driving Development with Tests; ATDD and TDD. Quality Tree Software, Inc.,

Functional Testing Process. สืบค้นวันที่ 6 กุมภาพันธ์ 2560, จาก http://sdlcservices.com/functional-testing.html

Functional Testing. สืบค้นวันที่ 6 กุมภาพันธ์ 2560, จาก https://charathbank.wordpress.com/2010/10/04/qa-knowledge-testing-type/

Filip Kiss. (2017). Analysis of Lean Software Development. Masaryk University.Czech Republic.

Jusaj Huska. (2012). Automated testing of the component best web application user interface. Masaryk University. Czech Republic.

Phat Chau. Tan. (2016). Automation testing with robot framework. Helsinki Metropolia University of Applied Sciences.

Robot Framework Architecture and Test Flow. สืบค้นวันที่ 6 มีนาคม 2560, จาก http://robotframework.org//#documation[online]

Robot Framework. สืบค้นวันที่ 6 มีนาคม 2560, จาก http://robotframework.org/#tools

Robot Framework. สืบค้นวันที่ 6 มีนาคม 2560, จาก https://networks.nokia.com/Wikipedia, The Free Encyclopedia

https://en.wikipedia.org//wiki//Automationtesting

https://en.wikipedia.org//wiki//SoftwareTesting

https://en.wikipedia.org//wili/ManualTesting