

การพัฒนาการทดสอบซอฟต์แวร์อัตโนมัติสำหรับเว็บแอปพลิเคชัน กรณีศึกษาระบบ การนำเข้าอุปกรณ์ และโอนย้ายอุปกรณ์

THE DEVELOPMENT OF AUTOMATION TESTING ON WEB APPLICATION: THE CASE STUDY OF IMPORT AND TRANSFER EQUIPMENT INFORMATION SYSTEM

ธวัชศักดิ์ บุรณะถาวร¹

ผู้ช่วยศาสตราจารย์ ดร.วรภัทร ไพรเกรง²

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาการใช้งานเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ ในการตรวจสอบการทำงานของซอฟต์แวร์ให้ถูกต้องตามข้อกำหนดของผู้ใช้งาน ผู้วิจัยได้ทำการศึกษาเปรียบเทียบ การทดสอบซอฟต์แวร์เว็บแอปพลิเคชันด้วย Automation Testing โดยใช้ Playwright จาก Microsoft กับการทดสอบแบบ Manual Testing ในการทดสอบด้วยวิธี Manual Testing 20 Test cases จำนวน 5 ครั้ง โดยใช้ผู้ทดสอบ 3 คนภายในทีมแล้วหาค่าเฉลี่ยเวลาในการทดสอบ จากนั้นทดสอบด้วย Automation Testing 20 Test cases เดียวกัน จำนวน 5 ครั้ง จากการศึกษาพบว่าค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบแบบ Automation Testing น้อยกว่าเวลาที่ใช้ในการทดสอบแบบ Manual Testing สรุปได้ว่าผลการศึกษาเป็นไปตาม วัตถุประสงค์กล่าวคือการใช้ Automation Testing ในการทดสอบซอฟต์แวร์เว็บแอปพลิเคชันทำให้ลด ระยะเวลาในการทดสอบ รายงานผลการทดสอบได้รวดเร็ว ลดความผิดพลาดในการบันทึกข้อมูล ลด แรงงานคน สามารถทำแบบเดิมซ้ำ ๆ ได้โดยไม่ต้องใช้คนในการควบคุม และถูกต้องครบถ้วนตาม ข้อกำหนด

คำสำคัญ: ซอฟต์แวร์, เว็บแอปพลิเคชัน, การทดสอบ, Manual Testing, Automation Testing

¹ นักศึกษาหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยธุรกิจบัณฑิต

² อาจารย์ที่ปรึกษาสารนิพนธ์

ABSTRACT

The objective of this research was to study the use of automated software testing tools to guarantee the functionality of the software according to user requirements. The study was conducted by the execution of software testing with an approach of Automation Testing : the Playwright of Microsoft, in comparison to Manual Testing. In Manual Testing, 20 test cases were executed 5 times by 3 testers of the team to find the average testing time. The similar 20 test cases were done in Automation Testing and the time used was recorded too. According to the experiment, the average time used in Automation Testing was less than in Manual Testing. The results of the study are in accordance with the objective of using Automation Testing. It appeared that Automation software testing can help to reduce time, run the repetitive tests without human control, reduce human efforts and errors, provide fast test report, improve test coverage and complete the accurate task.

Keywords: Software, Web Application, Testing, Manual testing, Automation Testing

1. บทนำ

เมื่อมีการพัฒนาแอปพลิเคชันขึ้นมา จะต้องมีการทดสอบตรวจสอบหาข้อผิดพลาดของแอปพลิเคชัน เพื่อให้มั่นใจว่าแอปพลิเคชันที่ทำการพัฒนานั้นสามารถนำมาใช้งานได้ถูกต้องและไม่พบปัญหาที่จะเกิดขึ้นในภายหลัง กระบวนการทดสอบ (Software Testing) มีหลายขั้นตอน ประกอบด้วย ทดสอบระดับหน่วยย่อย (Unit Test) การทดสอบรวมหน่วย (Integration Test) การทดสอบระบบ (System Test) และการทดสอบแบบยอมรับ (Acceptance Test) ในทุกขั้นตอนมุ่งเน้นไปที่การค้นหาข้อผิดพลาดของแอปพลิเคชันที่ซ่อนอยู่ให้เจอเพื่อทำการแก้ไข

การทดสอบซอฟต์แวร์ (Software Testing) มีความสำคัญในการพัฒนาซอฟต์แวร์ เนื่องจากการทดสอบเป็นส่วนในขั้นตอนของวงจรพัฒนาซอฟต์แวร์ (Software Development life cycle) โดยเป็นขั้นตอนการค้นหาข้อผิดพลาด (Error) ที่มีในระบบ และระบุสาเหตุที่เกิดขึ้น ส่งผลกลับไปให้นักพัฒนาซอฟต์แวร์สามารถแก้ปัญหา ทำให้ซอฟต์แวร์ทำงานได้ตรงตามความต้องการและมีประสิทธิภาพ ซึ่งการทดสอบสามารถแบ่งได้ 2 ประเภท คือ การทดสอบด้วยตนเอง (Manual Testing) และการทดสอบแบบอัตโนมัติ (Automated Testing)

ในกรณีนี้จะเป็นการศึกษาการควบคุมเกี่ยวกับเว็บแอปพลิเคชัน ซึ่งในปัจจุบันทีมของผู้จัดทำทดสอบระบบโดยเทคนิค Manual Testing ในการตรวจหาข้อผิดพลาดในระบบดังกล่าวเพียงอย่างเดียว จึงอาจทำให้พบปัญหาต่าง ๆ จากผู้ใช้งาน ที่ต้องจะได้รับการแก้ไขในเวอร์ชันต่อ ๆ ไปของเว็บแอปพลิเคชัน

อย่างไรก็ตามตั้งแต่มีเว็บแอปพลิเคชันดังกล่าวเกิดขึ้นยังไม่เคยมีการทำทดสอบด้วยการทำ Automated Test มาก่อน โดยวิธีการดังกล่าวจะช่วยลดระยะเวลาในการทดสอบเว็บแอปพลิเคชันและมีความถูกต้องแม่นยำมากยิ่งขึ้นกว่าวิธีการทำ Manual Test แบบเดิม ที่ใช้คนในการตรวจสอบที่มีการใช้เวลาค่อนข้างนานในการเตรียมข้อมูลเพื่อนำเข้า ซึ่งวิจัยฉบับนี้จะเป็นการนำเสนอข้อมูลทางทฤษฎีและปฏิบัติเกี่ยวกับกระบวนการดำเนินการทดสอบแอปพลิเคชัน แบบ Automated Test ในการพัฒนารวมถึงผลการปฏิบัติงานด้านการทดสอบระบบนำเข้าอุปกรณ์ และการโอนย้ายอุปกรณ์

วัตถุประสงค์ของการวิจัย

1. เพื่อเพิ่มประสิทธิภาพการในการทดสอบระบบสารสนเทศของการโอนคลังสินค้า โดยมีวิธี Automated Testing
2. เพื่อสามารถระบุแหล่งที่มาและสาเหตุของข้อผิดพลาดในการพัฒนาแอปพลิเคชัน และมีแนวทางแก้ไขปัญหที่เกิดขึ้น

2.งานวิจัยที่เกี่ยวข้อง

Abhijit A. Sawant , Pranit H. Bari , P.M. Chawan กล่าวในงานวิจัยเรื่อง “software Testing Techniques and Strategies” ระบุว่า “การทดสอบซอฟต์แวร์เป็นการตรวจสอบความถูกต้องของซอฟต์แวร์ทั้งเชิงทดลองและอย่างเป็นระบบ” และอธิบายว่า “การทดสอบซอฟต์แวร์เป็นเทคนิคที่มุ่งประเมินคุณลักษณะหรือความสามารถของโปรแกรมหรือผลิตภัณฑ์และพิจารณาว่า มีคุณภาพตรงตามที่ต้องการ” โดยเสริมว่า “การทดสอบซอฟต์แวร์ยังใช้เพื่อทดสอบปัจจัยด้านคุณภาพอื่นๆ ของซอฟต์แวร์ เช่น ความเชื่อถือได้ ความยากง่ายของการทำงาน ความครบถ้วนสมบูรณ์ ความปลอดภัย ประสิทธิภาพ ความสะดวกในการพกพา การบำรุงรักษา การใช้เทียบแทนกันได้ เป็นต้น ความจำเป็นในการทดสอบซอฟต์แวร์ และเป้าหมายของการทดสอบซอฟต์แวร์ Abhijit A. Sawant กล่าวว่า การพัฒนาซอฟต์แวร์ตามชุดข้อกำหนดจำเป็นต้องมีการทดสอบซอฟต์แวร์ เพื่อตรวจสอบและรับรองว่าซอฟต์แวร์ที่ถูกสร้างขึ้นได้ตรงตามข้อกำหนด

R.M. Shama ได้กล่าวในงานวิจัยเรื่อง “Quantitative Analysis of Automation and Manual Testing” กล่าวถึงการทดสอบด้วยตนเอง (Manual Testing) เป็นกิจกรรมที่ต้องใช้ความพยายามอย่างมาก ซึ่งกำหนดให้ผู้ทดสอบที่คุณสมบัติบางประการได้แก่ อดทน ช่างสังเกต ช่างสงสัย สร้างสรรค์ มีความคิดใหม่ๆ ใจเปิดกว้างที่จะยอมรับความคิดใหม่ และมีทักษะเชี่ยวชาญ

ขั้นตอนในการทำทดสอบโดยบุคคล (Manual Testing)

- การวิเคราะห์ความต้องการ (Requirements Analysis) ศึกษาเอกสารโครงการ คู่มือและแอปพลิเคชันที่ต้องการทดสอบ วิเคราะห์ข้อกำหนดจาก SRS (Software Requirements Specification) ข้อกำหนดที่ถูกกำหนดขึ้นในกระบวนการพัฒนาระบบ
- การสร้างแผนการทดสอบ (Test Plan Creation) สร้างแผนการทดสอบที่ครอบคลุมข้อกำหนดทั้งหมดของซอฟต์แวร์
- การสร้างกรณีทดสอบ (Test Case Creation) ออกแบบการทดสอบ (Test Case) ที่ครอบคลุมข้อกำหนดทั้งหมดที่อธิบายไว้ในเอกสารประกอบ หลักในการเขียน Test Case เพื่อการยืนยันความถูกต้อง (Verification) ของการใช้งานระบบจะถูกออกแบบให้มีเป้าหมายให้ระบบใช้งานได้ตรงตามความต้องการ (Requirements) และเงื่อนไข (Conditions) โดยจำลองสถานการณ์ที่จะเกิดขึ้น ตรงความต้องการของผู้ใช้งาน สมการ Test Case เป็นจุดเริ่มต้นในการเริ่มเขียน Test Case โดยสมการสรุปได้ดังนี้ $\text{Test Case} = \text{Test Rule} + \text{Test Data}$
 - Test Rule คือกฎของการทดสอบที่เป็นการวิเคราะห์จากความต้องการ (Requirements) ว่าจะต้องทดสอบความต้องการผ่านเงื่อนไขอะไรบ้าง Test Rule จะเป็นการรวบรวมเงื่อนไขต่างๆ ของความต้องการเพื่อนำมาเป็นสารตั้งต้นในการออกแบบ Test Case
 - Test Data คือข้อมูลในการทดสอบ ข้อมูลที่จะนำมาใช้ทดสอบจะต้องเป็นข้อมูลที่เสมือนจริงหรือในบางเคสอาจจะใช้ข้อมูลจริงมาเป็น Test Data และที่สำคัญข้อมูลจะต้องมีปริมาณที่มาก รวมไปถึงต้องเป็นข้อมูลที่มีคุณภาพ เพื่อการทดสอบจะได้ออกมาสมบูรณ์มากที่สุด
- ดำเนินกรณิการทดสอบ (Test Case Execution) ตรวจสอบและกำหนดแนวทางพื้นฐาน (Baseline) ของกรณีทดสอบกับหัวหน้าทีมและผู้ใช้งาน
- การบันทึกข้อบกพร่อง (Defect Logging) ตรวจสอบและรายงานให้นักพัฒนาทราบ
- แก้ไขข้อบกพร่อง และตรวจสอบอีกครั้ง (Defect Fix and Re-verification) เมื่อข้อบกพร่องได้รับการแก้ไขแล้ว ให้ดำเนินการทดสอบกรณีที่ล้มเหลวซ้ำจนกว่าจะผ่าน

งานวิจัยของ Shruti Malve and Pradeep Sharma (Dept. of Computer Science, Govt. Holkar Science (Autonomous) College, Indore, India) กล่าวถึงการทำ Automation Testing เมื่อเปรียบเทียบกับ การทดสอบด้วยตนเอง การทดสอบอัตโนมัติไม่เพียงแต่ดำเนินการกรณีทดสอบเท่านั้น แต่ยังรวมไปถึงการสร้างกรณีทดสอบและการตรวจสอบผลการทดสอบด้วย โดยการทดสอบอัตโนมัติดำเนินการต่าง ๆ โดยที่ผู้ใช้ไม่ต้องดำเนินการใด ๆ ยังหมายถึงการทดสอบระบบด้วยชุดข้อมูลที่แตกต่างกันอย่างต่อเนื่องโดยไม่มีแทรกแซงของมนุษย์

งานวิจัยของ Bhawna Kumari , Naresh Chauhan , Vedpal เรื่อง “A Comparison Between Manual Testing and Automated Testing” ได้อภิปรายเกี่ยวกับรูปแบบการทดสอบโดยบุคคล และรูปแบบการ

ทดสอบโดยขบวนการอัตโนมัติ และพบว่ามีข้อดีอย่างมากมาใช้ในการทดสอบโดยบุคคล เพื่อให้ข้อดีเหล่านี้ผ่านไปได้จึงมีการดำเนินการทดสอบโดยขบวนการอัตโนมัติ

กระบวนการทดสอบแบบอัตโนมัติ (Automated Testing)

-วิเคราะห์ความเป็นไปได้ของกระบวนการทดสอบซอฟต์แวร์โดยอัตโนมัติ (Test Automation Feasibility Analysis) ตรวจสอบว่าแอปพลิเคชันสามารถตรวจสอบด้วยวิธีอัตโนมัติได้หรือไม่

-เลือกเครื่องมือที่เหมาะสม (Appropriate Tool Selection) การเลือกเครื่องมือขึ้นอยู่กับเทคโนโลยีที่สร้างแอปพลิเคชัน คุณสมบัติและการใช้งานของแอปพลิเคชัน

-ประเมินเฟรมเวิร์ค (กรอบการทำงาน) ที่เหมาะสม มีเฟรมเวิร์คที่หลากหลายและแต่ละเฟรมเวิร์คมีลักษณะที่สำคัญในตัวเอง

-สร้าง POC (Proof of Concept) POC จะดำเนินการด้วยสถานการณ์แบบ end-to-end ซึ่งช่วยให้มั่นใจได้ว่าฟังก์ชันการทำงานหลักๆ จะเป็นไปได้โดยอัตโนมัติ

-จัดทำกรอบการทำงานอัตโนมัติ (Develop Automation Framework) เฟรมเวิร์คควรถูกสร้างขึ้นหลังจากการวิเคราะห์เทคโนโลยีที่ใช้โดยแอปพลิเคชันรวมถึงคุณสมบัติ (ฟีเจอร์/Features) ที่สำคัญของแอปพลิเคชัน

-จัดทำสคริปต์ทดสอบ ดำเนินการทดสอบ และวิเคราะห์ (Develop Test Script , Execute and Analyze) เมื่อจัดทำสคริปต์เสร็จสิ้น ดำเนินการทดสอบตามสคริปต์ วิเคราะห์ผลลัพธ์ และบันทึกข้อบกพร่องถ้ามี

3.ผลการดำเนินงาน

การศึกษาและทดสอบเว็บแอปพลิเคชัน ของบริษัทกรณิศศึกษาโดยใช้ Automation Testing เพื่อใช้สำหรับสร้างกรณีทดสอบเว็บแอปพลิเคชัน จึงได้นำมาสู่การออกแบบใช้บันทึกข้อมูลที่เกี่ยวข้องและพัฒนา ระบบ ซึ่งงานวิจัยนี้ได้พัฒนาระบบและพัฒนาโปรแกรมสำหรับทำ Automation Testing เพื่อใช้ทดสอบเว็บแอปพลิเคชัน ซึ่งมีรายละเอียดดังต่อไปนี้

3.1 Code ส่วนการทำงานนำเข้าอุปกรณ์สำเร็จ หรือ TC_01_1_ Import อุปกรณ์

```
30 const totePerBo = 16;
31
32 test.describe.serial('Automation test Tote import', () => {
33   let arrayCode: string[];
34   let firstTote: string;
35   let lastTote: string;
36   let newCode: string;
37
38   test.beforeAll(async () => {
39     const lastCodeFromSQL = await getLastToteCodeFromTPdtMTote();
40     newCode = await generateCodeName(lastCodeFromSQL);
41     arrayCode = await getCodeByAmount(newCode, totePerBo);
42     await generateExcel(arrayCode);
43     firstTote = arrayCode[0];
44     lastTote = arrayCode[arrayCode.length - 1];
45   });
46   test('#1 import tote new', async () => {
47     loginPage,
48     mainPage,
49     importToteByExcel,
50     searchBranch,
51     uploadFile,
52   }) => {
53     await loginPage.login(WEB_TOTE_USERNAME!, WEB_TOTE_PASSWORD!);
54     await searchBranch.searchByBranchCode(D001_CODE!);
55     await mainPage.goToTab('นำเข้าข้อมูล Tote');
56     await uploadFile.uploadToteByExcel(EXCEL_FILE_NAME!, true);
57     const queryFirstData = await getToteFromTPdtMTote(firstTote, lastTote);
58     await importToteByExcel.verifyToteStatus(queryFirstData, '5');
59     await updateStatusToteToZeroFromTPdtMTote(firstTote, lastTote);
60     const querySecondData = await getToteFromTPdtMTote(firstTote, lastTote);
61     await importToteByExcel.verifyToteStatus(querySecondData, '0');
62   });
63 }
```

ภาพที่ 3.1 แสดง Code ส่วนของ TC_01_1

3.2 Code ส่วนการทำงานโอนอุปกรณ์ไปยังสาขาโดยใช้ไฟล์จากการนำเข้าอุปกรณ์ ในข้อแรก หรือ TC_02_1_ โอนอุปกรณ์ไปสาขา

```
19 test.describe.serial('Automation test Tote import', () => {
20   let arrayCode: string[];
21   let groupToteData1: string[];
22   let groupToteData2: string[];
23   let groupToteData3: string[];
24   let groupToteData4: string[];
25   let groupToteData5: string[];
26   let groupToteData6: string[];
27   let groupToteData7: string[];
28   let groupToteData8: string[];
29
30   test.beforeAll(async () => {
31     arrayCode = readExcelFile(EXCEL_FILE_NAME!);
32   });
33
34   test('[-]งานโอน Tote ไปสาขาไม่ซ้ำข้อมูลโอน tote เป็น 8 เลขา (เลขาละ 10 tote) โอนมีสาขาปลายทางเป็นสี่ตัว (M04C)', async () => {
35     loginPage,
36     mainPage,
37     searchBranch,
38     dcTransferToBranch,
39     page,
40   }) => {
41     await loginPage.login(WEB_TOTE_USERNAME!, WEB_TOTE_PASSWORD!);
42     await searchBranch.searchByBranchCode(D001_CODE!);
43     await mainPage.goToTab('โอน Tote ไปสาขา');
44     // ไม่ซ้ำข้อมูลโอน tote เป็น 8 เลขา (เลขาละ 10 tote)
45
46     const groupToteData = await divideArrayBySize(arrayCode, 2);
47     // เลขาชุด 1
48     await page.waitForLoadState('domcontentloaded');
49     await dcTransferToBranch.addButton.click();
50     await page.waitForLoadState('domcontentloaded');
51     await dcTransferToBranch.detailButton.click();
52     await dcTransferToBranch.filterDestinationButton.click();
53     await searchBranch.searchByBranchCode(M04C_CODE!);
54     await dcTransferToBranch.toteListButton.click();
55     groupToteData1 = await groupToteData[0];
56     await dcTransferToBranch.genFileAndUpload(groupToteData1, 'Group1.xlsx');
57     // await dcTransferToBranch.verifyTableToteData(groupToteData1);
58     await dcTransferToBranch.saveAndVerifySQLAndApprove({
59       branchFrom: D001_BRANCH_NAME,
60       branchFromCode: D001_CODE,
61       branchTo: M04C_BRANCH_NAME,
62       branchToCode: M04C_CODE,
63       toteData: groupToteData1,
64     });
65   });
66 }
```

ภาพที่ 3.2 แสดง Code ส่วนของ TC_02_1

3.3 Code ส่วนการทำงานของ TC_03_1 โอนอุปกรณ์ กลับสำนักงานใหญ่/อนุมัติเรียบร้อยแล้ว

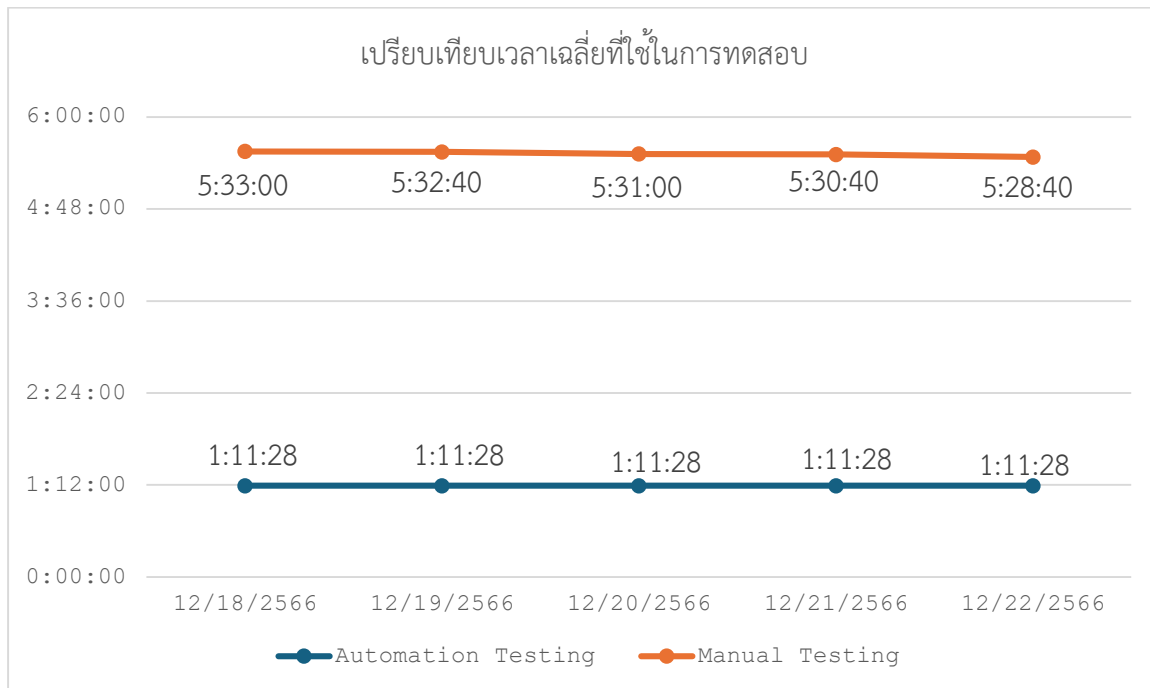
[illegible]

ภาพที่ 3.3 แสดง Code ส่วนของ TC_03_1

การวิเคราะห์เปรียบเทียบเวลาที่ใช้ในการทดสอบเว็บแอปพลิเคชัน ของบริษัทกรณีศึกษา โดยผู้วิจัยนำผลของการทดสอบเวลาที่ใช้ในการทดสอบแบบ Automate testing และการทดสอบแบบ Manual Testing มาวิเคราะห์เปรียบเทียบเวลาที่ใช้ในการทดสอบ

ตารางที่ 3.1 เปรียบเทียบเวลาเฉลี่ยที่ใช้ในการทดสอบ

ครั้งที่	Test Case (D/M/Y)	AVG Time Take by Automation Testing	AVG Time Take by Manual Testing	Duration Minute
1	20 Test Case: 18/12/2566	1:11:28	5:33:00	4:21:32
2	20 Test Case: 19/12/2566	1:11:28	5:32:40	4:21:12
3	20 Test Case: 20/12/2566	1:11:28	5:31:00	4:19:32
4	20 Test Case: 21/12/2566	1:11:28	5:30:40	4:19:12
5	20 Test Case: 22/12/2566	1:11:28	5:28:40	4:17:12
Total		5:57:20	27:36:00	21:38:40



ภาพที่ 3.4 เปรียบเทียบเวลาเฉลี่ยที่ใช้ในการทดสอบ

จากตารางที่ 3.1 และภาพที่ 3.4 พบว่า ค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบเว็บแอปพลิเคชัน แบบ Manual Testing มากกว่าเวลาในการทดสอบแบบ Automate testing อย่างมีนัยสำคัญทุกครั้ง ซึ่งค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบแบบ Automate testing คิดเป็นร้อยละ 21.58 ของเวลาที่ใช้ในการทดสอบแบบ Manual Testing เป็นไปตามวัตถุประสงค์การใช้ Automate testing ที่ช่วยลดเวลาในการทดสอบเว็บแอปพลิเคชัน

4.สรุป และการอภิปรายผล

ผลการศึกษาการใช้ Automate testing ในการทดสอบเว็บแอปพลิเคชัน บรรลุวัตถุประสงค์ การพัฒนาขั้นตอนการทดสอบเว็บแอปพลิเคชัน ใช้งานได้กับทุกกรณีทดสอบ และการใช้ Automate testing ทดสอบเว็บแอปพลิเคชัน สามารถลดเวลาการทดสอบได้จริง โดยผู้วิจัยเปรียบเทียบค่าเฉลี่ยของการทดสอบทั้ง 2 แบบ จำนวนเวลาที่ได้อ้างอิงเวลาของการทดสอบแบบ Automation ใช้เวลาคิดเป็นร้อยละ 21.58 ของเวลาที่ใช้ในการทดสอบแบบ Manual Testing และเมื่อตรวจสอบรายการ ซึ่งมี 20 กรณีทดสอบเวลาที่ใช้ในการทดสอบเทียบกันแต่ละกรณีทดสอบ จำนวนเวลาที่ใช้ในการทดสอบแบบ Automation Testing น้อยกว่าเวลาที่ใช้ทดสอบแบบ Manual Testing ทุกกรณีอย่างมีนัยสำคัญ

ผลจากการศึกษาแสดงว่า การทดสอบเว็บแอปพลิเคชัน โดยใช้วิธีการทดสอบแบบ Automation Testing ใช้เวลาคิดเป็นร้อยละ 21.58 ของเวลาที่ใช้ในการทดสอบแบบ Manual Testing ซึ่งการลดเวลาที่ใช้ในการทดสอบส่งผลทำให้ ลดต้นทุนค่าแรงงาน ลดค่าใช้จ่าย และลดเวลาการทำงานทดสอบ นอกจากนี้ยังช่วยลดความผิดพลาดในการบันทึกข้อมูลด้วยแรงงานคน เพราะเมื่อต้องบันทึกข้อมูลที่ทำทดสอบเป็นจำนวนมาก หรือบันทึกข้อมูลเดิมซ้ำหลายครั้ง ซึ่งผลจากการพัฒนาโปรแกรมการใช้งานของ Automation Testing Tool ระหว่างการทดสอบ พบว่ามีประเด็นสำคัญดังนี้

1. ช่วยเพิ่มความรวดเร็วในการพัฒนาระบบการทดสอบ
2. ทำให้เกิดความร่วมมือที่มีประสิทธิภาพระหว่าง Tester Team และ Developer Team จากการดำเนินการทดสอบ
3. ช่วยในการบริหารจัดการ Test Resource และการเชื่อมความสัมพันธ์กันระหว่าง Test Case กับข้อกำหนดที่ได้รับมาจากผู้ใช้งานระบบ
4. เพิ่มความมั่นใจว่า Test Asset ต่าง ๆ ถูกนำมาใช้งานเป็น Version ล่าสุด
5. ช่วยวางแผนการทำงานล่วงหน้า และสามารถตั้งเวลา Execute หรือรัน Test Script ได้อย่างมีประสิทธิภาพ

5.ข้อเสนอแนะ

บริษัทหรือองค์กรธุรกิจมีความจำเป็นต้องใช้ซอฟต์แวร์เข้ามาช่วยในการทำงาน เพราะปัจจุบันนี้บริษัทหรือองค์กรธุรกิจ มีความเกี่ยวข้องกับตัวแบบทางธุรกิจ (Business Model) ที่มีความยุ่งยาก และซับซ้อนมากยิ่งขึ้น จึงทำให้การบริหารงานมีความจำเป็นที่จะต้องใช้เครื่องมือที่มีประสิทธิภาพ หรือซอฟต์แวร์มาเป็นเครื่องมือช่วยเพิ่มความเร็วในการทำงาน เป็นเครื่องมือช่วยจัดการอย่างถูกต้องและมีประสิทธิภาพ ลดความเสี่ยงการนำซอฟต์แวร์เข้ามาใช้งาน ซึ่งการตรวจสอบการทำงานของซอฟต์แวร์ที่พัฒนาขึ้นใช้งานว่าทำงานได้ถูกต้องตรงตามข้อกำหนดของผู้ใช้ ดังนั้น จึงมีความสำคัญในการเลือก Automation Testing Tool เช่น

การใช้ Automation Testing จะช่วยให้ไม่ต้องบันทึกข้อมูลซ้ำ ช่วยลดเวลา ช่วยลด แรงงาน และช่วยลดค่าใช้จ่ายได้อย่างมีประสิทธิภาพ เมื่อมีกรณีทดสอบ (Test Case) เป็นจำนวนมาก หรือต้องทำการทดสอบเดียวกันซ้ำหลายครั้งในรอบการทดสอบครั้งเดียว ซึ่งทำให้เกิดความสะดวกรวดเร็ว และถูกต้อง เมื่อต้องทำรายงานหรือการวิเคราะห์งาน และยังช่วยเรื่องการแก้ไขกรณีทดสอบตามสถานการณ์ของการทดสอบที่ต่างกันได้อย่างมีประสิทธิภาพ ซึ่งการพัฒนาต่อในอนาคตจะเป็นลักษณะของการบันทึกข้อมูลอื่นๆ เพิ่มเติม เพื่อสร้างรายงาน อื่นๆ เพื่อนำไปใช้สำหรับการปรับปรุงการทดสอบ อย่างไรก็ตามการใช้ Automation Testing ซึ่งรูปแบบการใช้งานบางส่วนยังเป็น code จึงส่งผลทำให้ผู้ใช้งานที่ไม่มีพื้นฐานการ

เขียนโปรแกรมใช้งานก่อนข้างยาก อาจจะเป็นการศึกษาวิธีการพัฒนา Automated test ที่ใช้การหลักการ low code เพิ่มขึ้น รวมถึง เทคนิคใหม่ๆ ที่ใช้ในการพัฒนา Automated test ในอนาคต

บรรณานุกรม

Abhijit A. Sawant , Pranit H. Bari , and P.M. Chawan. Software Testing Techniques and Strategies.

Retrieved from

https://www.researchgate.net/publication/316510706_Software_Testing_Techniques_and_Strategies

Mehwash Rafiq , Rehan Ashraf , and Haris Abid. Automated VS Manual Testing: A Scenario Based

Approach Towards Application Development. Retrieved from

https://gjecs.gyancity.com/Vol5No1/GJECS_51006.pdf

R.M. Sharma. Quantitative Analysis of Automation and Manual Testing. Retrieved from

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4705f69033ea278213b4980326d7ba893e1e1cfd>

Khin Shin Thant and Hlaing Htake Khaung. The Impact of Manual and Automatic Testing on Software Testing Efficiency and Effectiveness. Retrieved from

https://www.researchgate.net/publication/370526552_THE_IMPACT_OF_MANUAL_AND_AUTOMATIC_TESTING_ON_SOFTWARE_TESTING_EFFICIENCY_AND_EFFECTIVENESS

Shruti Malve , and Pradeep Sharma. Investigation of Manual and Automation Testing using Assorted Approaches. Retrieved from

https://www.isroset.org/pdf_paper_view.php?paper_id=408&IJSRCSE-0293-55.pdf