

SAC-01-013

การเปรียบเทียบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ Performance Comparison of Automated Software Testing Tools

ชนสิษฐ์ รุ่งระวีศรีศิริธร^{1*} และ ชัยพร เขมะภาคะพันธ์²

Chanasit Rungraveesrisasithon^{1*} and Chaiyaporn Khemapatapan²

สาขาวิชาวิศวกรรมคอมพิวเตอร์ วิทยาลัยวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยมหาวิทยาลัฏฐกิจบัณฑิตย
Department of Computer Engineering, College of Engineering and Technology, Dhurakij Pundit University
e-mail: chaiyaporn@dpu.ac.th

บทคัดย่อ

การวิจัยครั้งนี้ เป็นการวิจัยเชิงพัฒนาและเชิงทดลองมีวัตถุประสงค์เพื่อ (1) ออกแบบ และพัฒนารูปแบบการทดสอบแบบอัตโนมัติสำหรับเว็บแอปพลิเคชัน (2) เพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ทดสอบอัตโนมัติ คือ Selenium, Cypress และ Playwright และ (3) เพื่อลดระยะเวลาในการคีย์ข้อมูลที่ใช้ในการทดสอบระบบเว็บแอปพลิเคชัน การใช้ระบบการทดสอบอัตโนมัติมาประยุกต์ใช้ในการคีย์ข้อมูลในระบบเพื่อช่วยให้การคีย์ข้อมูลในระบบจำนวนมากช่วยในเรื่องการประหยัดเวลาและลดข้อผิดพลาดที่เกิดขึ้นจากการคีย์ข้อมูล จึงมีแนวคิดจะทำการทดสอบประสิทธิภาพในส่วนของการทดสอบอัตโนมัติและระยะเวลาในการทำงานของเครื่องมือโดยในการทดสอบจะทำการทดสอบแบบอัตโนมัติ และจะใช้เครื่องมือในการทดสอบอัตโนมัติจำนวน 3 ซอฟต์แวร์ คือ Selenium, Cypress และ Playwright โดยจะทำการสร้างเทสต์สคริปต์ Test Script และกำหนดรูปแบบขั้นตอน เพื่อให้ทำตามชุดคำสั่งที่ได้กำหนดไว้ ดังนั้นจึงมีแนวคิดที่จะนำการทดสอบอัตโนมัติมาช่วยในการคีย์ข้อมูล จากนั้นนำชุดข้อมูลที่ทำเตรียมไว้สำหรับการทดสอบในเครื่องมือที่ได้ทำการทดสอบ เพื่อทำการเปรียบเทียบประสิทธิภาพ ระยะเวลาของซอฟต์แวร์ทดสอบ, จำนวนความสำเร็จของซอฟต์แวร์ทดสอบ, ร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบ และประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลาของทั้ง 3 เครื่องมือที่จะใช้ในการทดสอบ โดยข้อมูลที่จะนำมาใช้ในการทดสอบการคีย์ข้อมูลมีการผ่านการตรวจสอบ และในการใช้ระบบการทดสอบอัตโนมัติสามารถทำงานซ้ำ ๆ ผลการวิจัยพบว่าเครื่องมือ Playwright ที่ใช้ในการทดสอบอัตโนมัติใช้ระยะเวลาของซอฟต์แวร์ทดสอบในการคีย์ข้อมูลน้อย มีจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมาก และร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมาก ทำให้แสดงถึงประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลาที่มากกว่า ดังนั้นจึงสรุปได้ว่าเครื่องมือ Playwright สามารถนำมาทดสอบเว็บแอปพลิเคชันได้ตรงตามความต้องการที่จะสามารถนำมาปรับใช้ และประยุกต์ทำให้เกิดประสิทธิภาพในการทดสอบเว็บแอปพลิเคชัน แต่มีข้อพิจารณาอยู่หลายด้านของซอฟต์แวร์ทดสอบอัตโนมัติ รูปแบบของการทดสอบ, โครงสร้างและเทคโนโลยีที่ใช้ในการพัฒนา, ภาษาโปรแกรมที่ใช้ในการพัฒนา และโครงสร้างพื้นฐานของระบบ

คำสำคัญ: ระบบการทดสอบอัตโนมัติ ซอฟต์แวร์ทดสอบ เทสต์สคริปต์ คีย์ข้อมูล

Abstract

This research is a developmental and experimental research with the following objectives: (1) Design and develop an automated testing model for web applications (2) Compare the efficiency of automated testing software: Selenium, Cypress, and Playwright and (3) Reduce the time spent on data entry in web application testing. Applying automated testing to data entry in the system helps save time and errors from data entry. Therefore, the idea is to test the efficiency of automated testing and the duration of the tools. The testing will be automated testing and 3 automated testing tools will be used: Selenium, Cypress, and Playwright. Test scripts will be created and steps will be defined to follow the specified commands. Therefore, the idea is to use automated testing to help with data entry. Then, the data set prepared for testing in the tested tools will be used to compare the efficiency, duration of the tested software, number of successful test software, percentage of successful test software, and the efficiency of the software operation per unit time of all 3 tools to be tested. The data to be used in data entry testing has been verified and the automated testing system can be used repeatedly. The research results found that the Playwright tool used in automated testing used less time for the testing software to enter data, had a high number

ของนักพัฒนา โดยรูปแบบการทดสอบออกเป็น 2 ส่วน คือการทดสอบการสแกน Hyperlink Reference จาก URL สามารถทำงานได้ 100% และการทดสอบมาตรฐาน WCAG และเปรียบเทียบกับ Extension Axe Developer Tool ซึ่งผลออกมาทำงานได้ใกล้เคียงถึง 95.238%

ธวัชศักร์ บุรณะถาวร (2566) แนวคิดการวิจัยศึกษาเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติในการตรวจสอบความถูกต้องของซอฟต์แวร์ที่มีการพัฒนาขึ้นของนักพัฒนาให้เป็นไปตามข้อกำหนดของผู้ใช้งาน โดยผู้วิจัยได้ออกแบบการทดลองเปรียบเทียบระหว่างการทดสอบเว็บแอปพลิเคชันด้วยวิธีแบบแมนนวล Manual Testing และการทดสอบแบบอัตโนมัติ Automation Testing โดยใช้เครื่องมือทดสอบซอฟต์แวร์ คือ Playwright ขั้นตอนการทดลองเริ่มจากการทดสอบด้วยวิธี Manual Testing โดยมีผู้ทดสอบภายในทีมจำนวน 3 คน และมีจำนวน 20 กรณีทดสอบ จากนั้นทำซ้ำ 5 รอบเพื่อคำนวณหาค่าเฉลี่ยของเวลาในการทดสอบ จากนั้นจึงดำเนินการทดสอบรูปแบบเดียวกันเพื่อเปรียบเทียบกับ Playwright ในรูปแบบของ Automation Testing จำนวน 20 กรณีทดสอบ และทำซ้ำ 5 รอบ ผลการทดลองพบว่า การใช้เครื่องมืออัตโนมัติสามารถลดระยะเวลาในการทดสอบ ดังนั้นทางผู้วิจัยได้สรุปการเปรียบเทียบค่าเฉลี่ยของการทดสอบแบบ Manual Testing และ Automation Testing โดยการทดสอบรูปแบบ Automation Testing ใช้เวลาร้อยละ 21.58 ของเวลาที่ใช้ในการทดสอบรูปแบบ Manual Testing

Donvir (2024) แนวคิดวิเคราะห์เฟรมเวิร์กเพื่อประเมินและเปรียบเทียบเฟรมเวิร์ก สร้างแนวทางปฏิบัติที่มีประสิทธิภาพในการเลือกใช้ Unit และ End-to-End Testing Frameworks วิธีดำเนินการเลือกเฟรมเวิร์กการทดสอบ เช่นการตั้งเกณฑ์การประเมินของเฟรมเวิร์ก การเปรียบเทียบและการวิเคราะห์ผลลัพธ์งานวิจัยมุ่งเน้นการทดลองการใช้งานเฟรมเวิร์ก พิจารณาจากความง่ายในการใช้งาน, คุณสมบัติความหลากหลาย, มี Community Support, ความสามารถของ Scalability, ความสามารถของเฟรมเวิร์กในการเชื่อมต่อและทำงานร่วมกัน CI/CD pipelines รวมถึงการนำเสนอข้อแนะนำและแนวทางการเลือกเฟรมเวิร์กที่เหมาะสม Cypress ,Playwright และPuppeteer การทดสอบแบบทุกส่วนของระบบ End-to-End Testing (E2E) Cypress เป็นวิธีทดสอบที่ครอบคลุมทุกความต้องการ และ Cypress จะรองรับการทดสอบข้ามเบราว์เซอร์ Cross-Browser Testing แต่ Playwright มีคุณสมบัติด้านการรองรับเบราว์เซอร์ที่หลากหลายและการจำลองบนอุปกรณ์พกพาจะมีฟีเจอร์ที่ครอบคลุม แต่มีความซับซ้อนสูง และไม่มี GUI เหมือน Cypress ในขณะที่ Puppeteer รองรับเฉพาะ Chrome-Based Browsers แต่มีความ Flexible ที่ดี และเหมาะสำหรับการสร้างชุดทดสอบที่มีความซับซ้อน

Pelivani และคณะ (2022) แนวคิดงานวิจัยจะประเมินเฟรมเวิร์กของการทดสอบซอฟต์แวร์ และข้อดีและข้อจำกัดของเฟรมเวิร์ก รูปแบบในการทดสอบพิจารณาได้จากความต้องการ Requirements, ความสามารถ Abilities, และความสามารถโดยรวม Capabilities วิธีดำเนินการจะใช้ซอฟต์แวร์ และชุดข้อมูลทดสอบจริงเพื่อให้การทดสอบมีความเป็นจริง โดยเฟรมเวิร์กของการทดสอบคือ Selenium IDE, Cypress, WebDriverIO และ Robot Framework การทดสอบซอฟต์แวร์มีความสำคัญ เพื่อให้การทำงานของซอฟต์แวร์ไม่มีข้อผิดพลาดและทำงานได้อย่างมีประสิทธิภาพ โดยแต่ละเฟรมเวิร์กมีคุณสมบัติ หรือฟีเจอร์ที่มีในเฟรมเวิร์กในรูปแบบที่แตกต่างกัน และสรุปได้ว่า เครื่องมือที่ใช้ในการทดสอบมีประสิทธิภาพในการใช้งานสำหรับการทดสอบ แตกต่างกันโดยมีปัจจัยที่เกี่ยวข้องเป็นตัวกำหนด ทำให้หากเลือกใช้เครื่องมือที่เหมาะสมจะมีประสิทธิภาพที่มากกว่า ดังนั้นเครื่องมือการทดสอบ มีวัตถุประสงค์ของการทดสอบเป็นข้อกำหนด เพื่อให้การเลือกใช้เครื่องมือเหมาะสมในการใช้แพลตฟอร์มภาษาที่ใช้พัฒนา งบประมาณ และระดับความซับซ้อนที่กำลังพัฒนาและทดสอบ

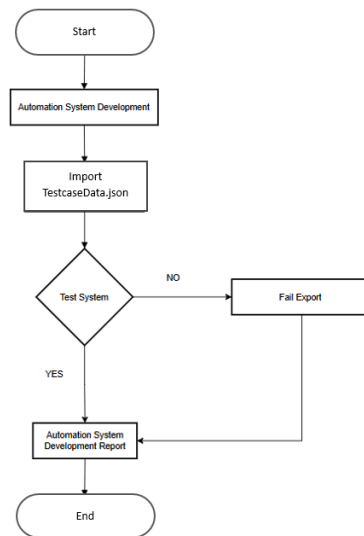
วัตถุประสงค์

1. เพื่อออกแบบ และพัฒนารูปแบบการทดสอบแบบอัตโนมัติสำหรับเว็บแอปพลิเคชัน
2. เพื่อเปรียบเทียบประสิทธิภาพการทดลองของเครื่องมือทดสอบซอฟต์แวร์แบบอัตโนมัติ คือ Selenium, Cypress และ Playwright
3. เพื่อลดระยะเวลาในการคีย์ข้อมูลที่ใช้ในการทดสอบระบบเว็บแอปพลิเคชัน

วิธีดำเนินการวิจัย

งานวิจัยมีการทดสอบ Web Application เริ่มต้นจากการทดสอบในระดับ End to End Testing ซึ่งเป็นการทดสอบการทำงานของระบบ (Leach, 2016) เช่น การทดสอบฟังก์ชันเพื่อให้แน่ใจว่าการทำงานของแต่ละหน่วยมีความถูกต้อง โดยการทดสอบใช้เทคนิค Black-Box Testing ซึ่งเป็นการทดสอบจากมุมมองของผู้ใช้ที่ไม่จำเป็นต้องทราบโครงสร้างภายในของระบบ เพื่อเน้นการตรวจสอบผลลัพธ์ที่ได้จากการป้อนข้อมูลเข้าไป และในการทดสอบ Functional การทดสอบการทำงานของระบบ และ Performance Testing ซึ่งมีวัตถุประสงค์ทดสอบประสิทธิภาพ ระยะเวลาของซอฟต์แวร์ทดสอบ, จำนวนความสำเร็จของซอฟต์แวร์ทดสอบ, ร้อยละ

จำนวนความสำเร็จของซอฟต์แวร์ทดสอบ, ประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลา และการทำ Regression Testing จะถูกนำมาใช้เพื่อให้แน่ใจว่าฟังก์ชันเดิมที่สามารถทำงานได้อย่างถูกต้อง ยังคงสามารถทำงานได้อย่างถูกต้องเช่นเดิมหลังจากการแก้ไขหรือเปลี่ยนแปลงระบบเพื่อป้องกันไม่ให้เกิดข้อผิดพลาดที่อาจจะกระทบต่อฟังก์ชันที่มีอยู่เดิม (Ahmed & Prasad, 2016) แสดงในภาพที่ 1



ภาพที่ 1 กระบวนการทำงานของระบบการทดสอบอัตโนมัติ

เครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนา

การทดสอบซอฟต์แวร์อัตโนมัติ เครื่องมือเทคโนโลยี ภาษาการเขียนโปรแกรม เฟรมเวิร์ก และรวมถึงฐานข้อมูลเพื่อให้สามารถตอบสนองต่อความต้องการของระบบ ซอฟต์แวร์ทดสอบอัตโนมัติ

1. เครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ 3 ซอฟต์แวร์ คือ Selenium, Cypress และ Playwright
2. ภาษาโปรแกรมที่ใช้ในการพัฒนาแอปพลิเคชัน ได้แก่ ภาษา Python โดยใช้เฟรมเวิร์ก Flask สำหรับพัฒนาส่วนประมวลผลฝั่งเซิร์ฟเวอร์ เพื่อระบบและการประมวลผลของข้อมูล ส่วนการแสดงผลต่อผู้ใช้ การพัฒนาโดยใช้ Cascading Style Sheets (CSS) และ JavaScript เพื่อรูปแบบการแสดงผลและการโต้ตอบของระบบ โดยใช้ Visual Studio Code เป็นเครื่องมือพัฒนาสำหรับการเขียนโปรแกรมและจัดการโครงสร้างของระบบ
3. ฐานข้อมูลที่ใช้ในการจัดเก็บและบริหารจัดการข้อมูลในระบบ คือ PostgreSQL

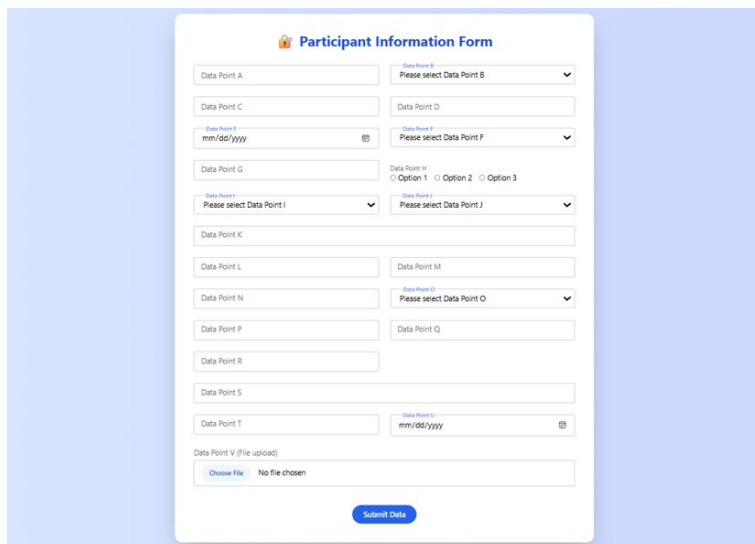
ขั้นตอนการดำเนินการวิจัย

การเปรียบเทียบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ รูปแบบกระบวนการของระบบ ดังนี้

1. ออกแบบและพัฒนาเว็บแอปพลิเคชันเพื่อใช้ในการทดสอบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ
2. ออกแบบและสร้างเทสต์สคริปต์สำหรับการทดสอบของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ
3. ทำการทดสอบเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ เพื่อเปรียบเทียบประสิทธิภาพในด้านระยะเวลาของซอฟต์แวร์ทดสอบ, จำนวนความสำเร็จของซอฟต์แวร์ทดสอบ, ร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบ, ประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลา
4. ทำการเปรียบเทียบประสิทธิภาพของซอฟต์แวร์อัตโนมัติ และเก็บบันทึกผลการทดสอบ
5. ทำ Regression Testing เพื่อให้ได้ผลลัพธ์ตามความต้องการ โดยเครื่องมือการทดสอบแบบอัตโนมัติจำนวน 3 ระบบ Selenium, Cypress, Playwright
6. สรุปผลและประเมินผลการทดสอบเปรียบเทียบข้อมูลที่ใช้ในการทดสอบเพื่อให้เป็นไปตามผลการทดสอบ

การทดสอบของซอฟต์แวร์

การทดสอบจะใช้ซอฟต์แวร์ (Brahmbhatt, 2023) การทดสอบ Selenium, Cypress และ Playwright ในการทดสอบโดยมี Input ที่มีรูปแบบ Data Point มีลักษณะ Text Input ข้อความ, Dropdown Select เลือกค่าจากรายการที่กำหนด, Date Input เลือกวัน/เดือน/ปี, Radio Buttons เลือกตัวเลือกจากรายการที่กำหนด และ File Upload type="file" เพื่ออัปโหลดไฟล์ที่ต้องการแสดงภาพที่ 2

A screenshot of a web form titled "Participant Information Form". The form contains several input fields and dropdown menus. Fields include "Data Point A", "Data Point C", "Data Point G", "Data Point I", "Data Point K", "Data Point L", "Data Point N", "Data Point P", "Data Point R", "Data Point S", "Data Point T", and "Data Point V (File upload)". Dropdown menus are labeled "Please select Data Point B", "Please select Data Point D", "Please select Data Point F", "Please select Data Point J", and "Please select Data Point O". There are also radio buttons for "Option 1", "Option 2", and "Option 3". A date input field is labeled "mm/dd/yyyy". A file upload section at the bottom has a "Choose file" button and a "No file chosen" text. A "Submit Data" button is at the bottom right.

ภาพที่ 2 การออกแบบ และพัฒนาเพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์

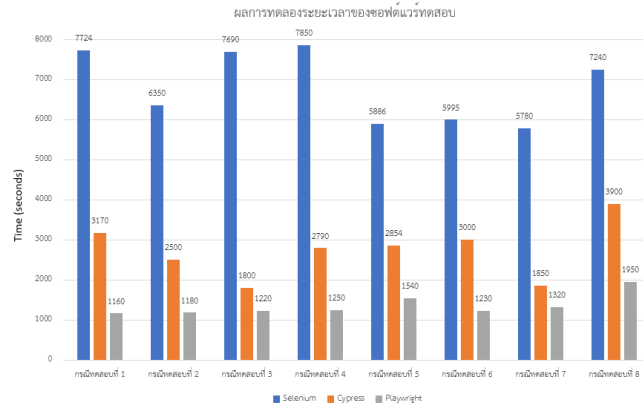
ชุดข้อมูลของการทดสอบซอฟต์แวร์

ข้อมูลที่ใช้ในการทดสอบ เป็นข้อมูลที่ทำกรสุ่มจากการทดสอบที่มีการทดสอบข้อมูลจริง โดยข้อมูลทั้งหมดทำการสุ่มเพื่อปกปิดข้อมูลส่วนบุคคล ในกรณีทดสอบ ทำการสุ่มข้อมูลจำนวน 1,000 รายการมาทำการทดสอบในรูปแบบการทดสอบ โดยข้อมูลทั้งหมดจะเก็บเป็นรูปแบบ .json โดยจะนำข้อมูลทั้งหมดไปทดสอบในระบบอัตโนมัติทั้ง 3 ระบบ

ผลการวิจัย

ในการดำเนินการวิจัย การทดสอบ และการเปรียบเทียบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ โดยมีซอฟต์แวร์ในการทดสอบ Selenium, Cypress และ Playwright เพื่อเปรียบเทียบซอฟต์แวร์ให้เป็นไปตามความต้องการความเหมาะสม เพื่อลดเวลาการทดสอบซอฟต์แวร์อัตโนมัติ ผู้วิจัยได้ดำเนินการพัฒนารูปแบบสำหรับทดสอบ และทำการทดสอบจากนั้นเก็บข้อมูล

การทดสอบระยะเวลาของซอฟต์แวร์ทดสอบ ผลลัพธ์การทดสอบ Playwright มีระยะเวลาในการทดสอบที่น้อยในทุกกรณีทดสอบ ส่วน Cypress ที่ใช้ระยะเวลาในการทดสอบมากกว่า Playwright และ Selenium ใช้ระยะเวลาในการทดสอบมากกว่า Playwright และ Cypress จากภาพที่ 3 ซึ่งในการทดสอบระยะเวลาของซอฟต์แวร์ทดสอบอัตโนมัติ ซอฟต์แวร์ Playwright สามารถใช้ระยะเวลาของการทดสอบได้น้อยที่สุด

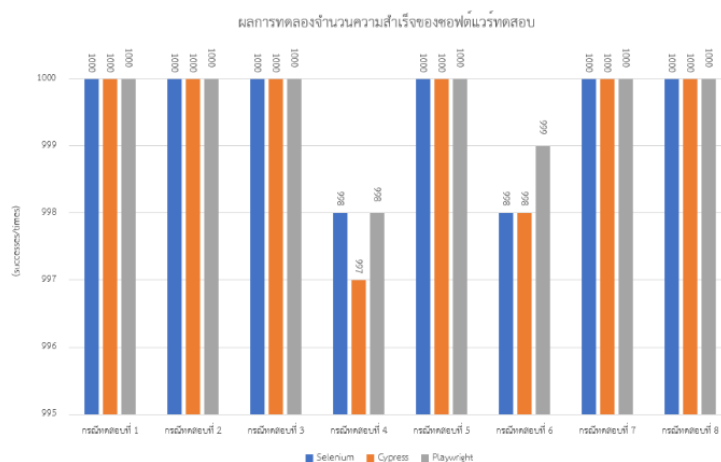


ภาพที่ 3 ผลการทดลองระยะเวลาของซอฟต์แวร์ทดสอบ

การทดสอบจำนวนความสำเร็จของซอฟต์แวร์ทดสอบ ผลลัพธ์การทดสอบจะสังเกตได้ว่ากรณีทดสอบที่ 4 ซอฟต์แวร์การทดสอบ Selenium, Cypress, Playwright สามารถดึงข้อมูลจำนวน 998, 997 และ 998 รายการตามลำดับ โดย Selenium ดึงข้อมูลได้จำนวน 998 รายการ Cypress ดึงข้อมูลได้จำนวน 997 รายการ และ Playwright ดึงข้อมูลได้จำนวน 998 รายการ

ส่วนกรณีทดสอบที่ 6 ซอฟต์แวร์การทดสอบ Selenium, Cypress, Playwright สามารถดึงข้อมูลจำนวน 998, 998 และ 999 รายการตามลำดับ โดย Selenium ดึงข้อมูลได้จำนวน 998 รายการ Cypress ดึงข้อมูลได้จำนวน 998 รายการ และ Playwright ดึงข้อมูลได้จำนวน 999 รายการ ดังนั้นจำนวนความสำเร็จของซอฟต์แวร์ของซอฟต์แวร์ทดสอบ Playwright สามารถดึงข้อมูลได้จำนวนความสำเร็จของการทดสอบมากที่สุด

ผลลัพธ์การทดสอบ Playwright มีจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากที่สุด ส่วน Selenium มีจำนวนความสำเร็จของซอฟต์แวร์น้อยกว่า Playwright และ Cypress มีจำนวนความสำเร็จของซอฟต์แวร์น้อยกว่า Playwright และ Selenium ดังนั้นในการทดสอบจำนวนความสำเร็จของซอฟต์แวร์ทดสอบอัตโนมัติ ซอฟต์แวร์ Playwright มีจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากที่สุด แสดงในภาพที่ 4



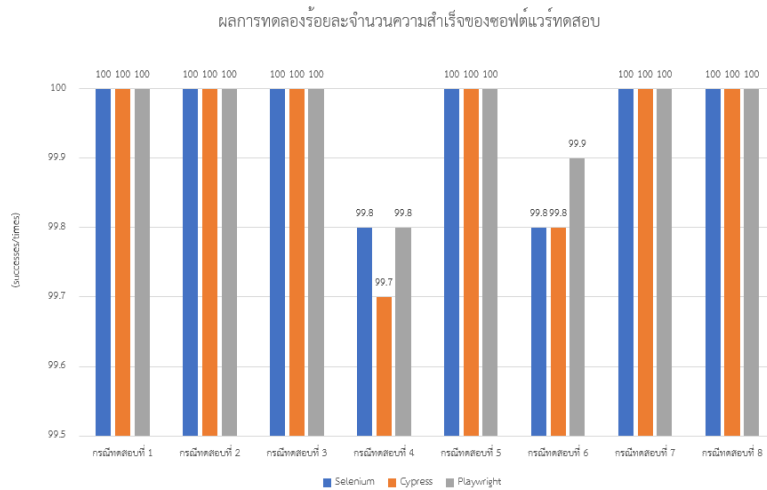
ภาพที่ 4 ผลการทดลองจำนวนความสำเร็จของซอฟต์แวร์ทดสอบ

การทดสอบจำนวนความสำเร็จของซอฟต์แวร์ทดสอบ ผลลัพธ์การทดสอบจะสังเกตได้ว่ากรณีทดสอบที่ 4 ซอฟต์แวร์การทดสอบ Selenium, Cypress, Playwright มีร้อยละความสำเร็จ 99.8, 99.7 และ 99.8 รายการตามลำดับ โดย Selenium มีร้อยละความสำเร็จ 99.8 รายการ Cypress มีร้อยละความสำเร็จ 99.7 รายการ และ Playwright มีร้อยละความสำเร็จ 99.8 รายการ

ส่วนกรณีทดสอบที่ 6 ซอฟต์แวร์การทดสอบ Selenium, Cypress, Playwright มีร้อยละความสำเร็จ 99.8, 99.8 และ 99.9 รายการตามลำดับ โดย Selenium มีร้อยละความสำเร็จ 99.8 Cypress มีร้อยละความสำเร็จ 99.8 และ Playwright มีร้อยละ

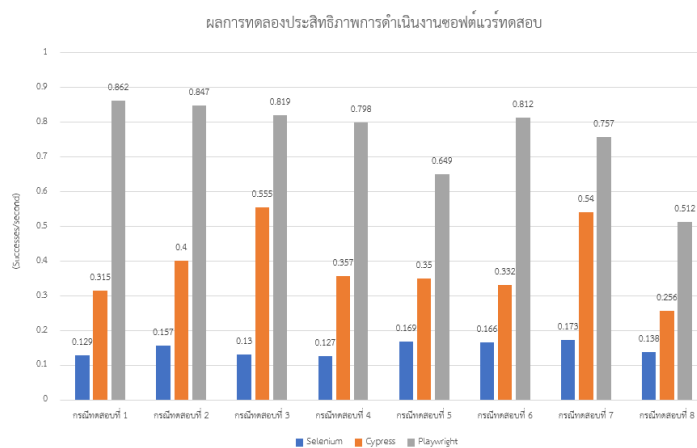
ความสำเร็จ 99.9 รายการ ดังนั้นจำนวนความสำเร็จของซอฟต์แวร์ Playwright มีร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากที่สุด

ผลลัพธ์การทดสอบ Playwright มีร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากที่สุด ส่วน Selenium มีร้อยละจำนวนความสำเร็จซอฟต์แวร์น้อยกว่า Playwright และ Cypress มีร้อยละจำนวนความสำเร็จ น้อยกว่า Playwright และ Selenium ดังนั้นในการทดสอบร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบอัตโนมัติ ซอฟต์แวร์ Playwright มีร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากที่สุด แสดงในภาพที่ 5



ภาพที่ 5 ผลการทดลองร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบ

การทดสอบประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลา ผลลัพธ์การทดสอบ Playwright มีประสิทธิภาพการดำเนินงานของซอฟต์แวร์ในทุกกรณีทดสอบมากที่สุด ส่วน Cypress ประสิทธิภาพการดำเนินงานของซอฟต์แวร์น้อยกว่า Playwright และ Selenium มีประสิทธิภาพการดำเนินงานของซอฟต์แวร์น้อยกว่า Playwright และ Cypress ดังนั้นประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลา ซอฟต์แวร์ Playwright มีประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลาของการทดสอบได้มากที่สุด แสดงในภาพที่ 6



ภาพที่ 6 ผลการทดลองประสิทธิภาพการดำเนินงานซอฟต์แวร์ทดสอบ

สรุปผลการวิจัย

การวิจัยการเปรียบเทียบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติเป็นไปตามวัตถุประสงค์การออกแบบและพัฒนาแบบการทดสอบแบบอัตโนมัติสำหรับเว็บแอปพลิเคชัน โดยมีรูปแบบสำหรับการทดสอบการเชื่อมต่อข้อมูลที่หลากหลายและสามารถนำมาทดสอบซอฟต์แวร์ทดสอบอัตโนมัติได้ตามที่กำหนด ในด้านประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ ดำเนินการทดสอบใช้ซอฟต์แวร์ ได้แก่ Selenium, Cypress และ Playwright โดยกรณีทดสอบจำนวน 8 กรณีทดสอบ จากผลการเปรียบเทียบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ Playwright มีด้านประสิทธิภาพที่ดีกว่า จากการทดลองการเปรียบเทียบประสิทธิภาพของเครื่องมือทดสอบซอฟต์แวร์อัตโนมัติ และในการใช้งานในการทดสอบระบบ จะต้องมีการพิจารณาในด้านปัจจัยที่เกี่ยวข้องเพื่อประกอบการเลือกซอฟต์แวร์ทดสอบอัตโนมัติ และลดระยะเวลาในการเชื่อมต่อข้อมูลที่ใช้ในการทดสอบระบบเว็บแอปพลิเคชัน ด้วยการทดสอบชุดข้อมูลซ้ำหลายรอบการใช้เครื่องมือทดสอบอัตโนมัติในการทดสอบ และช่วยให้สามารถลดระยะเวลาการทดสอบของระบบได้อย่างมีประสิทธิภาพ

การอภิปรายผล

ผู้วิจัยทำการทดสอบจากกรณีทดสอบ 1000 กรณีทดสอบ และทำการเปรียบเทียบ เก็บบันทึกผลการทดลอง โดยใช้ข้อมูล 8 รายการในการทดสอบใช้ชุดข้อมูลที่สุ่มที่มีความใกล้เคียง และทำการปกปิดข้อมูล เพื่อให้การทดลองมีประสิทธิภาพ พบว่า Playwright ระยะเวลาของซอฟต์แวร์ทดสอบน้อยกว่า มีจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากกว่า มีร้อยละจำนวนความสำเร็จของซอฟต์แวร์ทดสอบมากกว่า และมีประสิทธิภาพการดำเนินงานของซอฟต์แวร์ต่อหน่วยเวลามากกว่า ในการทดลองถึงแม้ว่า Playwright จะใช้เวลาน้อยกว่า แต่มีข้อพิจารณาอยู่หลายด้าน หากมีการทดสอบที่ใช้ภาษาโปรแกรม เช่น Java, C# และ Python มีความต้องการให้เข้ากับระบบเดิม Selenium เป็นตัวเลือกที่มีความเหมาะสม ยังมี Community ที่มากทำให้เป็นตัวเลือกที่ดี และรองรับเบราว์เซอร์ได้มากกว่า แต่หากมีการเน้นการพัฒนา Front-End ด้วย JavaScript หรือ TypeScript และต้องการซอฟต์แวร์ที่ง่ายต่อการค้นหาข้อผิดพลาด Cypress จะตอบโจทย์ เพราะมีการใช้ GUI (Graphic User Interface) ของทางซอฟต์แวร์ทำให้สามารถแสดงผลการรันเทสต์ของแต่ละคำสั่งได้อย่างละเอียด Cypress ยังใช้รูปแบบสคริปต์ที่เขียนด้วย JavaScript หรือ TypeScript ได้โดยตรง ซึ่งเหมาะสมกับทีมที่พัฒนา Front-End ทำให้มีความสอดคล้องของภาษาในการเขียนเทสต์เพื่อทดสอบระบบ ดังนั้นความเหมาะสมของซอฟต์แวร์ทดสอบอัตโนมัติปัจจัยที่เกี่ยวข้อง รูปแบบของการทดสอบ, โครงสร้างและเทคโนโลยีที่ใช้ในการพัฒนา, ภาษาโปรแกรมที่ใช้ในการพัฒนา และโครงสร้างพื้นฐานของระบบ

ข้อเสนอแนะสำหรับการวิจัยครั้งต่อไป

ศึกษาซอฟต์แวร์ทดสอบระบบอัตโนมัติในรูปแบบ หรือซอฟต์แวร์ใหม่ ๆ อาจจะมีฟีเจอร์และเทคโนโลยีเข้าช่วย เพื่อให้การทดสอบทำได้ง่ายขึ้น และเป็นการเพิ่มประสิทธิภาพ ทำให้การทดสอบซอฟต์แวร์มีระยะเวลาในการทดสอบที่น้อยลง และได้ผลการทดสอบ ความถูกต้อง และความสำเร็จที่มีประสิทธิภาพ

เอกสารอ้างอิง

- ชานน ลิ้มวารรัตน์. (2566). *การพัฒนาเครื่องมือทดสอบการเป็นไปตามข้อแนะนำของการเข้าถึงเนื้อหาของเว็บไซต์* [สารนิพนธ์ปริญญาโทบริหารธุรกิจ ภาควิชาเทคโนโลยีสารสนเทศ]. มหาวิทยาลัยธุรกิจบัณฑิตย์.
- ธวัชศักดิ์ บุณณะถาวร. (2566). *การพัฒนาการทดสอบซอฟต์แวร์อัตโนมัติสำหรับเว็บแอปพลิเคชันกรณีศึกษาการนำเข้าสู่อุปกรณ์และโอนย้ายอุปกรณ์* [สารนิพนธ์ปริญญาโทบริหารธุรกิจ ภาควิชาเทคโนโลยีสารสนเทศ]. มหาวิทยาลัยธุรกิจบัณฑิตย์.
- เศรษฐพงษ์ อัมสุวรรณ์. (2561). *ศึกษาการใช้ระบบอัตโนมัติทดสอบซอฟต์แวร์เว็บแอปพลิเคชัน: กรณีศึกษาบริษัทด้านเทคโนโลยีสารสนเทศและการสื่อสาร* [สารนิพนธ์ปริญญาโทบริหารธุรกิจ ภาควิชาเทคโนโลยีสารสนเทศ]. มหาวิทยาลัยธุรกิจบัณฑิตย์.
- Ahmed, A., & Prasad, B. (2016). *Foundations of software engineering*. CRC Press.
- Apriansah, M. R., & Desanti, R. I. (2024). Efficiency comparison of employee management automation using Selenium and Playwright: A case study of technology consulting company. In *Proceedings of the 2024 International Conference on Information Technology Systems and Innovation* (pp. 419-425). IEEE. <https://doi.org/10.1109/ICITSI65188.2024.10929217>
- Brahmbhatt, K. H. (2023). *Comparative analysis of selecting a test automation framework for an e-commerce website*. Tallinn University of Technology.

- Donvir, A. (2024). A comparative analysis of JavaScript unit and end-to-end testing frameworks: Enhancing quality assurance in modern web applications. In *Proceedings of the 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks* (pp. 1296–1289). IEEE. <https://doi.org/10.1109/CICN63059.2024.10847365>
- Leach, R. J. (2016). *Introduction to software engineering* (2nd ed.). CRC Press.
- Pelivani, E., Besimi, A., & Cico, B. (2022). A comparative study of UI testing framework. In *Proceedings of the 2022 11th Mediterranean Conference on Embedded Computing* (pp. 1-5). IEEE. <https://doi.org/10.1109/MECO55406.2022.9797165>