

ТЕХНОЛОГІЇ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА

Ми живемо в епоху розвитку технологій. Постійно з'являється нове програмне забезпечення (ПЗ), яке націлене на покращення життя корис-тувачів по всьому світу. Під час створення ПЗ, продукт проходить різні етапи життєвого циклу. Одне з основних етапів – тестування (40-50% затрат на розробку ПЗ відведено на тестування).

Тестування ПЗ - це процес перевірки програмного продукту, на меті якого перевірити відповідність реального продукту до очікуваного. За ступенем автоматизації є два види тестування: ручне тестування (manual testing) та автоматизоване тестування (automated testing). Автоматизоване тестування використовує спеціальне програмне забезпечення (крім тестованого) для порівняння очікуваного результату з фактичним. Цей тип тестування допомагає автоматизувати часто повторювані, але необхідні для максимізації тестового покриття завдання. Автоматизація тестування графічного інтерфейсу користувача (Graphical user interface testing) - спеціальна програма (фреймворк), який дозволяє генерувати дії користувача - натискання клавіш, кліки мишкою, і відслідковувати реакцію програми на ці дії - чи відповідає вона специфікації.

При виборі фреймворку потрібно звертати увагу на його сумісність з технологіями, що використовуються у системі, на його популярність, підтримку та оновлення, а також на особливості кожного інструменту. Розглянемо фреймворки для тестування, що базуються на Selenium - Protractor, Nightwatch.js, Cypress, WebdriverIO.

Protractor - система автоматизації тестування end-to-end з відкритим вихідним кодом, розроблена спеціально для веб-додатків AngularJS.

Плюси:

- простий у встановленні та налаштуванні;
- підтримує паралельне тестування в декількох браузерях;
- забезпечує переваги автоматичної синхронізації;
- має гарну швидкість тестування;

Мінуси:

- немає підтримки мобільного тестування;

Cypress - фреймворк для end-to-end тестування, він працює з власним браузером і впроваджує тести в код сторінок.

Плюси:

- має детальну та зрозумілу документацію;
- зрозумілий інтерфейс для запуску, перегляду і налагодження;
- доступні корисні утиліти для спрощення написання тестів;

Мінуси:

- через те, що запускає власний браузер, робота з цим браузером не завжди стабільна;
- немає підтримки мобільного тестування;
- паралелізація тестів не безкоштовна;

WebdriverIO - це бібліотека JavaScript для управління браузерами з Selenium API WebDriver.

Плюси:

- підтримка синхронного коду;
- зручне налаштування за допомогою налаштованого інтерфейсу командного рядка wdio;
- підтримка бібліотеки "webdrivercss", для порівняння CSS-стилів елементів на сторінці.

Мінуси:

- погано адаптований для автоматизації AngularJS, ніж Protractor.

Nightwatch.js - це бібліотека для написання та запуску автотестів, написане на Node.js. Він використовує W3C WebDriver API для керування браузерами.

Плюси:

- схожий на WebdriverIO;
- простий та зрозумілий синтаксис;
- паралельний запуск тестів.

Мінуси:

- немає підтримки мобільного тестування
- менше підтримки ніж у WebdriverIO і Protractor

Вибір інструмента є однією з найбільших проблем, які потрібно вирішити, перед тим як перейти до автоматизації. По-перше, потрібно визначити вимоги до продукту та які технології використовуються на проєкті. По-друге, для того щоб перейти з ручного тестування на автоматизовані тести, потрібно мати базові знання у програмуванні. Тому, правильний вибір інструменту автоматизації залежить від продукту, що буде тестуватись, від знань та навичок команди, а також від завдань, які потрібно вирішити.