

І. А. Олійник
В. П. Майданюк
Л. Г. Коваль

РОЗРОБКА ФРЕЙМВОРКУ ДЛЯ ТЕСТУВАННЯ МОБІЛЬНОГО ДОДАТКУ JETIQ

Вінницький національний технічний університет

Анотація

Фреймворк для тестування мобільного додатку JetIQ розроблено на основі BDD-методології. BDD (Behavior-driven development) – це технологія, заснована на описі поведінки з використанням синтаксису мови Gherkin.

Ключові слова: BDD-методологія, Gherkin, фреймворк, мобільний додаток, тестування.

Abstract

The JetIQ mobile application testing framework is based on the BDD methodology. BDD (Behavior-driven development) is a technology based on describing behavior using Gherkin syntax.

Keywords: BDD-methodology, Gherkin, framework, mobile application, testing.

Вступ

На сьогоднішній день головною метою людства є налаштування процесів, що необхідні для життєдіяльності на рівень автоматичного виконання. Нажаль ця утопічна ціль далека від впровадження у реальності. Проте і на даний момент ми йдемо у напрямку видалення людського фактору із буденних та рутинних процесів. На кшталт онлайн замовлень у закладах, що виключає шанс того що ваше замовлення перетвориться у «зіпсований телефон».

Роль автоматизації тестування

Автоматизоване тестування переймає на себе величезний об'єм роботи у який входять окрім перевірки функціоналу: багаторазові перевірки стандартних функцій таких як пошук або логін, постійна перевірка програмного забезпечення та внесених змін, виконання регресійного тестування що в собі містить перевірку чималої кількості тестів. І усе це займе величезну кількість часу при виконанні вручну. Таким чином впровадивши автоматизоване тестування ми сильно зменшуємо кількість людей, що відповідають за рутинну роботу, адже їм лише потрібно слідкувати за якістю тестів та оновлювати, що звільняє час для виконання більш важливих задач.

Саме тому розробка фреймворку для тестування є актуальною на сьогодні.

Автоматизацію тестування можна визначити як спосіб повторного запуску набору тестів без необхідності виконувати їх вручну. Введення тестів автоматизації у вашу стратегію тестування – це спосіб заощадити гроші та час.

На даний момент автоматизоване тестування задіяне у чималій кількості проєктів з розробки програмного забезпечення. Проте дуже важко розробити універсальний фреймворк, придатний для використання у різних проєктах. Від проєкту до проєкту підхід до розробки фреймворків для автоматизованого тестування відрізняється.

Процес покращення фреймворків полягає у розробці універсальних методів та високому рівні оптимізації проєкту, що впливає на стабільність та швидкість виконання тестів. Тип наборів тестів, які можна автоматизувати, залежить від множини різноманітного тестування необхідного для програмного забезпечення.

Відомі такі види тестування:

1. Функціональне тестування – написання тестів для перевірки ділової логіки та функціональних можливостей, які очікуються від програми.

2. Нефункціональне тестування – тести для перевірки вимог, що стосуються продуктивності, безпеки, баз даних тощо.
3. Модульне тестування – тести проводяться під час самої фази розробки, в ідеалі розробником після завершення розробки та перед передачею системи тестувальникам на тестування.
4. API тестування – тестування на основі запиту та відповіді, на яких побудована програма.
5. UI тестування – перевіряє функціональність та ділову логіку програми від лица користувача та націлені на тестування функціональних можливостей або просто на тестування елементів інтерфейсу програми.
6. Unit Tests – це тести, які побудовані для тестування коду програми і зазвичай пишуться розробниками. Вони націлені на стандарти кодування, такі як написання методів та функцій[1].

Аналіз особливостей реалізації

Підходів до розробки програмного забезпечення для автоматизації тестування відомо велика кількість. Серед них є різноманітні технології, методології та набори правил.

Для розробки фреймворку для тестування мобільного додатку JetIQ було обрано BDD-методологію. BDD (Behavior-driven development) – це технологія, заснована на описі поведінки з використанням синтаксису мови Gherkin, яка орієнтована на розробку сценаріїв для тестування програмного забезпечення. Ще однією перевагою цієї мови сценаріїв є велика кількість BDD-фреймворків, розроблених на різних мовах програмування для різних платформ, що дозволяє за короткі терміни виконувати розробку різноманітних програмних засобів для тестування програмного забезпечення.

Мова Gherkin містить спеціально виділені ключові слова синтаксису Gherkin, які використовуються для опису специфікації поведінки тестувальної програми: Given – початкова умова, When – подія, яка ініціює даний сценарій, Then – перевірка результату та інші [2]. Далі в залежності від платформи потрібно вибрати BDD-фреймворк. Наприклад, фреймворк JBehave написаний мовою Java, отже тести перетворюються на код Java. Для фреймворку JBehave сценарій мовою Gherkin передається як типовий текстовий файл, який читається порядково.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Види тестувань. [Електронний ресурс] – Режим доступу до ресурсу: https://uk.myservername.com/types-automation-testing#Automation_Tools.
2. Behavior Driven Development [Електронний ресурс] – Режим доступу до ресурсу: <https://scrumtrek.ru/blog/product-management/5615/behavior-driven-development/>.

Висновок

Аналіз підходів до тестування мобільних додатків показав, що BDD (Behavior-driven development) методологія та мова сценаріїв Gherkin є найбільш перспективними для автоматизації тестування мобільних додатків, оскільки розроблено велику кількість BDD-фреймворків різними мовами програмування для різних платформ, що дозволяє за короткі терміни створювати різноманітні програмні засоби для тестування програмного забезпечення. Тому саме на основі цих підходів і розроблено фреймворк для тестування мобільного додатку JetIQ.

Олійник Іван Андрійович – студент групи 2ПІ-186, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: zilper1903@gmail.com.

Майданюк Володимир Павлович — доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця.

Науковий керівник: **Коваль Леонід Григорович** – доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця.