

Бевз Світлана Володимирівна,
кандидат технічних наук, доцент кафедри електричних станцій і систем, Вінницький національний технічний університет,
Бурбело Сергій Михайлович,
кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет,
Войтко Вікторія Володимирівна,
кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет,
Драченко Ярослав Петрович,
студент групи ІІІ-І9м факультету інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет,
Коваленко Олена Олексіївна,
кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ГЕНЕРУВАННЯ ПРОГРАМНИХ ТЕСТІВ НА ОСНОВІ BDD-СПЕЦИФІКАЦІЙ

Розглянуто особливості тестування програмного забезпечення. Запропоновано метод автоматизації процесу тестування з допомогою генерування програмних тестів на основі BDD-специфікацій.

Ключові слова: *тестування програмного забезпечення, автоматизація, генерування тестів.*

This article is related to investigation of software testing process. A new method of automation based on programm tests generation using BDD-specifications is proposed as a solution in QA sphere.

Keywords: *testing process, QA automation, tests generation.*

Вступ. Тестування програмних засобів має важливе значення для забезпечення життєвого циклу програмного додатку [1]. За різних методологій та підходів до організації процесу розробки програмного продукту тестування може займати від 20 до 40 відсотків запланованого часу та потребувати набір кваліфікованих фахівців для підтвердження якості продукту, що тягне за собою залучення значних коштів на покриття витрат як на саме тестування з одного боку, так і на покриття витрат на відшкодування у випадку неякісного продукту з іншого боку [2].

В області тестування програмного забезпечення існує дві основних проблеми:

- 1) валідність програмних тестів відносно вимог до розроблюваного програмного забезпечення (наскільки зазначені тести підтверджують повноцінну і правильну роботу програми);
- 2) швидкість створення системи тестування.

Для забезпечення валідності тестування вимогам до програмного продукту існує підхід до організації розробки, в якому виконувани тести повинні бути прив'язаними до конкретних сценаріїв, написаних зрозумілою для людей мовою, та підтверджувати правильне виконання кожного пункту сценарію. Такий підхід називається BDD - з англ. behavior driven development [3].

Однак цей підхід має декілька недоліків:

- 1) процес тестування не стає швидшим (а іноді є навіть повільнішим);
- 2) розроблювані тести закріплюються за фіксованим текстом сценарію та не можуть бути використані для подібних задач;
- 3) кожен написаний сценарій потребує ручного програмування тестів з боку розробника, що не дає QA-фахівцю простої можливості перевіряти систему за довільними сценаріями.

Саме на вирішення вказаних проблем та недоліків направлений запропонований метод автоматизації процесу тестування шляхом генерування програмних тестів на основі BDD-специфікацій, що є актуальним у процесі створення програмних продуктів.

Аналіз стану питання. Запропонований метод автоматизації процесу тестування з допомогою генерування програмних тестів на основі BDD-специфікацій не має прямих аналогів серед існуючих рішень, оскільки наявні методи орієнтовані на забезпечення гнучкого API саме для програмування тестів, але не для автоматизації цього процесу. Серед них, зокрема:

- 1) «Cucumber» – набір програмних рішень для обробки форматованих мовою Gherkin специфікацій та виконання пов'язаних з нею тестів – має лише функціонал для генерування “заглушок” (порожніх тестів) [4];
- 2) «Mocha» – бібліотека для розробки модульних та інтеграційних тестів мовою JavaScript із підтримкою BDD-підходу – пропонує гнучке рішення для написання тестів, але не пропонує методів автоматизації [5];
- 3) «Serenity» – бібліотека для розробки тестів та формування звітів мовою Java з підтримкою BDD-підходу – пропонує потужний функціонал для звітів, але також не має засобів автоматизації [6].

Запропонований метод автоматизації процесу тестування з допомогою генерування програмних тестів на основі BDD-специфікацій направлений саме

на автоматизацію процесу тестування та реалізований у системі “Marble”. У таблиці 1 наведена порівняльна характеристика аналогів із розроблюваним програмним продуктом.

Таблиця 1 – Порівняльний аналіз з аналогами

Критерій	«Cucumber»	«Mocha»	«Serenity»	«Marble»
Робота з різними мовами програмування	+	-	-	+
Гнучкий інтерфейс розробки тестів	+	+	+	+
Робота з BDD	+	+	+	+
Автоматизація процесу тестування	-	-	-	+

Розробка методу автоматизації процесу тестування з допомогою генерування програмних тестів на основі BDD-специфікацій. Запропонований метод включає в себе вирішення певних задач на різних етапах розробки програмних тестів, зокрема:

1) аналіз BDD-специфікації для отримання чітких сценаріїв та кроків, зрозумілих для автоматизованої системи генерування тестів на етапі проектування – при цьому система одразу може визначити, має вона можливість створити тести під запропоновані сценарії чи ні;

2) генерування та форматування тестів із використанням зазначеної користувачем мови програмування – при цьому система має гнучку специфікацію API для написання модулів генерування, що дає можливість швидко підключати нові мови програмування;

3) багатоваріантна валідація згенерованих тестів – перевірка усіх відомих системі варіантів генерованого коду шляхом їх запуску, при чому користувач може сам вказати критерій валідності тестів.

Сам метод сформульовано в таких кроках:

1) провести семантичний аналіз тексту BDD-специфікацій на знаходження відомих семантичних конструкцій, на основі яких із певною імовірністю можна генерувати тести;

2) для кожної семантичної конструкції визначити належність її до певного регіону тестування програмних продуктів (валідація інтерфейсу користувача, тестування відповідності даних тощо);

- 3) для кожної конструкції у визначеному регіоні обрати найбільш релевантні регіону логічні моделі тестування із відповідними кроками;
- 4) інтерпретувати отримані моделі в синтаксичні конструкції визначеної мови програмування;
- 5) на основі отриманих синтаксичних конструкцій згенерувати код;
- 6) для кожного кроку сценарію BDD-специфікації перевірити на валідність відносно критерію користувача всі варіанти згенерованого коду;
- 7) обрати найбільш релевантні варіанти.

Такий метод дозволяє створити достатньо валідні тести на основі самих лише сценаріїв у текстовому вигляді, що дозволяє пришвидшити процес тестування.

Висновок.

Запропонований метод автоматизації процесу тестування з допомогою генерування програмних тестів на основі BDD-специфікацій реалізований у системі “Marble”. Метод дозволяє автоматизувати процес розробки та валідації програмних тестів шляхом їх генерування на основі аналізу BDD-специфікацій. Метод вирішує проблему пришвидшення процесу розробки програмних тестів та покращує сам BDD-підхід.

Список використаної літератури

1. Що це таке тестування програмного забезпечення. [Електронний ресурс]. – Режим доступу: <https://www.quality-assurance-group.com/shho-take-testuvannya-programnogo-zabezpechennya-ta-yake-jogo-znachennya>. – Назва з екрану.
2. Забезпечення якості. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Забезпечення_якості. - Назва з екрану.
3. Керована поведінкою розробка. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Керована_поведінкою_розробка. – Назва з екрану.
4. Cucumber (software) [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Cucumber_\(software\)](https://en.wikipedia.org/wiki/Cucumber_(software)). – Назва з екрану.
5. Mocha (JavaScript framework) [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Mocha_\(JavaScript_framework\)](https://en.wikipedia.org/wiki/Mocha_(JavaScript_framework)) – Назва з екрану.
6. The Serenity Reference Manual [Електронний ресурс] – Режим доступу: <http://thucydides.info/docs/serenity-staging>. – Назва з екрану.