BDD ПІДХІД В АВТОМАТИЗОВАНОМУ ТЕСТУВАННІ

Безсмертний О.П.
Науковий керівник: к.т.н., доц., Голян Н.В.
Харківський національний університет радіоелектроніки

Відсутність достатнього рівня контролю якості у програмному проекті може призвести до значних наслідків. Автоматизоване тестування ввібрало в себе процеси мануального тестування, а також розвивало та оптимізувало власні методи. Автоматизація допомогла зменшити необхідну кількість людських ресурсів при збільшенні потреби у кваліфікації та кількості необхідного часу на розробку тестів.

Behavior Driven Development або ж просто BDD є одним із підходів у тестуванні та розробці програмного забезпеченні в цілому. Це підхід у якому при розробці поведінка функціоналу грає основну роль [1]. Цей підхід є розширеною версією іншої техніки, а саме Test Driven Development або скорочено TDD [2] і є орієнтованим на контролі якості. Для написання BDD сценаріїв використовується читабельний текст написаний мовою домену (domain-specific language DSL) – Gherkin [3].

Найпоширеніша помилка людей, які не беруть участь у тестуванні, полягає в тому, що основним завданням процесу контролю якості є пошук помилок та помилок у розробленому проекті.

З одного боку, це правильне твердження, але лише частково. Завдання процесів забезпечення якості полягає в тому, щоб перевірити, чи протестований проект відповідає вимогам, а потім підготувати та надати документацію, яка це підтверджує. Крім того BDD сценарії, реалізовані відповідно до цього підходу, уже можна називати документацією, наприклад, документація попередньо виконаних тестових сценаріїв. Єдине, що потрібно, це написати сценарії BDD, дотримуючись усіх найкращих практик, та додати деякі метадані для доповнення сценаріїв поведінки:

- посилання на історії користувача;
- посилання на особливості;
- посилання на дефекти;
- номери версій;
- унікальні ідентифікатори;
- посилання на документацію розробників (наприклад, проектна документація низького рівня);
- посилання на вимоги (наприклад, проектна документація вимог або специфікація вимог користувача);
- посилання на ризики.

BDD вбирає в себе особливості своїх аналогів, та поєднує їх у достатньо гармонічну комбінацію. Таким чином виходить підхід де тестування грає важливу роль на проекті, при цьому люди, котрі розробляють специфікації мають достатню експертизу у цільовій сфері бізнесу. Тести написані з використанням BDD підходу є сценаріями, які у свою чергу складаються із назви сценарія та кроків. Назва описує суть тесту, та зазвичай є стислим переказом кроків. Кроки, у свою чергу описують послідовність дій, від початку до перевірки результату. Сценарії об’єднуються певною історієї, яка має відповідну коротку назву, та певну історію користувача, яка їй відповідає.

Щодо мови Gherkin, то вона не дуже відрізняється від звичайної, яку ми використовуємо для спілкування (наприклад, англійської), з тою лише різницею, що має певні ключові слова, які допомагають описувати тестові сценарії. Наведемо список необхідних для розуміння матеріалу ключових слів:

- Feature (Story)/Історія – позначає окрему історію, після ключового слова йде її назва, а з наступного рядка пишеться історія користувача, що її характеризує. Далі йдуть сценарії;

- Scenario/Сценарій – позначає окремий сценарій, після цього слова через пробіл йде його назва, а далі йдуть кроки з нового рядка, кожен крок у окремому рядку і так до нового ключового слова Scenario;
- Given/Дано – ключове слово, що позначає певний тип тестових кроків, що відповідає за умови сценарію;
- When/Коли – ключове слово, що позначає певний тип тестових кроків, що відповідає за дію сценарію;
- Then/Тоді – ключове слово, що позначає певний тип тестових кроків, що відповідає за результат сценарію;
- And/І – заміняє останній попередній тип кроку.

Далі наведено переваги BDD серед аналогів:

- низький поріг входження – це одна із головних переваг принципів BDD. Для написання тестових сценаріїв не потрібні технічні навички, адже сценарії написані читабельною мовою, а значить це може зробити навіть бізнес аналітик. Далі тестувальник або розробник доповнить сценарії реалізацією за допомогою тої чи іншої мови програмування;
- простота генерації тестової документації. Для того, щоб отримати гарну документацію необхідно лише доповнити сценарії деякими мета даними, адже сценарії уже написані у форматі, необхідному для документування;
- зрозумілість тестів для не залучених до проекту осіб. Ця перевага більш характерна для автоматизованих тестів, де зазвичай важко зрозуміти тестові сценарії комусь окрім команди розробки.

Усі переваги так чи інакше походять від застосування мови домену для написання тестів. Ця мова ідеально підходить для бізнес людей, а з іншого боку достатньо зрозуміла і зручна у роботі для технічних експертів. Це і є причиною того чому BDD підхід є ідеальним поєднанням технічних та бізнес аспектів процесів тестування.

Список літератури

1. Нагі Г. Discovery: Explore behaviour using examples (BDD Books) / Г. Нагі, С. Роуз., 2018. – 99 с.
2. Ленка К. Behavior Driven Development: Tools and Challenges / К. Ленка, С. Кумар, С. Мамейн. // 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). – 2018.
3. Вінн М. The Cucumber Book: Behaviour-Driven Development for Testers and Developers / М. Вінн, А. Геллесой., 2012. – 336 с.