

АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ НА МОВІ ПРОГРАМУВАННЯ PYTHON.

Вінницький національний технічний університет

Анотація

У даній роботі приділено увагу аналізу інструментів для автоматизації тестування на мові програмування Python.

Ключові слова: автоматизоване тестування, фреймворк, PyTest, Behave, CLI

Abstract

This article considered on analyzing Python programming automation tools.

Keywords: automated testing, framework, PyTest, Behave, CLI

Вступ

Автоматизоване тестування відіграє значну роль в процесі розробки програмного забезпечення з великим об'ємом функціоналу. Існує багато фреймворків та інструментів для розробки якісних тестів, кожен з яких має сильні і слабкі сторони[1].

Можна виділити два основних фреймворки для автоматизованого тестування на мові програмування Python, а саме Pytest та Behave.

Порівняємо дані фреймворки та визначимо основні особливості, переваги та недоліки.

Аналіз інструментів для автоматизації тестування на мові програмування Python

Pytest - це фреймворк для написання тестів на мові програмування Python. Він сумісний зі специфікацією QTI IMS Global Learning, Inc. та може використовуватися як редактор тестів або як сервер для створення мережевої структури тестування.

Фреймворк pytest дозволяє легко розробляти автоматизовані тести і масштабується для підтримки складного функціонального тестування додатків і бібліотек.

Pytest має багато ключових особливостей, з них можна виділити найголовніші, такі як автоматичне виявлення тестових модулів і функцій, ефективний CLI для поліпшення контролю над тим, що потрібно запустити або пропустити, велика стороння екосистема плагінів та можливість паралелізації. Однією з переваг є фікстури - різні типи та області застосування для підготовки оточення з заздалегідь фіксованим/відомим станом для гарантії повторюваності процесу тестування, а також робота з традиційною структурою модульного тестування.

Також серед переваг використання pytest є розширюваність і можливість швидкої зміни структури автоматизованих тестів, можливість фільтрації та можливість перезапуску з останнього невдалого тесту.

Pytest підтримує параметризацію, яка допомагає виконувати тести з "різними конфігураціями". Можна оперативно налаштувати систему звітування, наприклад, підключивши інструмент Allure, ReportPortal або

аналогічний інструмент для їх формування. Також важливим аспектом є детальна і актуальна технічна документація фреймворка.

В цілому знайти серйозні мінуси у даного фреймворка практично неможливо: це перевірений часом підхід, який відмінно себе показує на будь-яких проектах. З відносних недоліків можна назвати необхідність знання Python на високому рівні[2].

Behave - один з популярних BDD фреймворків мови програмування Python. Хоча цей фреймворк не є офіційним продуктом компанії Jenkins, але він підтримується, та оновлюється постійно. Зараз цей фреймворк все більше набирає популярності. Behave може інтегруватися з серверами безперервної інтеграції і може генерувати деталізовані звіти після тестування.

Перевагою Behave є можливість розробляти автоматизовані тести без знання мови програмування Python, але потрібно знати мову gherkin. Дана перевага виступає також недоліком, без умінь занурюватися в код і розуміти Python розробляти автоматизовані сценарії буде неможливо або дуже складно, занадто багато витраченого часу. Однак за допомогою подібного до фікстур з PyTest функціоналу зручно модифікувати поведінку функції, не змінюючи її код.

Одним з недоліків є відсутність можливості використовувати паралелізацію запуску автоматизованих сценаріїв, завдяки якій в багато разів зменшується час проходження сценаріїв. Незважаючи на те, що не все потрібно запускати паралельно, сама можливість це робити критично важлива для автоматизації проектів з великим об'ємом функціоналу.

Висновки

Для автоматизованого тестування на мові Python не існує універсального інструменту, у кожного з інструментів є свої переваги і недоліки, спираючись на які та відповідно до вимог замовників, робиться вибір в сторону того, чи іншого інструмента. Технології, інструменти та фреймворки продовжують розвиватись і пропонують все більший і зручніший функціонал.

На основі фреймворків, які були проаналізовані, можна зробити висновок, що Behave і Pytest на даний момент є найбільш ефективними інструментами, які дозволяють покрити автоматизованими сценаріями майже весь функціонал програмного забезпечення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рекс Блек. Ключові процеси тестування / Блек. Р/ Лори , 2013.–367 с.
2. Сравнение BDD фреймворков Python: плюсы и минусы [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/otus/blog/460046/>, вільний.
3. Канер Сем, Фолк Джек, Нгуєн Енг Кек. Тестування програмного забезпечення. ДіаСофт, 2008.- 544 с.

Марущак Ярослава Віталіївна – студентка групи ІАКІТ-19м, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м.Вінниця, e-mail: yaroslava.marushchak10@gmail.com

Юлія Олександрівна Дрожнікова — студентка групи ІАКІТ-19м, факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, Вінниця, e-mail: julidro1698@gmail.com

Науковий керівник: **Володимир Юрійович Коцюбинський** — к.т.н., доцент, Вінницький національний технічний університет, м. Вінниця

Marushchak Yaroslava V. – Faculty of computer systems and automation, Vinnytsia national technical University, Vinnytsia, e-mail: yaroslava.marushchak10@gmail.com

Drozhnikova Yulia O. — Faculty of computer systems and automation, Vinnytsia national technical University, Vinnytsia, e-mail: julidro1698@gmail.com

Supervisor: **Kotsiubynskyi Volodymyr Y.** — PhD, Associate Professor, Vinnytsia National Technical University, Vinnytsia