

УДК 681.518

А.М. Ковтко; В.Б. Савків, к.т.н., доц.; І.Р. Козбур

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

АВТОМАТИЗОВАНА СИСТЕМИ ГЕНЕРАЦІЇ МОДУЛЬНИХ ТЕСТІВ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ ТА МУТАЦІЙНОГО ТЕСТУВАННЯ

A.M. Kovtko; V.B. Savkiv, Ph.D., Assoc.Prof.; I.R. Kozbur

AUTOMATED SYSTEM FOR GENERATION OF MODULE TESTS BASED ON ARTIFICIAL INTELLIGENCE AND MUTATIONAL TESTING

Тестування програмного забезпечення є обов'язковим компонентом життєвого циклу розробки програмного забезпечення (SDLC – Software development lifecycle), метою якого є забезпечення безвідмовного та правильного функціонування програмного забезпечення, відповідності визначеним вимогам і забезпечення задовільної взаємодії з користувачем. В умовах прогресуючого розвитку програмного забезпечення все більше зростають вимоги до термінів виконання, обсягів і якості його тестування. Деякі з відомих викликів, які ускладнюють процес тестування на етапі сучасного розвитку ПЗ, включають:

- зростання складності програмного забезпечення, яке пов'язане з розвитком технологій та збільшенням вимог до функціональності та масштабованості. Більш складні програми вимагають ретельнішого управління, тестування та підтримки;

- програмний продукт, в умовах сучасного рівня конкуренції в даній сфері, необхідно впроваджувати у стислі терміни, оскільки він швидко може втратити свою привабливість, унікальність та потенційне місце у вибраній ніші. Встановлення балансу між надійним тестуванням і вимогами ринку є ще одним викликом для команди розробників.

Під впливом вищезгаданих викликів тестування ПЗ вимагає подальших трансформацій із застосуванням нових технологій, таких як штучний інтелект (AI) та машинне навчання (ML). Ці технології дозволять ефективно автоматизувати повторювані завдання, покращити виявлення дефектів і оптимізувати процеси тестування. Все більшої популярності набувають технології генеративного штучного інтелекту, що використовують LLM (великі мовні моделі), котрі відповідно змінюють сучасні принципи розробки та тестування програмного забезпечення. Інструменти на основі штучного інтелекту, такі як GitHub Copilot, діють як «розумні помічники» і виходять за рамки простих функцій автозаповнення.

Модульне тестування є критично важливим раннім етапом тестування ПЗ, який гарантує, що окремі компоненти програми функціонуватимуть належним чином. Якісні модульні тести можуть значно зекономити ресурси, виявляючи дефекти на початкових етапах розробки, що в подальшому гарантує загальну якість програмного продукту. Такі інструменти, як GitHub Copilot і йому подібні, надають можливість автоматизувати процес генеративної розробки модульних тестів, тим самим скорочуючи загальний цикл тестування, за рахунок чого у розробників залишається більше часу для інших задач, при цьому потенційно збільшується надійність системи. Наприклад, якщо розробник пише функцію програмного коду, Copilot пропонує набір модульних тестів, які охоплюють різні сценарії введення, крайові випадки та очікувані результати. Це не тільки заощаджує час, але й гарантує ретельнішу перевірку коду, що відповідно зменшує ймовірність потрапляння помилок у кінцевий програмний продукт.

Інструменти генеративного штучного інтелекту пропонують численні переваги та покращення, але вони також мають потенційні проблеми та недоліки. Якість згенерованих тестів може різнитись в залежності від типу модуля який тестується, кількості вже існуючих тестів в проекті та загальної якості коду. Непоодинокі ситуації, коли модульні тести,

згенеровані штучним інтелектом, некоректно перевіряють результати роботи функції чи класу. Потенційним вирішенням цієї проблеми є мутаційне тестування, ідея якого полягає в перевірці ефективності тестів шляхом створення “мутацій” в програмному коді (навмисні помилки або зміни в алгоритмах). Мутаційне тестування допомагає виявити слабкі місця в існуючому наборі тестів, показуючи, чи вони здатні виявити найдрібніші помилки в програмному коді. Таким чином подібна автоматизована система генерації та покращення модульних тестів допомагає виявити помилки на ранніх етапах та покращити результати тестування для уникнення проблем в подальшому.

Ідея створення автоматизованої системи генерації та покращення модульних тестів полягає у використанні переваг як інструментів генеративного штучного інтелекту так і мутаційного тестування для забезпечення високої якості згенерованих тестів. Інтеграція модульного тестування на базі мутацій з використанням генеративного ШІ включає такі основні компоненти:

- генерація тестів на основі ШІ: інструменти штучного інтелекту використовуються для створення різноманітного набору тестів на основі існуючої кодової бази. Ці тести спрямовані на охоплення широкого діапазону сценаріїв, у тому числі тих, які з великою ймовірністю можуть пропустити тестувальники;
- внесення мутацій: кодова база піддається тестуванню на мутації, за рахунок внесення навмисних дефектів програмного коду. Тести, згенеровані штучним інтелектом, запускаються зі зміненими версіями коду;
- аналіз та оптимізація: на основі аналізу результатів мутаційного тестування, штучний інтелект за визначеними критеріями покращуватиме уже існуючі тести;
- цикл зворотнього зв'язку: процес ітераційний, при цьому штучний інтелект постійно вивчає результати мутацій і покращує якість набору тестів. Цей зворотній зв'язок забезпечує зростання якості тестів і є основною ознакою традиційних автоматичних систем.

Очікується, що запропонована інтеграція дасть кілька важливих результатів:

1. Покращення ефективності тесту. Завдяки поєднанню сильних сторін мутаційного тестування та асистентів на основі штучного інтелекту отриманий набір тестів буде ефективнішим у виявленні дефектів, у тому числі різноманітних крайніх випадків.
2. Вивільнення часу. Запропонована система дозволить значно скоротити часові терміни процесу тестування та зменшити навантаження на розробників шляхом автоматичної генерації тестів з додатковою гарантією їх якості у вигляді ітераційного процесу їх покращення за допомогою мутаційного тестування. Таким чином, розробники витратимуть значно менше часу на оцінювання якості згенерованих тестів та їх модифікацію.
3. Універсальність підходу. Очікується, що запропонована автоматизована система зможе працювати з різними наборами інструментів та мов програмування тим самим забезпечуючи універсалізацію процесу тестування.
4. Підвищення якості фінального продукту. Оскільки дана система використовуватиметься на початкових етапах розробки, то вона має потенціал забезпечити раннє виявлення помилок та проблем у розробці, що значно дешевше та ефективніше, ніж на пізніх стадіях процесу розробки.

Література

1. Аналіз ефективності систем автоматизованого тестування програмних продуктів / А.М. Ковтко, Н.В. Лещук, І.Р. Козбур, І.В. Коноваленко // Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 6-7 грудня 2023 року. — Т. : ФОП Паляниця В. А., 2023. — С. 421–422. — (Комп'ютерно-інформаційні технології та системи зв'язку).