

СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ НА ОСНОВІ ФРЕЙМВОРКУ ЮНІТ-ТЕСТУВАННЯ

Вінницький національний технічний університет

Анотація

Визначено основні складові інтелектуальної системи автоматизованого тестування. Здійснено аналіз процесу розробки інтелектуальної системи автоматизованого тестування на основі фреймворку юніт-тестування.

Ключові слова: тестування, юніт-тестування, автоматизоване тестування, інтелектуальна система.

Abstract

The main components of the intelligent system for automated testing are determined. The process of creation of the intelligent system for automated testing on the basis of the unit-testing framework is analyzed.

Keywords: testing, unit-testing, automated testing, intelligent system.

Вступ

Складність програмних продуктів має тенденцію щорічно збільшуватись, що відповідно значно ускладнює їх розробку, можливість внесення змін, збільшує вартість адміністрування та потребує нових, розширеніших та складніших стандартів тестування. Відомо [1], що близько 53% програмних проєктів завершуються з перевитратами, а тільки для економіки США загальні збитки пов'язані із дефектами програмного забезпечення (ПЗ) щорічно складають 59,6 мільярдів доларів [2].

З метою мінімізувати такі витрати в процесі створення програмних продуктів розробники намагаються одночасно застосовувати різні методи тестування, що включають в себе ручне (мануальне) та автоматизоване, наприклад, модульне (юніт-тестування), функціональне та ряд інших.

Проте складність та мультикомпонентність сучасних систем не дозволяють провести вичерпне тестування продукту. Натепер значною популярністю користується модульний підхід у автоматизованому тестуванні, що застосовується багатьма великими та меншими розробниками програмного забезпечення у всьому світі. Він базується на створенні модульних (юніт) тестів, які визначають вимоги до правильності реалізації коду безпосередньо перед його написанням. Цей підхід дає можливість визначати значну кількість помилок на ранніх стадіях розробки, і в подальшому значно прискорити процес регресійного тестування при внесенні змін в існуючий код, оскільки великі обсяги програмного коду тестуються автоматично.

Для створення юніт-тестів використовуються спеціальні програмні бібліотеки (фреймворки), котрі допомагають при організації та реалізації процесу юніт-тестування.

В зв'язку з широким застосуванням, збільшуються вимоги до бібліотек юніт-тестування і їх розробники розширюють фреймворк до рівня, коли така бібліотека стає багатокомпонентною інтелектуальною інформаційною системою.

Мета дослідження – визначення необхідних передумов та специфіки реалізації фреймворку юніт-тестування для створення на його основі інтелектуальної системи автоматизованого тестування.

Об'єктом дослідження є процес автоматизованого тестування ПЗ з використанням інтелектуальних систем.

Предмет дослідження – інтелектуальні системи автоматизованого тестування та їх основні функціональні характеристики.

Результати дослідження

Юніт-тестування або модульне тестування – метод тестування програмного забезпечення, що дозволяє перевірити на правильність окремі модулі вихідного коду програми. Ідея полягає в написанні тестів для кожної функції або методу. Перевагою такого методу є ізоляція окремих частин додатку і доказ, що кожна окрема функція, метод та модуль коректний і здійснює очікувану поведінку [3].

Інтелектуальною інформаційною системою називають комплекс програмних та логіко-математичних засобів для вирішення різних задач в залежності від конкретних інформаційних потреб користувачів [4]. Основною особливістю реалізації та використання такої системи є те, що вона повинна вирішувати поставлену проблему якомога повніше, надаючи користувачам максимальний спектр програмних, інтелектуальних, рекомендаційних та інших інструментів. Основою інтелектуальної системи автоматизованого модульного тестування є програмна бібліотека для створення, запуску, опрацювання та отримання результатів виконання юніт-тестів.

Проте, така система має містити в собі ряд інших компонентів та функцій, що допомагають їй користувачам вирішувати задачу проведення якісного автоматичного тестування проекту. Такі компоненти та функції включають в себе:

Динамічний аналіз нового та модифікованого коду на рівень його покриття юніт-тестами. Один з найкращих рекомендаційних компонентів, що може надати інформаційна система користувачу. Він характеризується постійним аналізом нових та змінених функцій і методів на наявність для них відповідних юніт-тестів. Такий функціонал допомагає розробникам в реальному часі пам'ятати та розуміти, що код який вони пишуть чи змінюють потребує покриття модулями автоматизованого тестування, що в результаті приводить до підтримки максимального рівня відношення наявного коду проекту до відповідних модульних тестів для нього.

Статичний аналіз коду на рівень його покриття юніт-тестами. Відрізняється від динамічного тим, що виконується не постійно, а лише в кінці процесу опрацювання вже написаних модульних тестів, тобто під час кожного запуску інтелектуальної системи автоматизованого тестування. Головною функцією статичного аналізу коду є опрацювання всього програмного проекту на рівень його покриття тестами, тобто в результаті користувач системи має отримати повний список усіх методів, функцій та модулів проекту, які потребують написання або зміни відповідних юніт-тестів.

Розширений список функцій перевірки. Важливим функціоналом, що надає бібліотека юніт-тестування є вбудований набір функцій, що називаються функції перевірки та повертають значення правди (true) або неправди (false) очікуваному результату, що задає користувач. Такі функції можуть відповідати за порівняння (більше, менше, дорівнює), чи входить певне значення в множину, чи є об'єкт узагальненим, перевіряти тривалість часу виконання методів та інші і працювати з різними типами даних. Чим більший список доступних для розробників функцій перевірки, тим більше тест-кейсів та випадків можуть бути покриті та протестовані автоматично.

Збільшення швидкодії опрацювання юніт-тестів. Тривалість виконання модульного тестування залежить від часу пошуку відповідних тестових методів та часу їх виконання. Задача розробників системи максимально мінімізувати час за який повністю виконується процес автоматичного тестування, оскільки це впливає на рівень задоволення користувачів системою. Від специфіки реалізації фреймворку юніт-тестування залежать методи покращення швидкодії. Найкращими методами зменшення тривалості виконання є паралелізація запуску виконуваних тестових методів, застосування найшвидших алгоритмів та найбільш оптимальних функцій технології, що використовується, для пошуку та запуску тестових методів.

Висновки

Таким чином, у ході дослідження визначено, що створення інтелектуальної системи автоматизованого тестування на основі бібліотеки юніт-тестування є актуальною задачею. Проте, така система потребує розробки додаткових функцій та компонентів для того, щоб повністю вирішувати інформаційні потреби користувачів у проведенні якісного автоматичного тестування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Поморова О.В. Сучасні проблеми оцінювання якості програмного забезпечення / О.В. Поморова, Т. О. Говорушенко // Радіоелектронні та комп'ютерні програми : наук.-техн. журнал. – Національний лісотехнічний університет України

294 Серія економічна Харків : Вид-во НАУ ім. М.С. Жуковського "Харківський авіаційний інститут". – 2013. – № 5. – С. 319-327.

2. Software Bugs Cost U.S. Economy \$59.6 Billion Annually, RTI Study Finds [Електронний ресурс]. – Режим доступу: <http://www.nist.gov/director/prog-ofc/report02-3.pdf>
3. Ошеров Р. The art of unit testing with examples in .NET – Гринвіч: Manning Publication Co, 2009. — 20 с.
4. Things to Know About Intelligent Information System [Електронний ресурс] // Режим доступу: <https://www.hitechnectar.com/blogs/intelligent-information-system/>

Василевський Володимир Олегович, студент групи 2KN-20м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: vovavasilevskyi@gmail.com.

Яровий Андрій Анатолійович, доктор технічних наук, професор, завідувач кафедри комп'ютерних наук, Вінницький національний технічний університет, Вінниця, e-mail: a.yarovyy@vntu.edu.ua.

Vasylevskyi Volodymyr, student of group 2KN-20m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: vovavasilevskyi@gmail.com.

Yaroyvi Andrii, Doctor of Science (Eng.), Professor, Head of the Department for Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: a.yarovyy@vntu.edu.ua.