МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ В. Н. КАРАЗІНА

ІТ-ПРОСТІР СЬОГОДЕННЯ: ТЕНДЕНЦІЇ, ІННОВАЦІЇ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

Збірник тез доповідей
Всеукраїнської науково-практичної студентської конференції

(16 жовтня 2024 року, м. Харків, Україна)

Електронний ресурс

Харків – 2024

УДК 004.9

Скорін Ю.І.
к.т.н., доцент, доцент кафедри інформаційних систем
Харківський національний економічний університет імені Семена Кузнеця
Терентьєв О.О.
здобувач вищої освіти,
Харківський національний економічний університет імені Семена Кузнеця

КАТЕГОРИЗАЦІЯ РЕЗУЛЬТАТІВ АВТОМАТИЧНОГО ТЕСТУВАННЯ З ЗАСТОСУВАННЯМ ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ

Сучасним стандартом в розробці програмних продуктів стає процес випуску нових версій з високою частотою [2].

Проміжок між версіями може складати місяць, тиждень, а деколи й один день. Основною мотивацією для таких частих релізів є високе конкурентне середовище.

Комерційному підприємству треба випереджати конкурентів у задовільнені нових запитів від клієнтів. Запізнення може коштувати відтоком клієнтів та наступним за цим зниженням прибутків.

З іншого боку цього процесу знаходиться якість програмного забезпечення.

Неякісний продукт з великою кількістю помилок зменшить кількість клієнтів ще швидше ніж відсутність нового функціоналу.

Існують різноманітні схеми організації перевірки якості програмних продуктів, але кінцевим агентом, котрий приймає рішення стосовно придатності програми виконувати описаний новий функціонал на прийнятному рівні є людина.

Цю роль можуть виконувати розробники, менеджери проєктів, або окремі спеціалісти. Для подальшого розгляду будемо називати людину, котра виконує цю роль тестувальником.

З ростом розміру проєкту потреба в тестувальниках зростає. А також зростає кількість операцій, котрі вони повинні виконати аби дізнатись наявний стан відносно відповідності продукту заявленій якості.

А з ростом частоти випуску, робота по перевірці стає монотонною і це збільшує кількість помилок самих тестувальників.

Тому починаючи з певного розміру проєкту, кількості часу, необхідного на перевірку однієї версії продукту та частоти випусків, компанії починають автоматизування процесу тестування.

Що дуже сильно скорочує час котрий тестувальники витрачають на перевірки [2].

На жаль програмне забезпечення не рідка теж має помилки, котрі приводять до невірного визначення придатності системи.

Також робота систем в тестовому середовищі пов’язана з певними неполадками в фізичному обладнані, котре неможливо або дуже коштовно передбачити в автоматичних тестах.

Це призводить до ситуацій, коли знову тільки тестувальник, переглянувши результати тестів, що завершились з помилками, зможе визначити чи це реальний дефект продукту, чи це проблема з автоматичним тестом, або це проблема тестового середовища.

З ростом функціональності продукту, росте і кількість автоматичних тестів, а з ними й кількість помилок, котрі треба перевіряти тестувальникам.

Одним з варіантів розв'язання цієї проблеми може бути автоматичне визначення джерела проблеми.

Тобто по наявним записам сценарію тестування та відповідям системи й історії попередніх тестувань ми маємо автоматично визначити причину помилки.

Для вирішення такого класу проблем останнім часом набули популярності методи машинного навчання, котрі при певній кількості вхідних даних надають досить високу точність результатів [2].

Хоча методи для вирішення такого класу задач існують достатньо давно, лише в останні роки вони почали демонструвати прийнятну якість [2–4].

Основними факторами стали: загальний розвиток методів машинного навчання, збільшення об’єму даних для навчання, а також збільшення обчислюваної потужності сучасної техніки.

Існує дві основні категорії тестування програмного забезпечення: ручне та автоматизоване.

Ручне тестування займає багато часу, трудомісткість, а зі складним програмним забезпеченням воно також може бути дорогим, якщо ви використовуєте його виключно.

Автоматизоване тестування оптимізує процеси, скорочує час, необхідний для тестування, і усуває неефективність, наприклад розробники програмного забезпечення витрачають виснажливі години на тестування функціональності програмного забезпечення [1].

Автоматизоване тестування це процес використання програмних інструментів, які запускають нещодавно розроблене програмне забезпечення або оновлення через низку тестів для виявлення потенційних помилок кодування, вузьких місць та інших перешкод продуктивності.

Засоби автоматизації тестування програмного забезпечення виконують такі функції [1]:

- впровадження та виконання тестів;
- аналіз результатів;
- порівняння результатів з очікуваними результатами;
- формування звіту про продуктивність програмного забезпечення розробки.

Автоматизовані тести можуть допомогти швидше виявляти збої з меншим ризиком людської помилки. Крім того, їх легше запускати кілька разів для кожної зміни або до досягнення бажаних результатів.

Автоматизація також прискорює процес виведення програмного забезпечення на ринок. Автоматизація дозволяє проводити ретельне тестування в певних областях, щоб ви могли вирішити загальні проблеми, перш ніж переходити до наступного етапу [1].

Піраміда автоматизації тестування допомагає зрозуміти, як часто потрібно виконувати кожен тип тесту.

Піраміда автоматизації тестування поділяє тестування на чотири рівні. Нижній шар представляє тести, які ви повинні виконувати найчастіше.

Рівні стають меншими, чим ближче вони до вершини піраміди, тобто тести, які вам слід виконувати рідше [1].

Ось типи тестів, які слід виконати за пірамідою автоматизації тестування, від найбільшого до найменшого [1]:

- модульні тести;
- інтеграційні тести;
- тести API;
- тести інтерфейсу користувача.

Таким чином мета полягає в дослідженні застосування методів машинного навчання для категоризації результатів автоматичного тестування.

Необхідно провести аналіз наявних рішень. Визначити критерії вибору.

На прикладі обраного рішення провести моделювання, та визначити ефективність зазначеної методики.

Список використаних джерел:

1. Що таке автоматизація тестування? URL: https://www.zaptest.com/uk/%D1%89%D0%BE%D1%82%D0%B0%D0%BA%D0%B5%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D1%96%D1%8F%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%D0%B1%D0%B5%D0%B7
2. Arnon Axelrod. Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects. / Axelrod // Arnon New York, NY, USA: Apress, 2018. – 558 с.
3. Ian Goodfellow. Deep Learning. / Goodfellow Ian, Bengio Yoshua, Courville Aaron // Cambridge, MA, USA: The MIT Press, 2016. – 800 с.
4. Машинне навчання простими словами. URL: http://www.mmf.lnu.edu.ua/ar/1739
5. Розробка програмного забезпечення для розв'язання задачі категоризації текстових документів. URL: https://repository.kpi.kharkov.ua/-server/api/core/bitstreams/d19690b8-9fb7-41bf-ac34-e7016d1fa62b/content