

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ**

**Автоматизація тестування під час проектування мобільних
застосунків за технологією Agile**

Автор:
Павло Микита
студ. гр. КІУКІ-21-5

Керівник:
Наталія БОЛОГОВА
доц. каф.ЕОМ


Мета

Метою роботи є розробка ефективної методики автоматизованого тестування мобільних застосунків, що створюються за технологією Agile.

Об'єктом дослідження є процес тестування мобільних застосунків, розроблених за технологією Agile.

Предметом є методика автоматизації тестування таких застосунків у межах Agile-підходу.

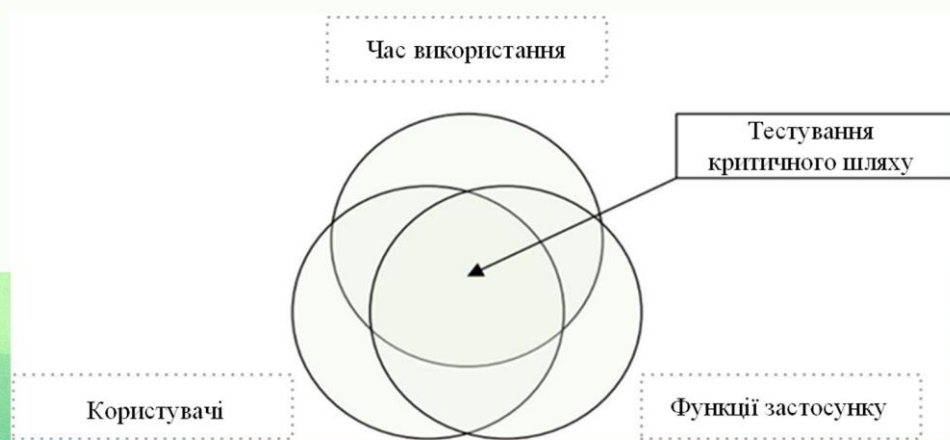
Актуальність дослідження обумовлена потребою розробити методику, яка дозволить ефективно автоматизувати як API-тестування, так і тестування графічного інтерфейсу мобільних застосунків у рамках Agile-проектів.



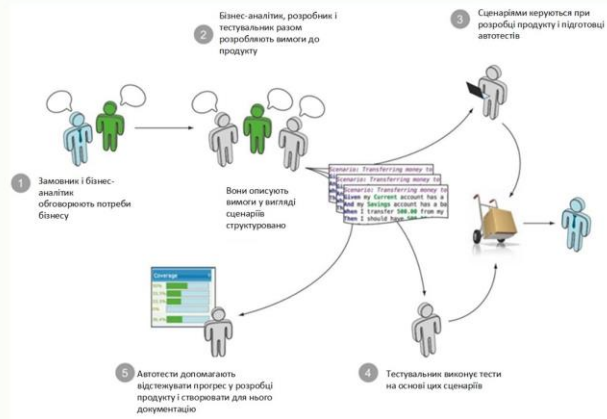
Сьогодні технологія Agile набирає все більшого значення у розробці програмного забезпечення, зокрема мобільних застосунків, що ставить завдання ефективного тестування систем у межах коротких ітерацій. У практиці Agile-проектів для автоматизації тестування зазвичай використовується підхід Scripting, який добре працює для тестування API, але виявляється недостатньо ефективним для створення автотестів графічного інтерфейсу (GUI), через що перевірка GUI переважно здійснюється вручну



Суть тестування критичного шляху

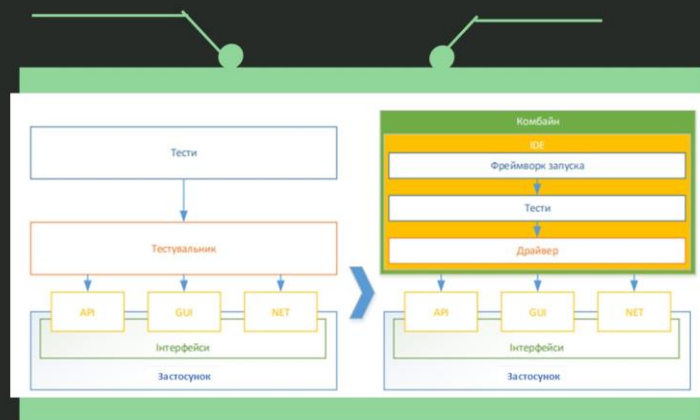


Застосування методології BDD розробки ПЗ



5

Зміна схеми тестування під час автоматизації

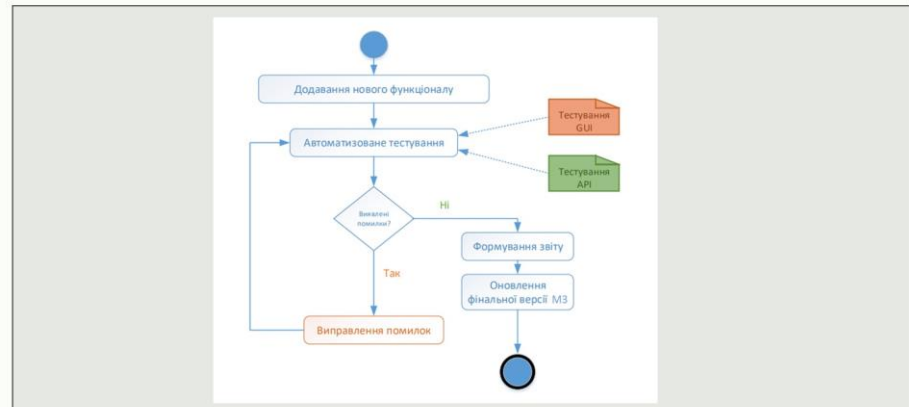


6

Варіант використання МЗ для власника магазину



Алгоритм проведення тестування із застосуванням розробленої методики



Результати запуску автотестів Record and Play в Android Studio

The screenshot shows the 'Run' tab in Android Studio with the following test results:

Test Name	Duration
p.AddBookToStore	10 s 310 ms
addBookToStoreTest_goodDayScenario	6 s 139 ms
addBookTest_badDayScenario_bookAlreadyAdded	1 s 969 ms
addBookTest_badDayScenario_bookNotFound	2 s 202 ms
DeleteBookFromStore	2 s 356 ms
deleteBookTest_badDayScenario_bookNotFound	777 ms
deleteBookTest_goodDayScenario	1 s 579 ms
EditBook	2 s 667 ms
editBookTest_badDayScenario_invalidBookName	855 ms
editBookTest_badDayScenario_bookNotFound	1 s 164 ms
editBookTest_goodDayScenario	648 ms

Summary: Tests passed: 98 of 98 tests – 2 m 10 s 211 ms. Testing started at 4:32 PM ...

Terminal output:

```

05/01 16:32:16: Launching 'Tests in
Running tests
$ adb shell am instrument -w -r
Connected to process 5975 on device
Started running tests
Tests ran to completion.

```

Підготовка тестового сценарію IntelliJ IDEA

```

1 @progression
2 @steps
3 Feature: Authentication good day
4 User specify login and password/>
5 App return validation error for incorrect login/>
6 App return authorization error if user not registered in app/>
7 Authorization is success if user registered in app and specified correct/>
8 login and password/>
9
10 Background:
11 Given start screen
12
13 Scenario Outline: Prepare users for login
14 And user "user" registered in app
15 Examples:
16 | user |
17 | account.user.active.owner.login |
18 | account.user.active.admin.login |
19 | account.user.active.dep-admin.login |
20 | account.user.active.mail.employee.login |
21
22 @correct
23 @severity-blocker
24 Scenario Outline: Success login with correct credentials and logout
25 When user specify "--login" and "--password"
26 Then authorization is success
27 And user push "logout" button "account.general-view.logout"
28 And request "logout" is success
29 Examples:
30 | login | password | comment
31 | account.user.active.owner.login | account.user.active.owner.pass | # login as owner
32 | account.user.active.admin.login | account.user.active.admin.pass | # login as admin
33 | account.user.active.dep-admin.login | account.user.active.dep-admin.pass | # login as dep-admin

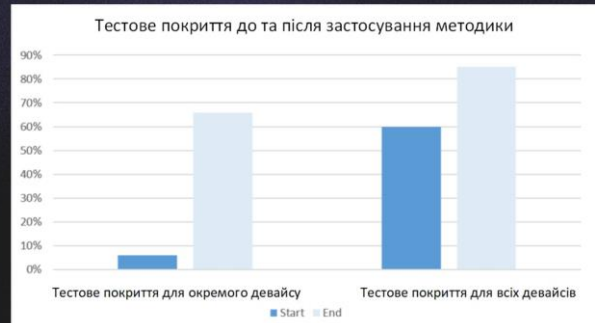
```

Результати виконання автоматизованого тестування МЗ, отримані під час апробації розробленої методики, представлені у таблиці

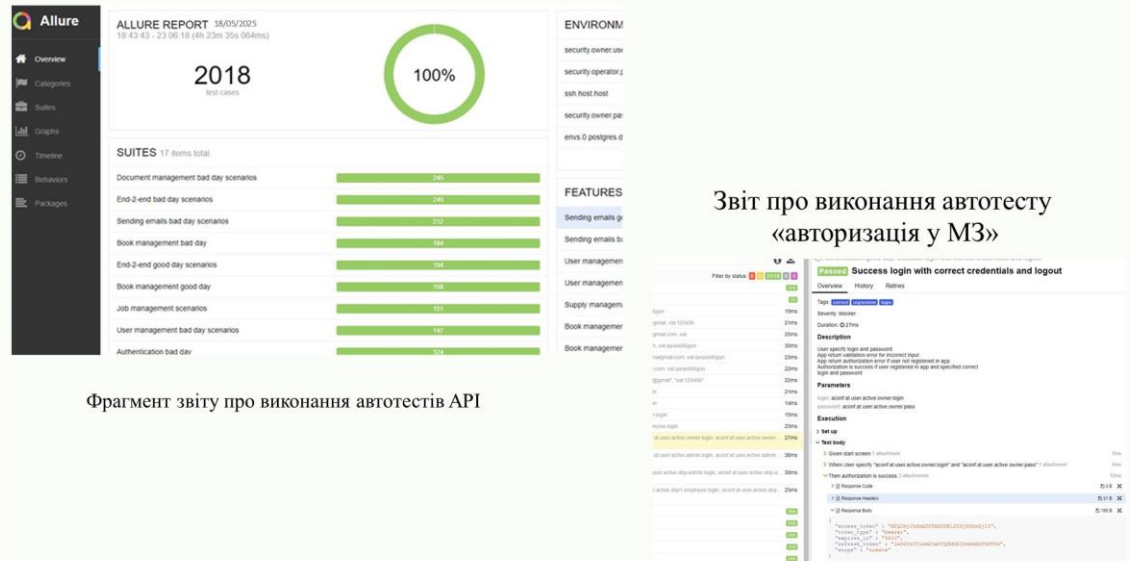
Перевірка	Опис перевірки	Результат тестування
Тестування GUI	Перевірка наявності нових елементів графічного інтерфейсу та їх функціонування	Відповідає специфікації
Тестування GUI	Перевірка працездатності нових елементів GUI на різних пристроях	Виявлені незначні дефекти на деяких комбінаціях пристрій + платформа + версія ОС
Тестування API	Перевірка API-частини нової функціональності на відповідність специфікації	Відповідає специфікації
	Регресійна перевірка раніше створених функціональностей, порушених при додаванні нової функціональності	Виявлені незначні дефекти
	Тестування безпеки передачі даних під час надсилання нових запитів	Виявлені незначні дефекти, пов'язані з порушенням рольової моделі

11

Тестове покриття до та після застосування методики



12



Фрагмент звіту про виконання автотестів API

ВИСНОВКИ

Метою цієї кваліфікаційної роботи є аналіз та розробка ефективної методики автоматизації тестування мобільних застосунків, створених за підходом Agile.

Проаналізовано методику Scripting, яка широко застосовується в Agile-проектах, але виявлено, що вона не повністю відповідає вимогам тестування мобільних застосунків.

Запропоновано та впроваджено комбіновану методику автоматизації, що поєднує Record and Play для тестування GUI та Scripting для API-тестування. Для реалізації використано драйвери, інтегровані у середовища Android Studio та Xcode, а також фреймворк Cucumber у поєднанні з IntelliJ IDEA. Порівняльний аналіз тестових метрик до і після впровадження показав збільшення загального тестового покриття мобільного додатка на 25% для всіх пристроїв та на 60% для кожного окремого, а кількість виявлених дефектів зросла з 12 до 72, переважно за рахунок помилок з низьким пріоритетом. Отримані результати свідчать про суттєве підвищення ефективності тестування завдяки розробленій методиці, орієнтованій на застосування в Agile-проектах.

