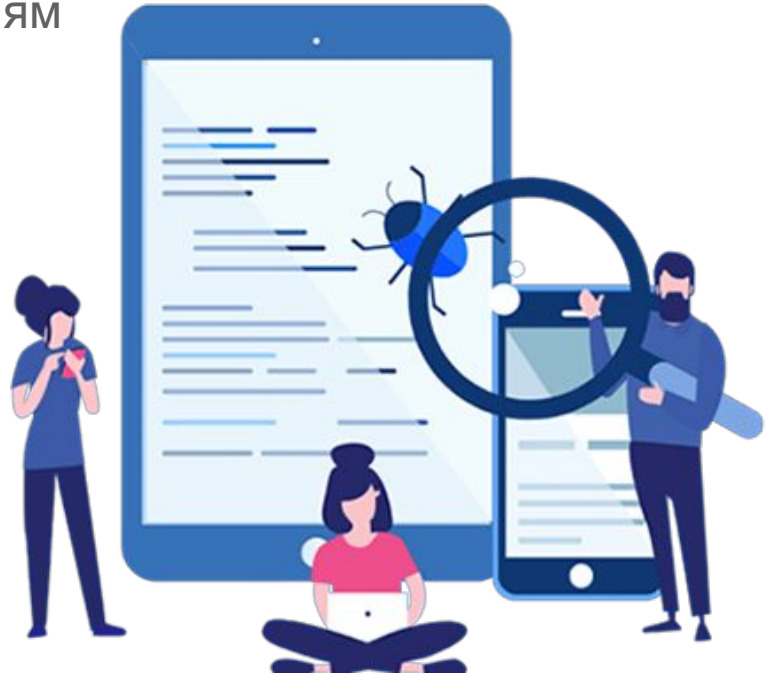
Практична робота 1

Мета: Ознайомити з основами тестування та створенням запису про помилку.

Завдання 2: Створити пять репортів багів з використанням різних технік дизайну. Можно не обмежуватися техніками які описані в завданні. Проект обрати на свій розсуд.

Вимоги до звіту: Використати заголовки
назву техніки тест дизайну, перед написанням багу короткий опис використання методу.
Не забути вказати группу та ПІБ.

Звіт завдання скидати на почту:
asp_kam1@student.ztu.edu.ua



Вимоги до оформлення багу

ID/ім’я : будьте короткими та використовуйте правильні терміни. Найкраще вказати назву функції, у якій ви виявили проблему. Хорошим прикладом може бути «КОШИК – Неможливо додати новий товар у кошик»

Опис/резюме: якщо ви вважаєте, що назва недостатня, поясніть помилку кількома словами. Поділіться ним легкою для розуміння мовою. Майте на увазі, що ваш опис може бути використаний для пошуку у вашій програмі відстеження помилок, тому використовуйте правильні слова.

Середовище: залежно від вашого браузера, операційної системи, рівня масштабування та розміру екрана веб-сайти можуть по-різному поводитися в одному середовищі. Переконайтеся, що ваші розробники знають ваше технічне середовище.

Вихідна інформація: інформація яка допоможе швидше знайти місце помилки та саму помилку (кеш, посилання на сторінку, помилка в консолі ...)

Візуальний доказ: зображення варте тисячі слів. Хоча цього може бути недостатньо, візуальний елемент, наприклад знімок екрана чи відео, допоможе вашим розробникам краще та швидше зрозуміти проблему.

Дії для відтворення: знімок екрана є доказом того, що у вас виникла проблема, але майте на увазі, що ваш розробник може не вдатися відтворити помилку. Обов’язково якомога докладніше опишіть кроки, які ви виконали до того, як виявили помилку.

Очікувані та фактичні результати (Expected vs. actual results) - поясніть, яких результатів ви очікували – будьте якомога конкретнішими. Просто сказати, що «додаток працює не так, як очікувалося», не має користі. Також корисно описати те, що ви насправді пережили.

Severity and Priority: Ви також можете включити додаткову інформацію, таку як серйозність (критична, велика, другорядна, тривіальна, покращення), пріоритет (високий, середній, низький), ім’я особи, яка повідомляє, призначену особу або термін виконання.

Event button doesn't work in voice room

Status: DONE

Assignee: [redacted]

Priority: = Major

Resolution: Done

More

Description

Devices Model:	iPhone 11; Xiaomi Redmi note 10; iPad Air 4
Platform:	Android, iOS
OS Version:	Android 12
Environment:	Test
Build Version:	0.4.1 APK#638 ios#134
Reproduce Rate:	100%
Reproduce Time:	5-10 min

Steps to reproduce:

1. Launch the game
2. Log in using valid method
3. Event Valentine's Day is active
4. Enter voice room
5. Tap on Event button

Actual result:
Event button doesn't work in the voice room,event leaderboard doesn't open by tapping on it

Expected result:
Tap on event button opens event leaderboard

Приклад

Техніки тест дизайну

Еквівалентний Поділ (Equivalence Partitioning — EP)
Аналіз Граничних Значень (Boundary Value Analysis — BVA)
Причина / Наслідок (Cause/Effect — CE)
Передбачити помилку (Error Guessing — EG)
Вичерпне тестування (Exhaustive Testing — ET)
Попарне тестування (Pairwise Testing)

Equivalence Partitioning

Еквівалентне розбиття класів передбачає поділ тестових даних на класи, де всі елементи певним чином схожі. Ця техніка має сенс лише в тому випадку, якщо компоненти схожі і можуть поміститися в загальну групу.

ПРИКЛАД ЕКВІВАЛЕНТНОГО РОЗПОДІЛУ КЛАСІВ

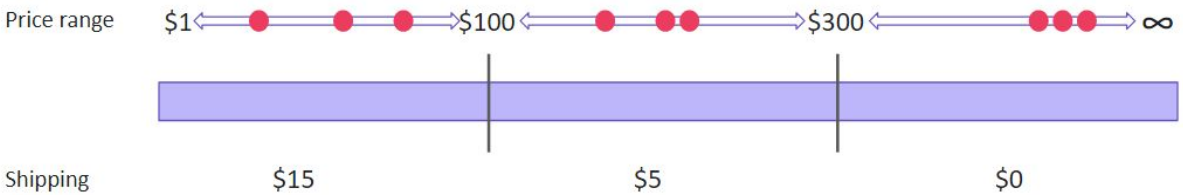
Скажімо, є інтернет-магазин, який пропонує різні тарифи на доставку в залежності від ціни кошика. Наприклад:

1. Вартість доставки замовлень на суму менше 100 доларів США становить 15 доларів США.
2. Вартість доставки замовлень на суму понад 100 доларів США становить 5 доларів США.
3. Безкоштовна доставка замовлень на суму понад 300 доларів США.

У нас є такі цінові діапазони для роботи:
Від $0 до $100.
Від 100 до 300 доларів.
300 доларів і вище.

Якщо ви використовуєте еквівалентну техніку розподілу класів, ви отримуєте три набори даних для перевірки:

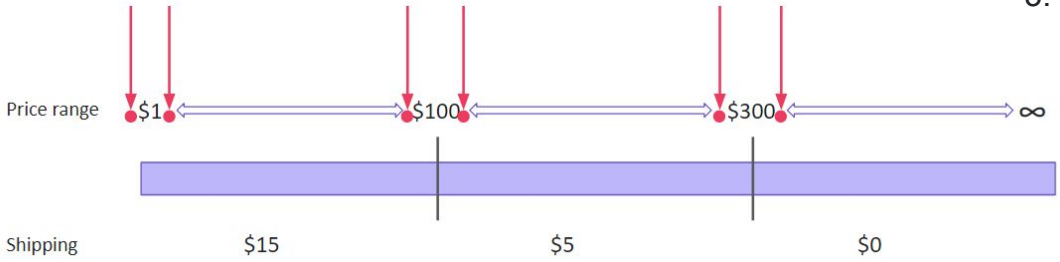
1. Від $1 до $100:
   – дійсні граничні умови: будь-яка ціна в діапазоні від 1 до 99,99;
   – недійсні граничні умови: будь-яка ціна нижче 1 або вище 99,99;
2. Від 100 до 300 доларів:
   – діючі граничні умови: будь-яка ціна в діапазоні від 100 до 299,99;
   – недійсні граничні умови: будь-яка ціна нижче 100 або вище 299,99;
3. $300 і вище:
   – дійсні граничні умови: будь-яка ціна вище 299,99;
   – недійсні граничні умови: будь-яка ціна нижче 300.



Отже, ми можемо просто вибрати кілька чисел із кожного цінового діапазону та припустити, що решта однакових вхідних даних дасть однакові результати.

Boundary Value Analysis

Аналіз граничних значень подібний до попереднього методу. Деякі навіть можуть сказати, що це засновано на еквівалентному розподілі класів. Отже, чим відрізняється аналіз граничних значень? Ми все ще групуємо дані в еквівалентні класи, але не перевіряємо значення лише з певного класу. Замість цього ми перевіряємо граничні значення, ті, що знаходяться на «кордонах» класів. Ця ж логіка ідеально працює для інтеграційного тестування. Ми перевіряємо менші елементи під час модульного тестування, і на наступному рівні помилки, швидше за все, з'являться на з'єднаннях блоків.



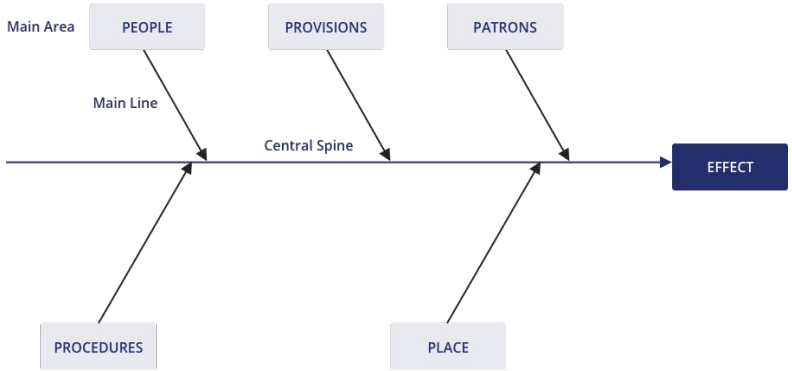
ПРИКЛАД АНАЛІЗУ ГРАНИЧНИХ ЗНАЧЕНЬ

Візьмемо попередній сценарій із різними тарифами доставки. У нас ті самі дані, але інший підхід до їх використання. Припускаючи, що помилки найімовірніше виникають на межах, ми перевіряємо лише «граничні» числа:

1. Від $1 до $100:
   – дійсні граничні умови: 1.00, 1.01, 99.99;
   – недійсні граничні умови: 0,99, 100,00, 100,01;
2. Від $100 до $300:
   – дійсні граничні умови: 100,00, 100,01, 299,99;
   – недійсні граничні умови: 99,99, 300,00;
3. $300 і вище:
   – дійсні граничні умови: 300,00, 300,01;
   – недійсні межі: 299,99.

Cause/Effect

Техніка на основі графіка причинно-наслідкових зв’язків — це техніка, у якій графік використовується для представлення ситуацій комбінацій вхідних умов. Потім графік перетворюється на таблицю рішень для отримання тестових випадків. Використовується метод побудови графіків причинно-наслідкових зв’язків, оскільки аналіз граничних значень і методи поділу класів еквівалентності не враховують комбінації вхідних умов. Але оскільки при розгляді деяких комбінацій вхідних умов може бути певна критична поведінка, яку потрібно перевірити, саме тому використовується техніка причинно-наслідкового графіка.



Наприклад, ви перевіряєте можливість додавати клієнта, використовуючи певну екранну форму. Для цього вам необхідно буде ввести кілька полів, таких як "Ім'я", "Адреса", "Номер Телефону", а потім натиснути кнопку "Додати" - це "Причина". Після натискання кнопки «Додати» система додає клієнта в базу даних і показує його номер на екрані — це «Слідство».

Error Guessing

Вгадування помилок є найбільш експериментальною практикою з усіх, яка зазвичай застосовується разом з іншою технікою розробки тестів. Під час вгадування помилок інженер із забезпечення якості прогнозує, де ймовірно з’являться помилки, спираючись на попередній досвід, знання системи та вимоги до продукту. Таким чином, фахівець з контролю якості повинен виявити місця накопичення дефектів і приділити їм підвищену увагу.

Exhaustive Testing

Це крайній випадок. У межах цієї техніки ви повинні перевірити всі можливі комбінації вхідних значень, і в принципі це має знайти всі проблеми. Насправді застосування цього методу неможливо, через велику кількість вхідних значень.

Pairwise Testing

Попарне тестування вважається найскладнішим і заплутаним із п’яти методів розробки тестів. І для цього є вагома причина. Попарне тестування базується на математичних алгоритмах, а саме комбінаториці. Це дає можливість створювати унікальні пари та тестувати величезну кількість вхідних даних у різних комбінаціях, але обчислення можуть бути складними. Щоб охопити максимум функцій тестовими сценаріями, які вимагатимуть мінімум часу для тестування, потрібно правильно зіставляти дані, комбінуючи пари певним чином на основі обчислень.

ПРИКЛАД ПАРНОГО ТЕСТУВАННЯ

Скажімо, є мережа пекарень, які продають яблучні пироги та сирники в Інтернеті. Кожен доступний у трьох розмірах – малий, середній і великий. Пекарня пропонує негайну та заплановану адресуну доставку, а також можливість самовивозу. Пекарня працює в трьох містах – Нью-Йорку, Лос-Анджелесі та Чикаго. Також користувач може замовити до трьох товарів одночасно.

Order	Size	City	Quantity	Delivery	Time
Apple Pie	Small	New York	1	Address	Now
Cheesecake	Medium	Los Angeles	2	Pick-Up	Schedule
	Big	Chicago	3		

Якщо ви хочете перевірити всі можливі вхідні дані, це буде 2x3x3x3x2x2=216 дійсних комбінацій порядку. Однак перевіряти кожен з них було б нерозумно. Натомість ви можете розташувати змінні таким чином, щоб охопити максимальну кількість сценаріїв.У результаті ми отримали 17 сценаріїв, здатних охопити всі 216 комбінацій. Ви можете побачити список комбінацій нижче.

	Order	Size	City	Quantity	Delivery	Time
1	Apple Pie	Big	Chicago	3	Address	Now
2	Cheesecake	Big	New York	2	Address	Schedule
3	Cheesecake	Small	Los Angeles	1	Pick-Up	Now
4	Cheesecake	Medium	New York	2	Address	Schedule
5	Cheesecake	Small	Los Angeles	3	Pick-Up	Now
6	Apple Pie	Big	Los Angeles	2	Pick-Up	Now
7	Apple Pie	Small	New York	3	Address	Schedule
8	Apple Pie	Small	Chicago	2	Pick-Up	Schedule
9	Apple Pie	Medium	New York	1	Address	Now
10	Cheesecake	Medium	Chicago	1	Pick-Up	Schedule
11	Cheesecake	Medium	Los Angeles	3	Address	Schedule
12	Cheesecake	Big	New York	1	Pick-Up	Now
13	Apple Pie	Medium	New York	3	Pick-Up	Now
14	Apple Pie	Small	Los Angeles	1	Address	Schedule
15	Apple Pie	Medium	New York	2	Pick-Up	Now
16	Apple Pie	Big	Los Angeles	1	Address	Schedule
17	Apple Pie	Small	Chicago	2	Address	Now