

## Лабораторна робота 4

### Моделювання процесу Конструювання. Кодування та тестування

Мета – отримати навички складання тестової документації відповідно до вимог.

#### Теоретичні відомості

До основної тестовій документації відносяться:

1. Test Case – це тестовий артефакт, суть якого полягає у виконанні певної кількості дій і/або умов, необхідних для перевірки певної функціональності програмної системи, що розробляється.

Структура даного артефакту полягає в наступному: що треба зробити (Action) – очікуваний результат (Expected result) – фактичний результат (Test result) (рис.1).

Тест-кейс ID	TC_9		
Назва	Перевірка автозаповнення		
Пріоритет	Medium		
Опис	Перевірка на правильність автозаповнення поля пошуку товарів при переході на категорію		
Модуль	Перевірки вмісту, UI та введення		
Створено	Зозуля Ірина	Дата створення	17.02.2021
Протестовано	Зозуля Ірина	Дата тестування	17.02.2021
Попередні умови	Користувач бажає перейти на сторінку певної категорії		
Постумови	-		

№	Крок тесту	Дані тесту	Очікуваний результат	Фактичний результат	Статус
1	Перехід на сторінку категорії	-	Переадресація на сторінку категорії, сторінка відкривається	Як очікувано	Passed
2	Перегляд сторінки категорії	-	Вміст сторінки виводиться коректно та поле пошуку правильно автозаповнюється	У полі пошуку з'явилась неправильна назва категорії	Failed (BG_3)

Рис.1 Test Case

2. Testplan (план тестування) – документ, що описує весь обсяг робіт з тестування, починаючи з опису тестованих об'єктів, стратегії, розкладу, критеріїв початку і закінчення тестування, до необхідного в процесі роботи обладнання, спеціальних знань, оцінки ризиків з варіантами їх вирішення. Тест-план є важливою складовою будь-якого процесу тестування, оскільки містить усю необхідну інформацію, що описує цей процес. Але в більшості випадків Тест-план переважно має формальне значення, але, все ж, наявність надає багато переваг. Залежно від конкретизації описуваних завдань, тест-план може мати два рівні деталізації: майстер тест-план і детальний тест-план.

Детальний тест-план містить завдання тестування для кожної команди, для кожного релізу або ітерації проекту. Детальний тест-план створюється або для декомпованих частин проекту, або для невеликих проектів і складається з:

- переліку областей тестування з пріоритетами;
- стратегії тестування;
- переліку можливих ризиків;

- переліку необхідних ресурсів;
- плану виконання проекту.

Майстер тест-план створюється або для організації процесу тестування між декількома командами, які тестують один проект, але мають різні завдання, або для проекту, який складається з безлічі ітерацій, які пов'язує якась загальна інформація, повторення якої в кожному релізі займає надто багато часу. Майстер тест-план містить:

- загальну інформацію про проект (посилання на документацію, багтрекер, і т.д.);
- положення, що описують процес тестування, закладу дефектів і т. д.;
- критерії готовності продукту до випуску.

Існують кілька шаблонів тест-планів (IEEE, RUP).

3.Баг репорт (bugreport) – це технічний документ, який містить повний опис бага з інформацією як про сам баг (короткий опис, серйозність, пріоритет і т. д.), так і про умови виникнення цього бага (рис.2). Баг репорт повинен містити правильну, єдину термінологію, що описує елементи призначеного для користувача інтерфейсу та події цих елементів, що призводять до виникнення бага. У загальному випадку, баг-репорт містить:

1) шапку:

- короткий опис (короткий опис проблеми);
- проект (назва поточного проекту);
- компонент додатка, у якому виник дефект;
- версія (версія білда, у якому знайдено баг);
- серйозність (градація ступеня впливу на додаток бага);
- пріоритет (черга виправлення бага);
- статус (відображає статус бага в своєму життєвому циклі);
- автор (автор баг репорт);
- призначення (хто повинен виправити дефект);

2) оточення:

- операційна система, розрядність, браузер, його версія і т. д.;

3) опис:

- кроки відтворення (опис шляху, який призводить до виникнення дефекту);
- фактичний результат (результат, який отримується після виконання всіх кроків відтворення);
- очікуваний результат (результат, який бути відповідно до вимог).

4) додатки:

- приєднаний файл (логи, скріншоти, інші документи, які можуть допомогти відтворити проблему або розв'язати цю проблему).

Основні поля, присутність яких необхідна:

– короткий опис. Поле, де ви хочете розмістити весь сенс усього баг репорт. Найчастіше, у короткому описі лаконічно відповідають на 3 питання: «Де?», «Що?», «Коли?» (саме в такій послідовності, як би не хотілося змінити її за прикладом відомої всім гри);

– серйозність. Дефект або повністю зупиняє працездатність програми, або тільки частину функціональності, або інше;

– кроки до відтворення. Точний і зрозумілий опис усіх кроків, які призводять до появи дефекту, з урахуванням усіх необхідних вхідних даних;

- фактичний результат;
- очікуваний результат.


ID	BG_3
Пріоритет	низький
Серйозність	5-низька
URL	<a href="https://eva.ua/ua/q-tester-zhinocha/">https://eva.ua/ua/q-tester-zhinocha/</a>
Версія	Десктопна, IOS, Android
Опис	У рядку «Пошук» відображається «тестер жіноча» замість «Тестери парфумів для жінок»
Кроки для відтворення	Перехід по даному шляху: Головна сторінка->Меню->Парфумерія->Тестери->Для Жінок
Фактичний результат	 <p>У полі пошуку з'явилась неправильна назва категорії</p>
Очікуваний результат	Відображення «Тестери парфумів для жінок» або пустого рядка у полі пошуку

Рис.2 Баг репорт

### Завдання

1. Згідно виконаної роботи (лабораторні роботи 2-3) створити працюючий прототип.
2. Мову програмування визначити згідно архітектури (лабораторні роботи 2-3)
3. Підготувати сценарії для тестування (Таб. 1).
4. Створити 10 тест-кейсів.
5. Написати 5 баг-репортів згідно зі структурою, наведеною в коротких теоретичних відомостях.
6. Протестувати прототип. Презентувати результат.

### Результат тестування

Таблиця 1

Етап	№ вимоги	Вхідні дані	Кроки	Очікувані результати	Реальні результати
1					
2					
3					
4					
...					

### Контрольні питання