

1 ЛАБОРАТОРНА РОБОТА №1

СТВОРЕННЯ БАГ-РЕПОРТІВ

Мета лабораторної роботи: отримання базових теоретичних та практичних навичок в розпізнаванні етапів тестування та створенні баг-репортів

1.1 Короткі теоретичні відомості

Баг полягає в основній ідеї тестування та характеризується як *невідповідність фактичної роботи програмного забезпечення очікуваному результату*, тобто технічному завданню чи специфікації від замовника. Перевірка на баги відбувається не тільки безпосередньо вже на етапі тестування готового продукту, а й на етапі аналізу техзавдання. Під тестування може потрапляти як інтерфейс у зручності користування, так і точність дизайну продукту, або функціональна складова бекенду програми. **Кожна помилка системи має бути задокументована.** І звіт, в якому вказані всі баги, називається баг-репорт. У ньому вказується детальний опис помилки, що сталася на етапі тестування та всі дії, що до неї призвели.

Кожен баг-репорт складається з кількох важливих атрибутів, які обов'язково вказуються у ньому:

- **Саммарі** (стислий опис помилки в одному рядку) - будується за принципом: що не працює? де не працює? коли? чи за яких умов не працює? Наприклад,- "Не здійснюється оплата, після натискання на кнопку "Купити" на сторінці купівлі товару".
- **Опис помилки** (кроки відтворення помилки), має бути коротким і ясным програмісту. Тут вказуються всі кроки, тобто умови, виникнення бага, фактичний результат та очікуваний результат роботи програми.
- **Проект** - вказується найменування програми, сайту чи іншої програми, до якої належить наш баг, для подальшого відстеження усунення цього дефекту.
- **Компонент** - під ним слід розуміти частину програми, або ділянку розробки, в якій було допущено баг. Якщо у різних частинах продукту є схожі ділянки коду, рекомендується перевірити один баг на кількох ділянках розробки. Наприклад,

каталог інтернет-магазину передбачає наявність фільтрів для деталізації пошуку певного товару, але також на різних категоріях товару є ймовірність ділянки розробки коду, що повторюється, для фільтрів, які можуть бути просто скопійовані розробником. Якщо є баг в одній частині, він неодмінно буде і в іншій ділянці.

- **Серйозність** - тобто критичність помилки, або наскільки дана помилка впливає на працездатність нашого програмного забезпечення. Найчастіше це - несуттєвий (Minor), значний (Major), критичний (Critical) чи блокуючий (Blocker).
- **Пріоритет** – послідовність виправлення багів, виходячи з попередньої атрибуту бага. Зазвичай це високий (High), середній (Medium) і низький (Low). Важливо відзначити, що пріоритет тестувальник не проставляє - цим займається менеджер або той, хто займається розподілом завдань на програмістів.
- **Статус** - це фіксація етапу усунення бага у його життєвому циклі. Наприклад, відкритий (Open), закритий (Closed), новий (New) або перевідкритий (Reopen).
- **Версія білда**, тобто версія нашого продукту, в якій припущена помилка. Іноді цей атрибут може бути не вказаний, якщо немає версії продукту або бага відтворюється у всіх версіях програмного забезпечення.
- **Автор** - той учасник команди, який вносить інформацію про баг.
- **Оточення** - під це поняття може потрапляти ОС, розряд системи, браузер, у якому було виявлено помилку.

Найважливішими атрибутами кожної помилки в баг-репорті є **серйозність і пріоритет**. І якщо рівень серйозності вказує тестувальник, то **за пріоритетністю їх усунення відповідає РО (Product Owner) з боку замовника або проджект менеджер (PM)**. Оскільки є незначні помилки, які не ламають додаток, наприклад, найменування інтернет-магазину, в якому допущена помилка, але їх усунення першочергово важливе для престижу компанії замовника.

Розглянемо класифікації ступеня серйозності багів та його вплив на працездатність програмного забезпечення.

- **Blocker** – та помилка, яка призводить до непрацездатності нашої програми. Тобто її подальше використання користувачем стає неможливим.
- **Critical** – є кілька варіантів: відхилення від логіки роботи програмного забезпечення, задуманого замовником; неробочий функціонал окремих елементів ПЗ (наприклад, відсутність переходу в кошик після натискання кнопки "Купити" на сайті інтернет-магазину); не збереження даних сеансу або користувача.
- **Major** - є схожість з критичними помилками, але в даному випадку програма працює, але не відповідає запитам замовника.
- **Minor** – незначний дефект, що не порушує працездатність ПЗ. Наприклад, помилка в дизайні програми.
- **Trivial** - найменш незначний баг програмного забезпечення, наприклад, текстова друкарська помилка. Вона ніяк не впливає на працездатність, але не відповідає бажаному результату технічного завдання.

Пріоритети відповідають за порядок виправлення багів на проєкті. Існує три найпоширеніші пріоритети: **High, Medium і Low** (іноді їх можуть класифікувати цифрами за тією ж логікою). Баг High повинен виправитися в першу чергу, зазвичай цей пріоритет поширюється на Blocker і Critical ступені серйозності, які можуть зашкодити програмному забезпеченню. Відповідно, після виправляються Medium і Low пріоритети, які критично не впливають на працездатність функцій софту. Для правильної назви заголовка бага в атрибуті опису можна використовувати золоте правило "**що?**", "**де?**" і "**коли?**". Що - що не працює коректно, Де – місце, в якому виявлено помилку, коли – за яких умов. Чим коректнішим і зрозумілішим буде опис бага, тим швидше він буде усунений. Оскільки, якщо баг буде не відтворюваний, то найімовірніше програміст відхилить цю помилку і вона не буде виправлена. **Усі баг-репорти вносяться до баг-трекінгових систем.** По суті, баг-трекінгові системи створені не тільки для фіксації багів. В них також вносяться і User-стори, технічні вимоги від РМ реалізації у продукті та інша документація. Найпоширенішими баг-трекінговими системами є Jira (найбільш використовувана), Asana, Redmine, Trello и Bugzilla.

Після початкових усунень багів, розроблений **продукт тестується повторно**, окремо на відтворення бага, і окремо за функціоналом, який міг бути порушений під час виправлення помилок. Потрібно розділяти повторне тестування бага і повторне тестування суміжного компонента, щоб не дублювати фіксацію однієї помилки. Перед написанням баг-репорту ***обов'язково відтворіть помилку кілька разів***. Оскільки вона могла виникнути у зв'язку з нестабільною роботою мережі, або через неполадки ПК та інші причини, не пов'язані з працездатністю самого програмного забезпечення. Якщо виникнення бага повторюється, необхідно відразу зафіксувати помилку та повідомити про неї програміста. Також краще заздалегідь перевірити помилку з різних браузерів та з мобільної версії, щоб максимально точно локалізувати місця, де щось не працює. Дуже часто, можна спостерігати помилки тільки в браузері Chrome, у свою чергу, як у Firefox все буде працювати коректно. І навпаки. Перед надсиланням баг-репорту програмістам необхідно перечитати його кілька разів, перевірити на орфографічні помилки, ще раз оцінити стислість і ясність написаного звіту. Баг-репорт є одним із найважливіших документів тестувальника, він показує вашу грамотність та кваліфікованість.

Заповнення баг-репортів.

Поля:

- Поле "Summary" (короткий опис дефекту) потрібно описувати за принципом "Що? Де? Коли"
- У темі, описі та результатах повинна дотримуватися послідовність "Що? Де? Коли?". Відповідати на запитання необхідно саме в такій послідовності.
- Тему, опис та результати необхідно описувати у теперішньому часі, відповідаючи на питання: "Що відбувається?".
- Допускається формулювання теми, опису та результатів в баг-репортах за принципом "Що?, Де?" без вказівки "Коли?", в разі, якщо баг відображається на сторінці завжди без будь-яких додаткових дій/подій/умов (наприклад, в багах верстки).
- Тема бага повинна повністю вміщатись в рядку "Summary".

- Поле "Description" потрібно описувати одним реченням за принципом "Що? Де? Коли?" згідно "Summary". Можна додати 1 додаткове речення, якщо це є важливою інформацією і не вміщується в "Summary" через обмеження за кількістю символів.

Кроки;

- В першому кроці необхідно вказати посилання на головний домен, а в наступних кроках описувати послідовність дій.
- Посилання необхідно вказувати тільки в першому кроці. (Див. детальний опис в статті.
- У кроках необхідно відповідати на запитання: "Що необхідно зробити?", без звернень до третьої особи. Наприклад: Відкрити ..., Натиснути ..., Звернути увагу ...
- У кроках необхідно коротко писати, що зробити, куди натискати, не уточнюючи назви сторінки, поп-ап вікна у кожному кроці.
- У кроках необхідно вказувати які дані вводяться в поле (валідні або невалідні). У разі невалідних повинні бути наведені приклади (наприклад: "Ввести спецсимволи в поле" Name "(*, @, /, <, >)").
- Необхідно скоротити кількість кроків. Як правило, в баг-репорті описується не більше 7-8 кроків. Частина інформації можна винести у передумови.
- В останньому кроці не потрібно детально описувати помилку і дублювати фактичний результат. Потрібно акцентувати увагу на сам елемент з дефектом (Наприклад: Pay attention to the ..., Звернути увагу на ...).
- Кілька кроків варто об'єднати в один (наприклад, "Заповнити обов'язкові поля форми валідними даними"). Як правило, в баг-репорті описується не більше 7-8 кроків.
- Не варто об'єднувати багато кроків в один (наприклад: Click the "Login" button and fill the "Facebook Account" form with valid data and then click the "OK" button). Необхідно розділити складовий крок на кілька окремих кроків.
- У кроках не зазначено інформацію про роль користувача або його авторизаційні дані.

Результати :

- Очікуваний результат слід вказувати після фактичного результату
- Результати необхідно описувати згідно теми за принципом "Що? Де? Коли?", відповідаючи по порядку на ці питання.
- Результати не потрібно нумерувати та не варто писати кілька результатів в одному дефекті.
- Результати в поточному баг-трекері необхідно описувати відразу після кроків в блоці "Steps To Reproduce".
- Краще уникати формулювань очікуваного і фактичного результатів, які відрізняються лише негативною формою. Очікуваний результат необхідно детально описати, недостатньо просто додати заперечення в одне з речень.

Скріншот/відео:

- Розширення скріншота має бути: ".png", ".bmp", ".jpeg".
На скріншоті потрібно відобразити курсор (вказівник миші).
На скріншоті достатньо одного прямокутника і однієї стрілки (прямокутником потрібно виділяти місце з помилкою, а не весь блок).
- На скріншоті/відео не повинна відображатися панель закладок браузера або інша зайва інформація (задній фон робочого столу, вікно програвача відео-файлів, сторонні повідомлення та ін.).
- На скріншоті/відео необхідно показати вікно браузера з адресним рядком.
На скріншоті необхідно додавати червону стрілку і прямокутник, використовуючи вбудовані засоби графічного редактора (наприклад, Snagit).
- На скріншоті проблемне місце потрібно виділити червоним прямокутником і вказати на нього червоною стрілкою.
- На скріншоті червоним прямокутником і червоною стрілкою виділяється проблемне місце, а зеленим прямокутником і зеленою стрілкою - правильний приклад або макет.
- Необхідно прикріпити посилання на відео, яке демонструє описаний дефект на додаток до скріншоту. Скріншот видаляти не потрібно.

- Відео необхідно завантажувати формату ".avi", ".mp4", ".mkv".

Оформлення

Інше:

- При тестуванні сайтів на десктопних пристроях в поле "OS" пишеться назва операційної системи (наприклад, Windows), в поле "OS Version" - версія операційної системи (наприклад, 7x64), а в полі "Platform" - назва браузера і його версія.
- Необхідно вказати браузер і його версію в полі "Platform". Якщо баг повторюється в різних браузерах, їх потрібно вказати в розділі "Additional information".
- Необхідно вказати версію браузера, так як поява багатьох дефектів залежить саме від версії браузера.
- Пропущені деякі атрибути в баг-репорті (тема, опис, кроки, результати, скріншот/відео, тестове оточення). (Див. детальний опис в
- У темі та результатах потрібно вказати, яка саме помилка відображається без застосування слів "некоректно/коректно", "невірно/вірно", "правильно/неправильно", "incorrect/correct", "wrong/right". Наприклад: Зображення товару розтягується (Що?) у формі "Надіслати другу" (Де?)
- Необхідно перефразувати тему або результати, вказавши яка саме помилка відображається. Не варто писати "нічого не відбувається", "не працює", "ламається верстка", "nothing happens", "the layout is broken". Наприклад: Вікно завантаження зображення не з'являється на сторінці редагування профілю ... The menu items are shifted to the left.
- В баг-репорт не варто використовувати слова на кшталт "можна/не можна", "можливо/неможливо", "possible/impossible", "can/can not". Необхідно точно описувати суть бага, вказуючи, в чому саме невідповідність.
- Особисті займенники ("ви", "ми", "я"), а також слова "користувач/user" краще не використовувати, перефразувавши опис баг-репорта від третьої особи, акцентуючи увагу на об'єкт дії: "сайт", "кнопка", "посилання" та ін.

- Тему та результати слід описувати повними реченнями з підметом і присудком.
 - Поряд з назвою елемента потрібно писати його тип (кнопка, посилання, поле і т.д.), так як на сайті може бути декілька елементів з однаковими назвами.
 - Не потрібно перекладати назви елементів сайту на мову оформлення звіту.
Назви кнопок, полів, посилань, повідомлень необхідно вказувати саме так, як вони відображені на сайті.
 - Коли баг відображається після скоєної дії правильно писати "після" замість "при". Наприклад, "після натискання ...", "після наведення ...".
 - Не потрібно створювати дублікати з описом одного і того ж дефекту, який відтворюється на різних сторінках або для різних елементів. У розділі "Additional Information" вказується перелік сторінок та елементів з однотипними багами.
 - Тему, опис, кроки, назви результатів необхідно писати з великої літери.
 - Необхідно використовувати лапки в назвах елементів сайту.
- Приклад наведено у таблиці 1.1.

Таблиця 1.1 – Створення баг-репорту

номер ID	Проект	Категорія	Рівень доступу	Дата реєстрації	Дата зміни
1					
Тестувальник		Пріоритет		Серйозність	
Відтворюваність		Статус		Резолюція	
Платформа		Операційна система		Версія ОС	
Тема					
Опис					
Кроки для відтворення					
Фактичний результат					
Очікуваний результат					
Додаткова інформація					
Прикріплені файли					

1.2 Завдання до лабораторної роботи

- 1.2.1. Відкрити сайт, наприклад: <http://prestashop.gatestlab.com.ua/en/>.
- 1.2.2. Знайти 3 баги, та оформити баг-репорти, див. таблицю 1.1. Додати скріншоти оформлених баг-репортів до звіту.
- 1.2.3. Оформити звіт.

1.3 Зміст звіту

- 1.3.1 Тема та мета роботи.
- 1.3.2 Завдання до роботи.
- 1.3.3 Короткі теоретичні відомості.
- 1.3.4 Виконати завдання до лабораторної роботи.
- 1.3.5 Висновки, що відображують результати виконання роботи та їх критичний аналіз.
- 1.3.6 Перелік використаних джерел.

1.4 Контрольні запитання

- 1.4.1. Визначити основні властивості та відмінності встановлених браузерів: Mozilla Firefox, Opera, Google Chrome, Edge. Обґрунтувати розповсюдженість використання встановлених браузерів.
- 1.4.2. Для чого призначені та які існують системи відслідковування помилок?
- 1.4.3. Що повинен містити якісний опис помилки?
- 1.4.4. Що відносять до атрибутів дефекту?