IT CAREER HUB

Урок 3.
Баг-репорты

Error, defect, failure

В тестировании программного обеспечения (ПО) понятия ошибка (error), дефект (defect) и сбой (failure)имеют четкие различия:

Ошибка — это результат человеческого фактора в работе, когда разработчик или другой участник процесса сделал неправильное действие или допустил недочет.

Примеры:

- Неверное понимание требований.
- Ошибка при написании кода (например, опечатка или неверная логика).
- Неправильное проектирование архитектуры.

Следствие:
Ошибка может привести к появлению дефекта в коде или системе, но сама по себе она не проявляется в продукте, пока не реализована.

Дефект (иногда называют багом, ошибкой или недочетом) — это результат ошибки, выраженный в коде или другом компоненте системы.
Это отклонение от спецификаций, ожидаемого поведения или требований.

Примеры:

- Логическая ошибка в функции.
- Неправильно настроенный интерфейс пользователя.
- Несоответствие между фактическим поведением системы и её спецификацией.

Следствие:
Дефект может существовать в системе, но не всегда приводит к сбою. Например, дефект может "проявляться" только при определенных условиях.

Сбой — это проявление дефекта в работе системы. Это ситуация, когда система работает некорректно или не выполняет ожидаемую задачу.

Примеры:

- Программа выдает неверный результат.
- Приложение зависает или аварийно завершает работу.
- Сайт отображается некорректно в браузере.

Следствие:
Сбой приводит к тому, что пользователь замечает проблему. Не все дефекты приводят к сбоям, но любой сбой указывает на наличие дефекта.

Взаимосвязь:

1. Ошибка (Error) → совершается человеком.
2. Дефект (Defect) → результат ошибки, присутствующий в системе.
3. Сбой (Failure) → проявление дефекта во время выполнения программы.

Пример:

- Разработчик неправильно реализовал формулу для расчета налога (ошибка).
- В коде появился неверный расчет налогового значения (дефект).
- Пользователь замечает, что система выдает неверную сумму налога при определенных входных данных (сбой).

Причины возникновения ошибок в разработке ПО:

1. **Человеческий фактор:**
    - Ошибки ввода данных или опечатки в коде.
    - Неверное понимание задачи или требований.
    - Пропуск важных деталей при тестировании.
2. **Нехватка времени:**
    - Сжатые сроки приводят к тому, что разработчики и тестировщики работают в условиях стресса и торопятся.
    - Не успевают провести достаточное тестирование.
3. **Недостаток квалификации участников:**
    - Разработчики не знают всех возможностей используемого языка программирования.
    - Новички допускают простые ошибки из-за отсутствия опыта.
4. **Недопонимание в команде:**
    - Неверно переданные требования заказчика.
    - Отсутствие четкой документации.
    - Конфликты между участниками, приводящие к разным подходам к решению одной задачи.
5. **Сложность системы:**
    - Использование сложных архитектурных решений, которые трудно понять и поддерживать.

- Проблемы с интеграцией большого количества модулей.
- Ошибки в проектировании алгоритмов.

6. **Непонимание интерфейсов:**
   - Неправильная работа между API и другими системами из-за неверной интерпретации документации.
   - Несогласованность форматов данных.
7. **Использование новых технологий:**
   - Команда впервые работает с новым инструментом или фреймворком.
   - Недостаток знаний о правильной настройке и использовании технологии.

Примеры различных типов дефектов, которые могут быть выявлены в процессе тестирования:

1. Функциональные дефекты

Дефекты, связанные с тем, что система не выполняет заявленные функции или выполняет их неправильно.
Примеры:

- Кнопка «Отправить» в форме не отправляет данные на сервер.
- Поле ввода даты позволяет вводить некорректные форматы (например, текст вместо чисел).
- Пользователь не может восстановить пароль через функцию "Забыли пароль", так как письмо с инструкцией не отправляется.

2. Визуальные (UI) дефекты

Дефекты, которые касаются интерфейса и дизайна приложения, включая отображение элементов.
Примеры:

- Кнопка выходит за границы экрана на мобильном устройстве.
- Шрифты на одной странице не соответствуют стандартам дизайна (разные размеры или стили).
- Изображение логотипа компании не загружается и отображается как "битая" ссылка.

3. Логические дефекты

Ошибки, связанные с неправильной реализацией логики программы.
Примеры:

- В интернет-магазине скидка применяется дважды, если пользователь обновляет страницу.
- В калькуляторе приложение неправильно вычисляет результат при выполнении определенных операций (например, 2 × 2 + 2 = 6 вместо 8).
- При создании отчета выводятся данные за неправильный период времени.

4. Дефекты контента

Ошибки, связанные с неправильным или отсутствующим содержимым.
Примеры:

- На странице указан неверный номер телефона службы поддержки.
- В описании продукта на сайте содержатся грамматические или орфографические ошибки.
- В инструкции пользователя отсутствует шаг, необходимый для завершения процесса.

5. Дефекты удобства использования (Usability)

Ошибки, из-за которых пользователю неудобно или сложно работать с системой.
Примеры:

- Слишком маленький размер кнопок на мобильной версии сайта, что затрудняет нажатие.
- Отсутствие сообщения об ошибке, если пользователь не заполнил обязательное поле в форме.
- Сложная или не интуитивная навигация, из-за которой пользователю трудно найти нужную функцию.

📎 **Дефект (баг) — это отклонение фактического результата от ожиданий наблюдателя, сформированных на основе требований.**

Баг возникает, когда поведение системы не соответствует тому, что было заявлено в требованиях.

IT CAREER HUB

Задание для закрепления

1. Кнопка «Назад» на экране профиля не активна. Какой это тип?
   Ответ: Defect
2. Неправильная функция вызывается в коде. Какой это тип?
   Ответ: Error
3. Сбой приложения после быстрой прокрутки на главной странице. Какой это тип?
   Ответ: Failure
4. Пользователь не может перейти на сайт партнера по ссылке. Какой это тип?
   Ответ: Failure
5. Ссылка на пользовательское соглашение не кликабельна. Какой это тип?
   Ответ: Defect
6. Низкое качество фотографии, когда пользователь открывает ее в полноэкранном режиме. Какой это тип?
   Ответ: Defect
7. Буквы кириллицы отображаются в адресной книге в виде символов. Какой это тип?
   Ответ: Defect

IT CAREER HUB

Структура и основные поля баг-репорта

Баг-репорт — это документ или запись, описывающая найденный дефект в программном обеспечении. Хороший баг-репорт должен быть понятным, полным и лаконичным, чтобы команда разработки могла эффективно воспроизвести и исправить проблему.

1. Заголовок (Title)

- **Описание:** Краткое название бага, которое отражает его суть.
- **Пример:**
  "Кнопка «Назад» не работает на странице профиля".

2. Шаги для воспроизведения (Steps to Reproduce)

- **Описание:** Пошаговая инструкция, как воспроизвести баг. Включает подробности, начиная с открытия приложения и заканчивая выполнением действий, которые вызывают дефект.
- **Пример:**
  1. Открыть приложение.
  2. Перейти в профиль пользователя.
  3. Нажать кнопку «Назад».

3. Ожидаемый результат (Expected Result)

- **Описание:** Что должно было произойти, если бы система работала корректно.
- **Пример:**
  Пользователь переходит на предыдущую страницу.

4. Фактический результат (Actual Result)

- **Описание:** Что на самом деле произошло в результате выполнения шагов.
- **Пример:**
  Кнопка «Назад» не реагирует на нажатие.

5. Среда (Environment)

- **Описание:** Описание системы, на которой был обнаружен баг. Это может включать:
  - Версию операционной системы.
  - Версию приложения.

    - Устройство.
    - Браузер (если это веб-приложение).
- **Пример:**
    - ОС: Android 12.
    - Версия приложения: 1.2.3.
    - Устройство: Samsung Galaxy S21.

6. Приоритет (Priority)

- **Описание:** Указывает на важность исправления бага с точки зрения бизнеса. Например:
    - **Высокий (High):** Блокирующий дефект, который сильно влияет на функциональность.
    - **Средний (Medium):** Ошибка, влияющая на важную функцию, но не блокирующая её.
    - **Низкий (Low):** Косметическая или малозаметная ошибка.

7. Серьезность (Severity)

- **Описание:** Уровень влияния дефекта на функциональность. Например:
    - **Blocker:** Невозможно работать с приложением.
    - **Critical:** Основная функция не работает.
    - **Major:** Заметный сбой, но есть обходные пути.
    - **Minor:** Незначительная ошибка.
    - **Trivial:** Едва заметный дефект (например, орфографическая ошибка).

8. Скриншоты или видео (Attachments)

- **Описание:** Визуальное подтверждение бага. Это могут быть скриншоты или видео, показывающие ошибку.
- **Пример:** Скриншот неактивной кнопки «Назад».

9. Дополнительная информация (Additional Information)

- **Описание:** Любая другая информация, которая может помочь в диагностике и исправлении бага. Например:
    - Логи.
    - Наличие сетевого подключения.
    - Время возникновения дефекта.

10. ID бага (Bug ID)

- **Описание:** Уникальный идентификатор для отслеживания бага в системе управления задачами.

- **Пример:** BUG-12345.

Пример полного баг-репорта:

- **Заголовок:** "Кнопка «Назад» не работает на странице профиля".
- **Шаги для воспроизведения:**
  - Открыть приложение.
  - Перейти в профиль пользователя.
  - Нажать кнопку «Назад».
- **Ожидаемый результат:** Пользователь возвращается на предыдущую страницу.
- **Фактический результат:** Кнопка «Назад» не реагирует.
- **Среда:**
  - ОС: Android 12.
  - Версия приложения: 1.2.3.
  - Устройство: Samsung Galaxy S21.
- **Приоритет:** Высокий.
- **Серьезность:** Major.
- **Скриншоты:** Прикреплены.
- **Дополнительная информация:** Ошибка наблюдается только на Android, на iOS проблема не воспроизводится.

IT CAREER HUB

Задание для закрепления

Какой из баг-репортов описан хорошо, а какой плохо?

Баг-репорт 1:

Заголовок:
Ошибка загрузки изображений

Описание:
Иногда изображения не загружаются.

Шаги для воспроизведения:

1. Зайти в галерею.
2. Попробовать загрузить изображение.

Ожидаемое поведение:
Изображение загружается.

Фактическое поведение:
Изображение не загружается.

Скриншоты:
(нет)

Окружение:

- Браузер: Google Chrome
- ОС: Windows 10

Баг-репорт 2:

Заголовок:
[BUG] Загрузка изображений зависает в галерее профиля при использовании JPEG

Описание:
При попытке загрузить изображение формата JPEG через галерею профиля загрузка зависает. Проблема возникает только при загрузке файлов размером более 2 МБ.

Шаги для воспроизведения:

1. Авторизоваться на сайте.
2. Перейти в раздел "Галерея" в профиле.

3. Нажать "Добавить изображение".
4. Выбрать файл формата JPEG размером 2 МБ или больше.
5. Нажать "Загрузить".

Ожидаемое поведение:
Изображение загружается без ошибок, и оно появляется в галерее.

Фактическое поведение:
Загрузка зависает, а изображение не появляется в галерее.

Скриншоты/видео:
Приложен скриншот с зависшей загрузкой (иконка загрузки крутится).

Окружение:

- Браузер: Google Chrome (версия 118.0.5993.88)
- ОС: Windows 10 Pro (22H2)
- Сеть: Wi-Fi, стабильное соединение (скорость 50 Мбит/с).

Ответ: первый баг-репорт плохой, второй - хороший.

Что было не так в первой версии?

1. **Заголовок** – не уточнено, где и при каких условиях возникает проблема. "Ошибка загрузки изображений" слишком общее название.
2. **Описание** – слишком краткое. Фраза "иногда изображения не загружаются" не дает представления о проблеме.
3. **Шаги для воспроизведения** – не хватает точности. Например, не указаны конкретные действия (какой формат изображения, его размер).
4. **Фактическое поведение** – нет уточнения, что именно происходит ("не загружается" слишком размыто, важно описать, что происходит вместо загрузки).
5. **Скриншоты/видео** – их отсутствие снижает шансы быстро понять и воспроизвести проблему.
6. **Окружение** – не указана версия браузера или состояние сети, которые могут быть важны.

Эта структура помогает эффективно и точно описать баг, чтобы разработчики могли быстро разобраться в проблеме и ее устранить.

Правила написания баг-репортов

Пример описания баг-репорта:

Bug report pattern

1. **Do not assume all the companies have the same approach to writing bug reports**
   Не думайте, что во всех компаниях сообщения об ошибках пишут одинаково.
2. **Rule of WWW – What happened, Where it happened, under Which circumstances**
   Хорошее сообщение об ошибке содержит информацию о том: Что случилось? Где это случилось? При каких обстоятельствах (если они есть).
3. **“Problem” bug report versus “Solution” bug report**
   Сообщение о РЕШЕНИИ более ценно, чем о ПРОБЛЕМЕ. НО, если вы не уверены в решении на 100%, то сообщайте о проблеме.
4. **Bug report is not about perfect English**
   Для написания сообщений об ошибках на английском не требуется хорошего знания языка.
5. **Before reporting a bug, make sure that you are using the LATEST version of the App**
   Убедитесь, что вы находите ошибки в САМОЙ ПОСЛЕДНЕЙ версии тестируемого приложения.
6. **Report a bug immediately, do not postpone**
   Сообщение об ошибке пишется сразу, не откладывая.
7. **Make sure the bug is reproducible before reporting**
   Прежде чем написать сообщение об ошибке, убедитесь, что она воспроизводится в других условиях.
8. **Minimize number of steps-to-reproduce**
   Минимизируйте (по возможности) количество действий по воспроизведению ошибки.
9. **Write one bug report for each fix to be verified**
   Отдельное сообщение для каждой ошибки, требующей последующей проверки (починили или не починили).
10. **The difference between actual and expected results should be clear**
    Bug report должен ЧЕТКО показать РАСХОЖДЕНИЕ между ожидаемым и реальным поведением системы.
11. **Do not quote the violated rules/requirements (developers know them) – just talk about the problem itself**
    Программист, в основном, знает правила, не нужно на них ссылаться, если они очевидны.

12. **Do not assume developer knows less than you do about the application**
    Не думайте, что программист знает меньше вас о работе тестируемого приложения.
13. **Bug reports should be as concise as possible**
    Сообщение об ошибке должно быть предельно лаконичным.
14. **Bug report should be as complete as possible**
    Сообщение об ошибке должно содержать ПОЛНУЮ информацию, относящуюся к ошибке.
15. **Attach screenshots, data files, logs to clarify the bug description**
    Используйте скриншоты, видео, файлы и прочее для иллюстрации того, что произошло.
16. **Each “problem” has a story (each decision is a compromise) – research before reporting**
    Не спешите критиковать странные вещи на новом месте — спросите, почему сделано так.
17. **Use technical terms, not “people off the street” language**
    Пользуйтесь терминологией, избегайте непрофессиональной лексики.

IT CAREER HUB

Жизненный цикл бага

Жизненный цикл бага (Bug Life Cycle) описывает этапы, через которые проходит дефект в программном обеспечении, начиная с момента его обнаружения и заканчивая окончательным исправлением и проверкой.

Основные этапы жизненного цикла бага включают:

1. Новый (New)

- **Описание:** Дефект только что обнаружен и зарегистрирован тестировщиком в системе управления задачами (например, JIRA, Bugzilla).
- **Действия:**
  - Баг описывается и оформляется с указанием всех необходимых данных.
  - Он ожидает проверки.

2. Открыт (Open)

- **Описание:** Руководитель тестирования или разработчик подтверждает, что баг является валидным и требует исправления.
- **Действия:**
  - Проверка бага: действительно ли это ошибка.
  - Определение приоритета и серьезности.
  - Передача задачи разработчику.

3. В работе (In Progress/Assigned)

- **Описание:** Баг назначен конкретному разработчику для исправления.
- **Действия:**
  - Разработчик начинает работать над устранением дефекта.
  - Возможны уточнения и запросы дополнительной информации.

4. Исправлен (Fixed)

- **Описание:** Разработчик устранил проблему и обновил статус дефекта.
- **Действия:**
  - Код с исправлением заливается в соответствующую ветку.
  - Задача передается обратно тестировщикам для проверки.

5. На проверке (Retesting)

- **Описание:** Тестировщик проверяет исправленный баг, воспроизводя шаги, описанные в баг-репорте.
- **Действия:**
    - Если баг больше не воспроизводится, он переходит на следующий этап.
    - Если баг всё ещё воспроизводится, он переводится обратно в статус **"Открыт"** или **"Переоткрыт"**.

6. Закрыт (Closed)

- **Описание:** Тестировщик подтверждает, что дефект исправлен, и проблема больше не воспроизводится.
- **Действия:**
    - Баг закрывается в системе отслеживания.

7. Переоткрыт (Reopened)

- **Описание:** Если тестировщик обнаруживает, что баг все еще присутствует или исправление вызвало новые проблемы, баг переводится в статус **"Переоткрыт"**.
- **Действия:**
    - Баг снова отправляется разработчику для исправления.

8. Отклонен (Rejected)

- **Описание:** Руководитель проекта или разработчик считает, что зарегистрированный баг не является дефектом.
- **Причины отклонения:**
    - Некорректная интерпретация требований.
    - Поведение системы соответствует спецификации.

9. Не воспроизводится (Not a Bug/Not Reproducible)

- **Описание:** Баг не может быть воспроизведен на указанной среде.
- **Действия:**
    - Возможное уточнение шагов или данных от тестировщика.
    - Баг может быть закрыт или доработан.

10. Отложен (Deferred)

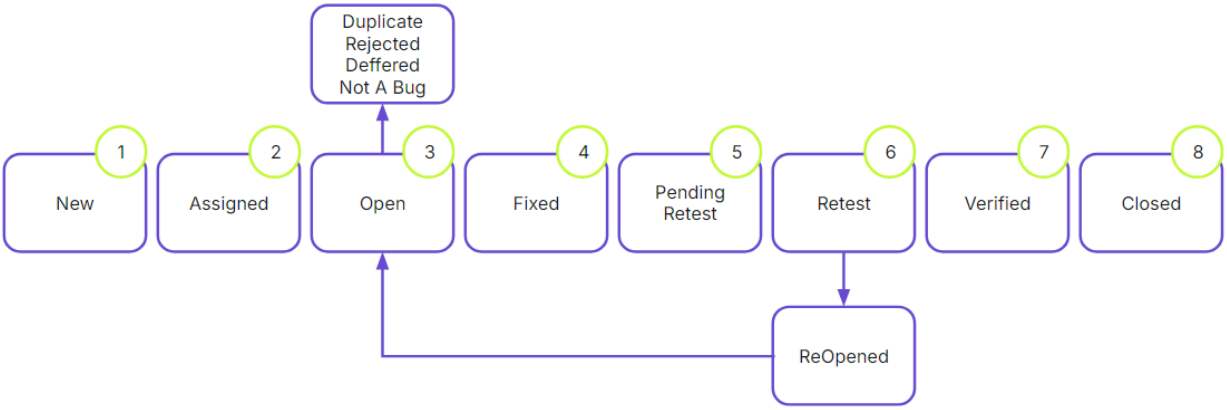
- **Описание:** Баг признан валидным, но его исправление откладывается.
- **Причины:**
    - Низкий приоритет.
    - Баг не критичен для текущего релиза.

11. Дубликат (Duplicate)

- **Описание:** Обнаружено, что баг уже был зарегистрирован ранее.
- **Действия:**
    - Новый баг закрывается как дубликат, и работа продолжается по исходной задаче.

Пример потока:

1. Баг зарегистрирован тестировщиком (**New**).
2. Подтверждён руководителем и назначен разработчику (**Open → In Progress**).
3. Исправлен разработчиком (**Fixed**).
4. Протестирован тестировщиком (**Retesting**).
5. Если исправление успешно, баг закрывается (**Closed**).
6. Если баг всё ещё воспроизводится, он переоткрывается (**Reopened**) и отправляется на доработку.



IT CAREER HUB

Задание для закрепления

1. Какие компоненты включаются в баг-репорт?
2. Какие поля вы заполняете в баг-репорте как QA-инженер?
3. Если вы находите баг, а разработчик утверждает, что это "так задумано", что вы можете сделать?
4. Какая часть баг-репорта является самой важной?
5. Каков жизненный цикл бага?
6. Как тестировщик может убедиться, что баг был исправлен?

Ответы:

1. Какие компоненты включаются в баг-репорт?

Компоненты баг-репорта включают:

- Заголовок (Title).
- Шаги для воспроизведения (Steps to Reproduce).
- Ожидаемый результат (Expected Result).
- Фактический результат (Actual Result).
- Среда (Environment).
- Приоритет (Priority).
- Серьезность (Severity).
- Вложения (Attachments).
- Дополнительная информация (Additional Information).

2. Какие поля вы заполняете в баг-репорте как QA-инженер?

QA-инженер заполняет следующие поля:

- Название (что за ошибка).
- Шаги для воспроизведения.
- Описание ошибки (разница между ожидаемым и реальным результатом).
- Тестовая среда (устройство, ОС, версия браузера или приложения).
- Скриншоты или логи для подтверждения.
- Приоритет и серьезность ошибки.

3. Если вы находите баг, а разработчик утверждает, что это "так задумано", что вы можете сделать?

- Пересмотреть требования или спецификацию, чтобы проверить, соответствует ли текущее поведение документации.
- Связаться с аналитиком или владельцем продукта (Product Owner) для уточнения.
- Если требования не ясны, поднять вопрос на командном обсуждении.

4. Какая часть баг-репорта является самой важной?

Самая важная часть баг-репорта — это шаги для воспроизведения (Steps to Reproduce). Если баг нельзя воспроизвести, его будет сложно устранить. Также важна разница между ожидаемым и фактическим результатом.

5. Каков жизненный цикл бага?

Жизненный цикл бага включает следующие этапы:

1. Новый (New) — баг зарегистрирован.
2. Подтвержден (Open) — валидность бага подтверждена.
3. Назначен (Assigned) — назначен разработчику.
4. Исправлен (Fixed) — разработчик устранил баг.
5. На проверке (Retesting) — баг проверяется тестировщиком.
6. Закрыт (Closed) — баг подтверждён как исправленный.
7. Переоткрыт (Reopened) — баг всё еще воспроизводится (при необходимости).
8. Отклонен (Rejected) — баг признан не валидным.

6. Как тестировщик может убедиться, что баг был исправлен?

- Воспроизвести описанные шаги в баг-репорте и убедиться, что проблема больше не возникает.
- Провести регрессионное тестирование, чтобы убедиться, что исправление не повлияло на другие части системы.
- Проверить изменения в коде, если доступна документация на патч или pull request.
- Протестировать баг в разных средах и с различными данными.

IT CAREER HUB

Практическая работа

Заводим баг репорты (минимум 1) в qase.io для https://www.saucedemo.com/ , авторизуясь под пользователем **problem_user**

Баг-репорт 1:

Заголовок: Некорректное отображение изображений товаров при входе под пользователем "problem_user"

Шаги для воспроизведения:

1. Перейти на сайт https://www.saucedemo.com.
2. Ввести логин: problem_user.
3. Ввести пароль: secret_sauce.
4. Нажать кнопку "Login".
5. Обратить внимание на изображения товаров на странице "Products".

Ожидаемый результат: Все изображения товаров отображаются корректно и соответствуют описанию каждого товара.

Фактический результат: Некоторые изображения товаров не загружаются или отображаются некорректно (например, отсутствуют или заменены на заглушки).

Среда:

- Браузер: Google Chrome версии 96.0.4664.45
- Операционная система: Windows 10 Pro

Приоритет: Средний

Серьезность: Средняя

Вложения: Скриншоты страницы "Products" с некорректно отображаемыми изображениями товаров.

Дополнительная информация: Проблема наблюдается только при входе под пользователем problem_user. При использовании других учетных записей (например, standard_user) изображения товаров отображаются корректно.

Баг-репорт 2:

Заголовок:

Кнопка "Remove" неактивна после добавления товара в корзину.

Шаги для воспроизведения:

1. Перейти на сайт https://www.saucedemo.com.
2. Авторизоваться под пользователем:
    - **Логин:** problem_user
    - **Пароль:** secret_sauce.
3. На странице "Products" нажать кнопку **"Add to cart"** у любого товара.
4. Нажать кнопку **"Remove"** для того же товара.

Ожидаемый результат:

После нажатия кнопки **"Remove"** товар должен быть удален из корзины, и кнопка должна измениться на **"Add to cart"**.

Фактический результат:

- При нажатии кнопки **"Remove"**:
    - Кнопка остаётся в неактивном состоянии.
    - Товар не удаляется из корзины.
    - Иногда кнопка остаётся визуально в статусе **"Remove"**, но товар все равно добавлен в корзину.

Среда:

- **Браузер:** Google Chrome 96.0.4664.45
- **Операционная система:** Windows 10 Pro
- **Устройство:** PC

Приоритет:

Высокий

Серьезность:

Критическая

Вложения:

- Скриншоты корзины, где товар остается добавленным, несмотря на нажатие **"Remove"**.
- Видео демонстрации попытки удалить товар из корзины.

Дополнительная информация:

Проблема воспроизводится только под пользователем problem_user. Для других пользователей (например, standard_user) кнопка работает корректно.