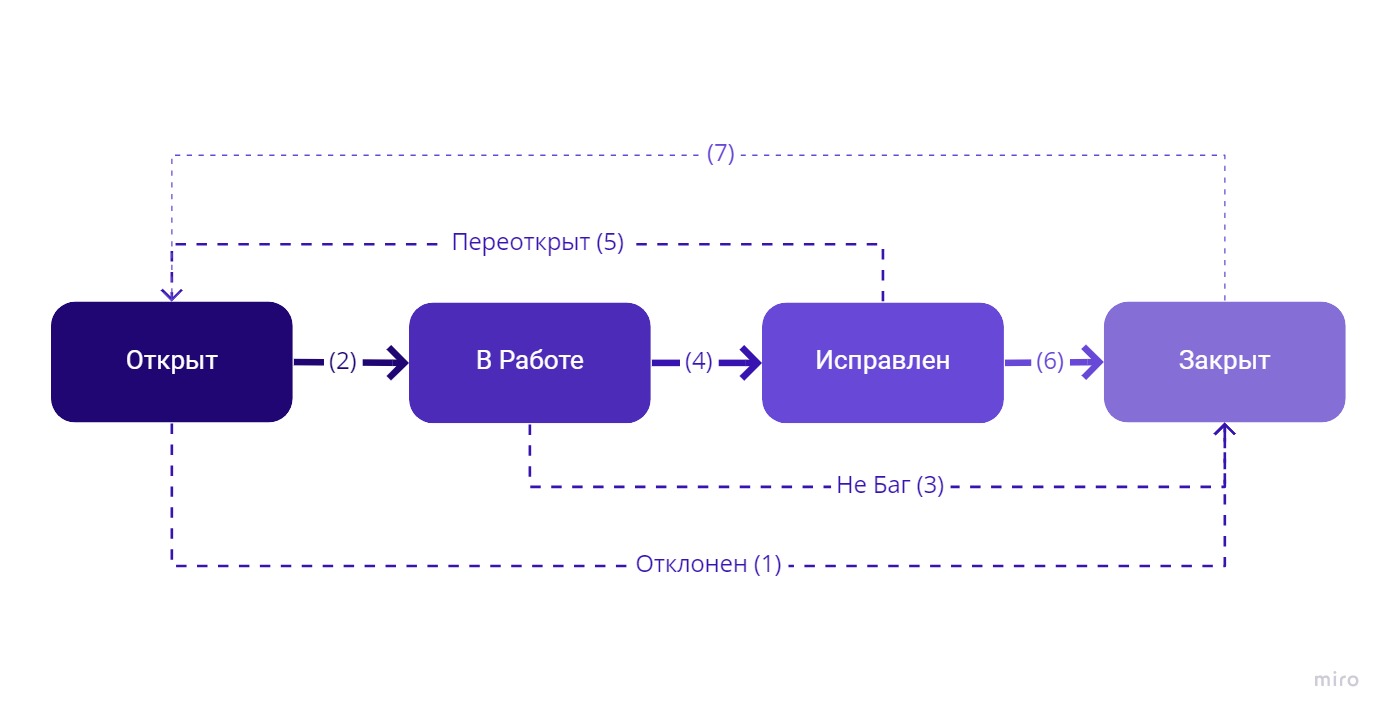
**Лабораторная работа №3**

**Тестирование ПО**При тестировании возникает необходимость документирования найденных дефектов. Это позволяет исправить их в кратчайшие сроки.

Если кратко, то хороший баг-репорт позволяет:

* воспроизвести проблему;
* понять, в чем проблема, и какова ее важность.



### **Что такое баг, типы багов**

По версии международной комиссии по сертификации тестирования программного обеспечения (ISTQB), баг(дефект) - изъян в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию. Например, неверный оператор или определение данных может привести к отказам компонента или системы.

### **Критичность и приоритет бага. Атрибуты баг-репорта**

Необходимо рассказать о приоритете и критичности бага. Важно понимать разницу между ними. Примеры приведу из личного опыта. Объектом тестирования в моей работе является ПО приёмников цифрового телевидения. Главной задачей ПО приёмника является расшифровка контента, передаваемого в зашифрованном виде. Для успешной расшифровки абонент должен приобрести у оператора подписку на соответствующий пакет телеканалов.

**Критичность бага** – это атрибут, который характеризует влияние бага на общую функциональность разрабатываемого ПО.

**ПО КРИТИЧНОСТИ БАГИ ДЕЛЯТ НА:**

**S1. Блокирующий (Blocker)**. Всё тестируемое ПО не может работать без устранения бага. Например, приёмник начинает перезагружаться сразу после включения, мы не сможем больше ничего протестировать из-за этого бага.

**S2. Критический (Critical).** Большая часть ПО не может корректно работать. Например, приёмник не может открывать закодированные каналы. До устранения этого дефекта можно протестировать UI, а также функционал, не связанный с расшифровыванием каналов.

**S3. Значительный (Major).** Блокирует работу одной из основных логических цепочек ПО. Например, неправильное сообщение об ошибке при отсутствии подписки на пакет оператора.

**S4. Незначительный (Minor).** Не нарушает основные логические цепочки приложения, с ним можно продолжать работать почти без потери качества. Здесь можно привести неточный перевод с русского на английский в меню приёмника.

**S5. Тривиальный (Trivial).** Эта степень присваивается, когда баг вообще не влияет на общее качество работы ПО. Например, незначительное пересечение элементов в меню.

ПРИОРИТЕТ БАГА — это то, в каком порядке нужно решать проблемы. Существует три степени приоритетности:

**P1. Высокий приоритет (High).** Нужно исправить немедленно, потому что баг является крайне важным для всего релиза. Например, старое сообщение об отсутствии подписки на пакет, хотя обновление текстов являлось целью этого релиза.

**P2. Средний приоритет (Medium).** Точно нужно будет исправить, баг достаточно важен, но не требует немедленного решения. Например, некорректный перевод в меню приёмника.

**P3. Низкий приоритет (Low).** Нужно будет исправить, но баг не очень важный и не требует немедленного решения. Например, это могут быть баги в функционале, который уже не используется оператором, но ещё не был удалён из кода.

**Что такое баг репорт, его типичная структура:**

Глоссарий ISTQB говорит, что баг репорт — это документ, содержащий отчет о любом недостатке в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию.

Но можно сказать и проще: баг репорт (bug report) – это технический документ, который содержит в себе полное описание бага, включающее информацию как о самом баге (краткое описание, критичность, приоритет и т.д.), так и об условиях возникновения данного бага.

**Как правильно оформить баг-репорт**

1. Для начала нужно убедиться, что найденный баг ещё не был оформлен. Следует провести поиск его в соответствующем проекте по всем подходящим ключевым словам и\или полям. Если баг уже есть, следует обновить его описание.
2. Если баг не найден – фиксируем в таблице багов. Не стоит забывать важное правило: один дефект - один баг в трекере.
3. Далее нужно постараться описать, что не работает - это и будет заголовок баг-репорта.
4. После этого перейти к подробному описанию бага: указать шаги к воспроизведению.
5. Указать ожидаемый результат. Можно добавить ссылку на спецификацию.
6. Указать полученный результат.
7. Указать версию ПО, также указать версию окружения.
8. Если необходимо, приложить соответствующие артефакты: логи, скриншоты, дампы и т.д.

**Пример состава 1-го баг репорта приведен в таблице:**

|  |  |
| --- | --- |
| Заголовок (Summary) | Короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации. |
| Проект (Project) | Название тестируемого проекта |
| Компонент приложения (Component) | Название части или функции тестируемого продукта |
| Номер версии (Version) | Версия, на которой была найдена ошибка |
| Окружение (Environment) | Модель устройства, версия операциорнной системы, характеристики на котором происходит тест |
| Критичность  (Severity) | Наиболее распространена пятиуровневая система критичности:  S1 Блокирующий (Blocker)  S2 Критический (Critical)  S3 Значительный (Major)  S4 Незначительный (Minor)  S5 Тривиальный (Trivial) |
| Приоритет (Priority) | Приоритет дефекта:  P1 Высокий (High)  P2 Средний (Medium)  P3 Низкий (Low) |
| Статус (Status) | Статус бага. Зависит от используемой процедуры и жизненного цикла бага. Например:   * Новый * Открыт * Закрыт |
| Автор (Author) | Создатель баг репорта |
| Назначен на (Assigned To) | Имя сотрудника, назначенного на решение проблемы |
| Описание (Description) | Информация об окружении, на котором был найден баг: операционная система, сервис пак, имя и версия браузера, версия ПО чипа, версия библиотеки  и т.д. |
| Шаги, по которым можно легко воспроизвести ситуацию, приведшую к ошибке. |
| Полученный результат |
| Ожидаемый результат |
| Прикрепленный файл (Attachment) | Файл с логами, скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на способ решения проблемы |

**Задание 1 – Выписать теорию в тетрадь и ответить на контрольные вопросы устно.**

Контрольные вопросы:

- Что такое баг, типы багов.

- Что такое критичность и приоритет бага. Атрибуты баг-репорта.

- По критичности баги делят на:

- Что такое приоритет бага.

- Что такое баг репорт, его типичная структура

- Что такое Заголовок (Summary)

- Что такое Статус бага

- Типы багов:

- Что такое SEO?

- Отличие фактического результата от ожидаемого

- Что такое серьезность?

- Что такое окружение бага

**Задание 2 – По скрину главного вида игрового приложения (рисунок 1) версия 1.5.8 найти и описать баг-репорты по форме-таблице**

# 

Рисунок 1 – Окно игрового приложения

Для каждого найденного бага, необходимо заполнить таблицу баг-репорта, придумать Заголовок, описать критичность, приоритеты, описание и сделать скрин-шоты проблемных моментов.

|  |  |
| --- | --- |
| Заголовок (Summary) |  |
| Проект (Project) |  |
| Дата начала проверки (Scan start date) |  |
| Компонент приложения (Component) |  |
| Номер версии (Version) |  |
| Окружение (Environment) |  |
| Критичность  (Severity) |  |
| Приоритет (Priority) |  |
| Статус (Status) |  |
| Автор (Author) |  |
| Назначен на (Assigned To) |  |
| Описание (Description) |  |
| Прикрепленный файл (Attachment) |  |

# Задание 3 – Тестирование сайта на предмет наличие багов и заполнить таблицу баг-репорта

Перейдите по ссылке - https://www.demoblaze.com/index.html

Приведенные ниже типы багов относятся к WEB сайтам.

**UI (ошибка в верстке, то есть в составлении шаблона сайта)**

Баг в верстке — следствие ошибки в разметке (HTML) или стилизации (CSS) элемента страницы в специфическом окружении.

Примеры:

1. Не отображается блок Х на странице Y (в дизайне блок есть, на странице — нет)
2. Неправильное расположение блока на странице X (в дизайне блок слева, на странице — справа)
3. Не переносится на новую строку / Не обрезается текст ссылки “Our TOP 20 projects” в блоке «How it works?» на странице Х

**UX (ошибка в удобстве)**

Баг в удобстве — неудобство / нелогичность работы с элементами / функционалом страницы.

Примеры:

1. Не получается с первого раза нажать на кнопку Х в футере на мобильном (очень маленькая зона клика, кнопку нужно сделать больше)
2. Удаляется заказ после нажатия на кнопку Х в модальном окне на странице Б (ожидаешь закрытия окна, а фактически удаляется заказ — UX путает)

**Functional (ошибка в функционале сайта)**

Баг в функционале — несоответствие логики работы компонента заявленным функциональным требованиям.

Примеры:

1. Отображается неправильное количество ссылок в блоке Related Papers в sidebar
   1. требование: выводить 5 ссылок
   2. фактически: выводится 10 ссылок
2. Не происходит прокрутка страницы вверх после нажатия на кнопку To Top
   1. требование: происходит прокрутка страницы вверх после нажатия на кнопку To Top
   2. фактически: ничего не происходит
3. Не показалось сообщение об ошибке при вводе числа в поле Name
   1. требование: допустимые символы для поля Name = буквы (обязательны) + пробелы (не обязательны). При вводе других символов — показываем сообщение об ошибке.
   2. фактически: сообщение об ошибке не отображается
4. Не отображается модальное окно А после нажатия на кнопку Х
   1. требование: после нажатия на кнопку X показывается окно А
   2. фактически: после нажатия на кнопку X показывается окно С
5. Не отображается текст “Нет заказов” на профиле райтера, если количество заказов, назначенных райтеру = 0
   1. требование: отображается текст “Нет заказов“, если количество заказов на профиле райтера = 0
   2. фактически: не отображается текст “Нет заказов“, если количество заказов на профиле райтера = 0

**SEO (ошибка в seo)**

Баг в seo — ошибка, которая влияет на SEO (нарушение нефункциональных требований, касающихся seo). SEO (англ. Search Engine Optimization) – это **комплекс мер по улучшению сайта для его ранжирования в поисковых системах**.

Примеры:

1. Отображается неправильная структура заголовков блоков на странице Х
2. Найдены 4 ошибки в коде на странице Х после проверки
3. Указан неправильный title на странице Х
4. Закрыта для индексации страница Х
5. Отсутствует атрибут ALT на изображении Z на странице Х

**Алгоритм создания баг репорта для WEB-сайтов (пункты для таблицы отчета)**

Ниже приведен алгоритм описания:

1. Понять “суть” проблемы, а не ее проявление (если получится, но это требует технических знаний)
2. Воспроизвести дефект один-два раза (удостовериться, что он повторяется)
3. Проверить наличие найденного вами дефекта (возможно, баг уже создали или он повторяется)
4. Написать заголовок (отвечает на вопросы “что? где? когда?”)
5. Написать основные поля отчета

* **ID** — уникальный идентификатор бага
* **Заголовок / Краткое описание / Тема / Summary / Title** — четко и кратко описывает суть бага. Оформляется в виде одного предложения, состоящего из трех частей отвечающих на вопросы “Что? Где? Когда?”. Редко бывает, что ответ на вопрос “Где?” или “Когда?” может опускаться, если он не дает полезной информации.
* **Шаги к воспроизведению** — **четкое, последовательное**описание шагов / действий, которые необходимо совершить, чтоб воспроизвести баг со всей необходимой информацией
* **Фактический результат** — результат, который мы **видим**
* **Ожидаемый результат —**результат, который мы **хотели / ожидали увидеть**
* **Серьезность —**показывает, насколько серьезные последствия от дефекта с точки зрения влияния на систему

1. Заполнить дополнительные поля отчета

* **Скриншот / видео** — изображение / видео, которое четко и наглядно демонстрирует баг. Если видео или скриншот сделан качественно, его может быть достаточно для понимания сути ошибки и ее исправления
* **Требование —**ссылка на требование, которое не соблюдено. Наличие этой информации в 99% случаев предотвращает разговор “баг — не баг” и испорченное настроение 🙂
* **Тип бага —**для анализа “слабых” мест в ПО, баги могут разделять на типы (см. выше примеры для WEB)
* **Приоритет —**очередь, в которой баг будет исправляться (Высокий -> Средний -> Низкий)
* **Дополнительные файлы —**файлы, которые нужны для воспроизведения бага (файлы определенного размера, типа, логи и т.п.)
* **Окружение —**информация об окружении, на котором воспроизводится баг (версия браузера, операционная система, размер экрана, тестовый сервер и т.п.)
* **Статус —**текущий статус бага в его жизненном цикле (Открыт, В работе…)
* **Автор —**человек, который создал баг (нужен для уточнения информации, если потребуется)
* **Исполнитель —**человек, которые работает над багом в данный момент времени
* **Комментарии —**обсуждение исправления ошибки
* **Версия —**версия ПО, в которой был обнаружен баг
* **Версия исправления —**версия ПО, в которую будет добавлено исправление бага

1. Внести все данные в таблицу/Сохранить отчет.

Таблица приведена ниже:

Таблица для заполнения багов WEB-сайта:

|  |  |
| --- | --- |
| ID: |  |
| Заголовок: |  |
| Шаги к воспроизведению: |  |
| Фактический результат: |  |
| Ожидаемый результат: |  |
| **Серьезность:** |  |
| **Скриншот:** |  |
| **Требование:** |  |
| **Тип бага:** |  |
| Приоритет: |  |
| **Дополнительные файлы:** |  |
| **Окружение:** |  |
| Статус: |  |
| Автор: |  |
| Исполнитель: |  |
| Комментарий: |  |
| Версия: |  |
| Версия исправления: |  |