

# Курс Тестировщик ПО

---

## Блок2 Тестирование Mobile, Desktop, Web

**бруноям**

# Структура:

---

- Десктоп
- Мобильные приложения
- Веб-приложения
- Тестовые артефакты

# Какие приложения бывают:

- **Десктопные (Настольные)**

Программы, логика которых требует наличие оператора, содержащие в себе всю полную функциональность и способные работать отдельно на любой машине изолированно от других приложений.

- **Мобильные**

Тестируется не девайс, а приложения, установленные на телефон/планшет под управлением ОС

- **Веб**

Приложения, которые требуют выхода в интернет, являются кроссплатформенными, интерактивны и чаще всего имеют клиент-серверную архитектуру

# Десктопные

---

- Не требуют подключения к интернету
- Для взаимодействия с пользователем имеют интерфейс
- Работают довольно быстро
- Могут быть установлены на любой компьютер

## Виды тестирования:

- Тестирование установки
- Тестирование удаления
- Тестирование обновлений

# О чем важно помнить

- Разные ОС
- Разрядность
- Периферия (мыши, клавиатуры итд.)
- Цветовая схема
- Полноэкранный режим/режим небольшого окна
- Выход в интернет

# Мобильные приложения

## Нативные

- Загружаются через магазины приложений
- Разрабатываются под конкретную систему
- Им доступны все функции смартфона
- Потребление ресурсов этими приложениями более оптимизировано
- Могут работать без интернета (если не связаны с функциональностью)

## WEB

- Сайты, которые мы используем на мобильных устройствах (адаптированные версии)
- Кроссплатформенные
- Не нужна спец.разработка, но нужен интернет

## Гибридные

- Скачиваются из магазина приложений
- Вся логика в себе
- Обязательно нужен интернет

# Мобильные приложения:

---

- На каких физических девайсах тестировать?

Опираемся на ЦА и аналитику.

- Эмуляторы и симуляторы

# Мобильные приложения:

## Особенности тестирования:

- Связь
- Прерывания
- Сенсоры
- Ориентация и жесты
- Геолокация
- Нагрузка на память устройства и батарею
- Обновление/удаление/установка
- Настройки смартфона

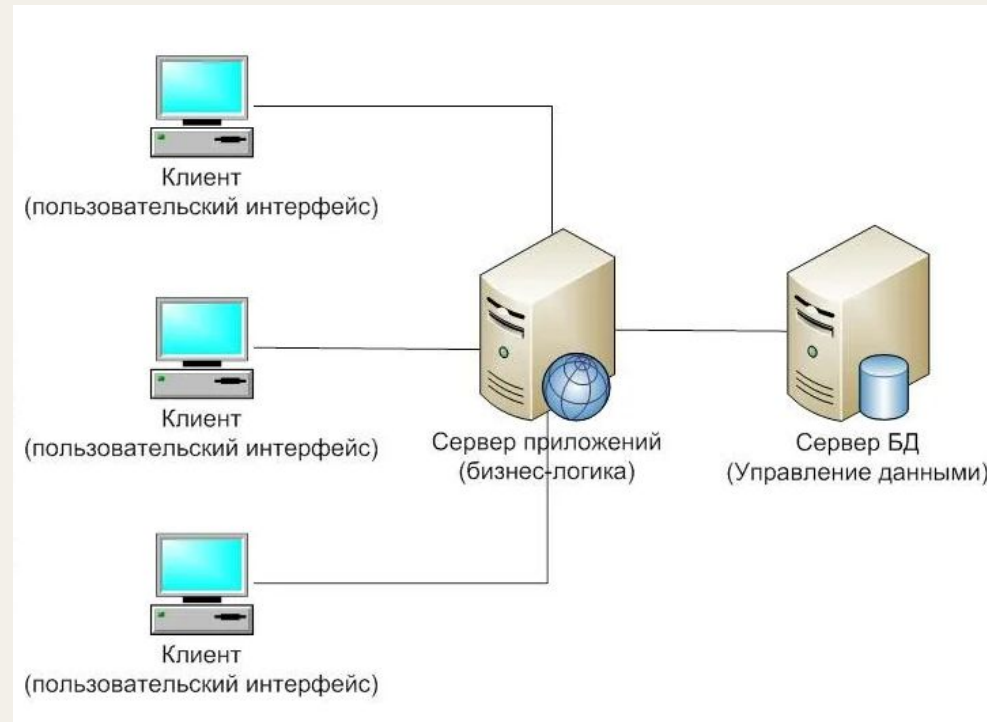
## Важно учитывать:

- Адаптивность (разные разрешения)
- Кроссбраузерность



# Веб-приложения

Веб-приложения – это приложения, которые требуют выхода в интернет, являются кроссплатформенными, интерактивными и чаще всего имеют клиент-серверную архитектуру.



# Веб-приложения

---

Сайт и Веб-приложение

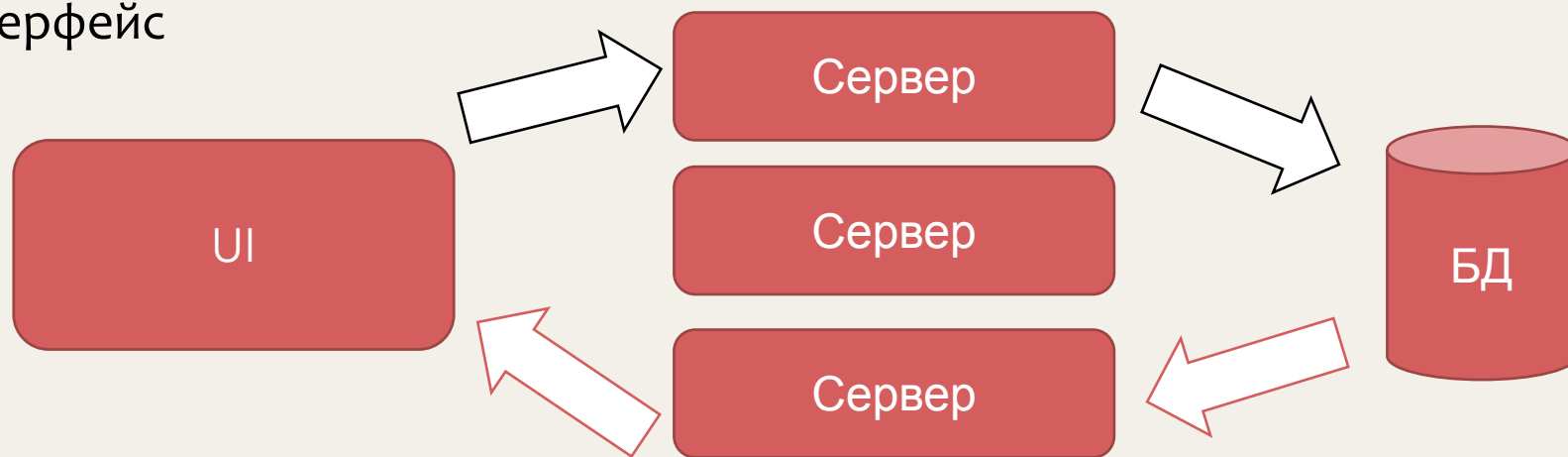
Сайт - нечто статичное, например, информационный портал города с новостями.

Добавим возможность оставлять комментарии под новостями - и это уже веб-приложение

Для удобства Сайт = Веб-приложение

# Веб-приложения. Структура

- Графический интерфейс UI
- Сетевой интерфейс



# Браузеры

---

## Основные функции:

- Загрузка страниц
- Парсинг страниц - то есть анализ текста, кода
- Рендеринг страницы - визуализация вышеупомянутого кода, создание UI
- Взаимодействие с пользователем
- Выполнение JS-кода - то есть информация может быть обработана в браузере

# Языки разметки:

---

HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки веб-страниц в браузере.

Разметка Markdown - облегченный язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций (например, HTML)

Разметка HTML – язык разметки с помощью тегов

XML - расширяемый язык разметки

# Особенности Веб-приложений:

- Тестирование разметки
- Текст
- Переходы по ссылкам
- Локализация
- Кроссбраузерность и адаптивность
- Тестирование доступов
- API и Базы данных

# Практическое задание

---

- В каком теге может быть размещен логотип сайта?
- Самые популярные мобильные браузеры.

Список особенных проверок для

Мобильного приложения с Афишей мероприятий.

# Домашка

---

## Задание 1 (normal)

Найдите приложение, которое можно установить как на ПК, так и на ваш смартфон. Если не сможете придумать - установите YouTube Music. Напишите все различия в установке и настройках, которые только заметите. Удалите приложения и снова опишите различия.

## Задание 2 (normal)

Ниже вы видите текст. В скобках после слов указано, каким образом должно быть размечено слово при помощи разметки markdown. Скопируйте текст себе в текстовый редактор и разметьте тегами/

V for Vendetta (заголовок). Нас учат помнить не человека (курсив) но идею(жирный). Ибо человек слаб. Его могут поймать, его могут убить (зачеркнуто) и предать забвению. (горизонтальная черта) Но идея и четыреста лет спустя способна изменить мир(цитата). Я из личного опыта знаю, что такое сила идеи. Я видела, как во имя идеи совершались убийства. Как люди шли на смерть(перенос строки). Но идею нельзя поцеловать... к ней нельзя прикоснуться, обнять ее(список). Идея не может истечь кровью, не чувствует боли (нумерованный список). Идея не умеет любить. (горизонтальная черта). Так что не идея причина моей скорби, а человек. Из-за него я помню пятое ноября(ссылка на википедию, что случилось 5го ноября). Его я никогда не забуду (инлайн код).



# Домашка

---

## Задание 3 (normal)

Найдите и запишите теги HTML, которые соответствую разметке markdown выше.

### Пример:

Жирный текст:

- markdown - **\*\*текст\*\***
- HTML - **текст**

Пришлите текстом в ответе к модулю.

## Задание 4 (hard)

Вы тестируете новый веб-сайт. И конечно же время на тестирование ограничено. Вам нужно протестировать его кроссплатформенность, в том числе и на мобильных устройствах. Напишите, в каких браузерах вы будете проводить тестирование, их версии и на каких операционных системах. Обоснуйте свой выбор.

Сайт нацелен на европейский, российский и китайский рынки.

Можно использовать инструмент статистика: (<https://gs.statcounter.com/>)

# Тестовая стратегия

---

Проблемы при отсутствии тестовой стратегии:

- Пропущенные дефекты
- Отсутствие прозрачности
- Приоритеты
- Человеческий фактор (личные предпочтения). Лучше, когда все задокументировано!

Тестовая стратегия определяет то, как мы тестируем продукт. Это набор мыслей и идей, которые направляют процесс тестирования.

# Тестовая стратегия

---

## Составление:

- Зачем мы тестируем? Определить цель тестирования.
- Что тестировать?
- Как тестировать?
- Когда тестировать?

## Стратегия дает понять:

- Как идет прогресс
- Какие решения были приняты
- Почему именно так
- Как расставлены приоритеты

# Тестовая стратегия

---

## Полезные книги:

- Романа Савина «Тестирование dot com»
- Святослав Куликов «Тестирование программного обеспечения. Базовый курс».
- Джеймса Уиттакера «Как тестируют в Google»

# Тест-план

---

*"Планы бесполезны, но планирование бесценно" – Дуайт Эйзенхауэр.*

**Тест план (Test Plan)** — это документ, который описывает весь объем работ по **тестированию**, начиная с описания объекта, стратегии, расписания, критериев начала и окончания **тестирования**, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков.

## Содержит:

- Критерии начала тестирования
- Критерии окончания тестирования

## Составление:

Задача --> Составить проверки (разбить на приоритеты, написать доп.проверки) --> Отправить тест-план на согласование

# Тестовая стратегия и Тест-план

---

Тестовая стратегия - это общие положения тестирования для всего отдела или компании.

Тест-план - это конкретные проверки для функциональности или задачи.

# Отчет о тестировании:

---

**Отчёт о результатах тестирования** — документ, обобщающий результаты работ по тестированию и содержащий информацию, достаточную для соотнесения текущей ситуации с тест-планом и принятия необходимых управленческих решений.

Зачастую строятся автоматически с помощью Багтрекинг-систем или Систем управления тестированием.

# Тест-кейс

---

Основной инструмент тестировщика.

- Тест-кейс - это тестовый артефакт, суть которого заключается в выполнении некоторого количества действий и/или условий, необходимых для проверки определенной функциональности разрабатываемой системы.
- Тест-сюит - набор тест-кейсов. Кейсы объединяются в сюиты по функциональностям или по модулям, по бизнес-логике и т.д.



# Чек-лист

<b>Регистрация</b>
Email
Соцсети
Валидация полей
<b>Авторизация</b>
Пользователь активирован
Пользователь не активирован <input type="checkbox"/>
Восстановление пароля
Валидация полей
<b>Профиль</b>
Изменение имени
Изменение email
Изменение пароля
Валидация полей
Подписка
Удаление аккаунта



# Чек-листы и тест-кейсы

---

Чек-лист отличается от тест-кейса степенью подробности.

- **Чек-лист** - список, содержащий ряд необходимых проверок для какой-либо работы.

В тестировании чек-лист — это список проверок для тестирования продукта.

- **Тест-кейс** - это проверка. Это такое описание проверки работы системы, которое может выполнить **любой** человек из команды, будь то тестировщик, разработчик, аналитик или даже бизнес-заказчик.

# TMS. Test Management System

---

**Системы** управления тестированием используются для хранения информации о том, как должным образом проводить тестирование, осуществление очередности проведения тестирования в соответствии с его планом, а также для получения информации в виде отчетов о стадии тестирования и качестве тестируемого продукта. Инструменты имеют различные подходы к тестированию и, таким образом, включают в себя различные наборы функций.

TestRail, TestLink, TestIT, sitechko, надстройки в JIRA.

# Багтрекинговые системы

---

Система отслеживания ошибок ([англ. bug tracking system](#)) — [прикладная программа](#), разработанная с целью помочь разработчикам [программного обеспечения](#) ([программистам](#), [тестировщикам](#) и др.) учитывать и контролировать ошибки и неполадки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий.

[JIRA](#), Redmine, Kaiten, Trello, YouTrack, HP ALM

# Test Rail

---

Регистрируемся и трогаем руками)

Тестируем тестовый интернет-магазин <http://shop.bugred.ru/>

# Правила составления Тест-кейсов:

- Обязательно указывать предусловия
- Описываем Ожидаемый результат
- Атомарность тест-кейса и шагов. Т.е один кейс = одна проверка.
- Тест-кейс не должен зависеть от других тест-кейсов
- В тест-кейсе не должно быть лишних деталей
- В тест-кейсе должна быть вся необходимая информация для его выполнения

## Практическое задание:

---

- Составляем чек-лист для проверки формы авторизации.
- Несколько (2-3) проверок из чеклиста описываем подробнее в формате тест-кейсов

# ***Баг. Ошибка. Дефект.***

---

Баг - несоответствие фактического результата выполнения программы ожидаемому.





# Правила создания бага:

---

## Атрибуты баг-репорта:

- Уникальный id
- Емкое название, из которого понятно, что/где/когда
- Предусловия (особенности, окружение)
- Шаги воспроизведения
- Фактический результат
- Ожидаемый результат
- Аттачи (картинки, видео, логи, файлы и т. д.)
- Приоритет
- Серьезность

# Severity & Priority:

---

## Приоритеты:

- Высокий
- Средний
- Низкий

## Серьезность:

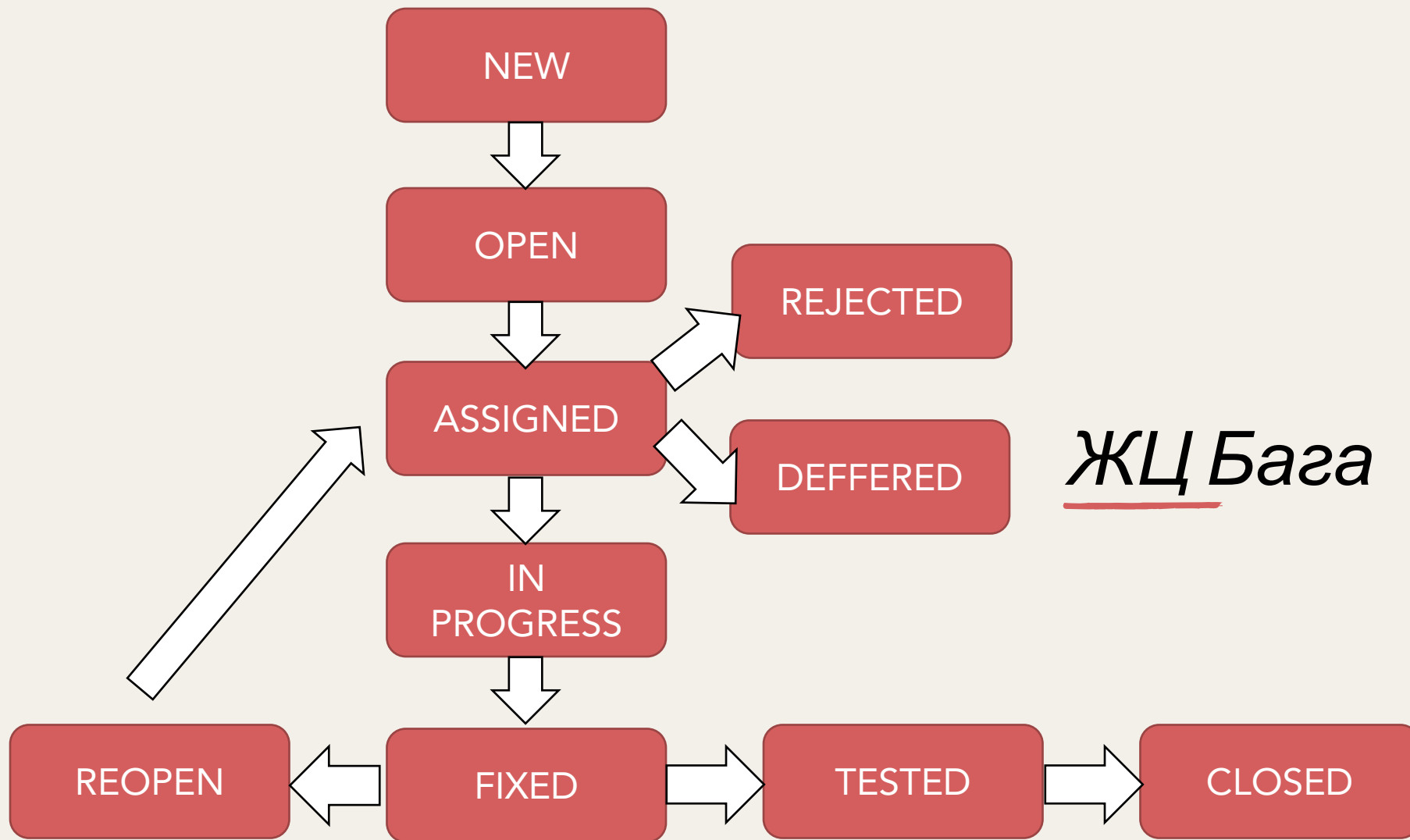
- Блокирующий (Blocker)
- Критический (Critical)
- Значительный (Major)
- Незначительный (Minor)
- Тривиальный (Trivial)

# JIRA

---

Регистрируемся и трогаем руками)

<http://bugred.ru/>



# Домашка:

---

## Задание 1 (normal)

Зарегистрируйтесь в системе тест-менеджмента Testrail, создайте новый проект.

Напишите тест-кейс для регистрации нового пользователя с валидными данными (такими, которые не повлекут за собой вероятные ошибки) на сайте (<http://prestashop.gatestlab.com.ua/en/>)

Ответ приложите в виде скриншотов

## Задание 2 (normal)

Напишите несколько тест-кейсов на работу "Корзины": отображение/удаление товаров в ней, расчет цены, скидок, изменение количества товаров и так далее.

Затем создайте тест-ран по этим кейсам и выполните его. Отмечайте, проходят тесты успешно или были ошибки в процессе выполнения. Если были - заведите багрепорты в Jira.

Ответ приложите в виде скриншотов

# Домашка:

---

## Задание 3 (hard)

Найдите в интернете определения следующих понятий: "негативный тест-кейс" и "постусловие тест-кейса".

Напишите несколько негативных тест-кейсов для тестирования функциональности поиска на <http://prestashop.qatestlab.com.ua/en/>, добавьте к ним постусловия. Пройдите по созданным тест-кейсам. Если в процессе прохождения возникают ошибки - заводите багрепорты в Jira.

Также у вас в Jira вероятнее всего не будет возможности выставить своим багрепортам приоритеты и серьезность. Сделайте это отдельным абзацем в своих багрепортах с обоснованием того, почему вы выбрали именно такие приоритет и серьезность.

## Дополнительное задание (just for fun)

Исследуйте "страницу товара". Заведите все баги, которые вы там только заметите. Ожидаемый результат пишите исходя из жизненного опыта, взаимодействия с другими интернет-магазинами.