**ДЗ №1**

1. ОПИШИТЕ ПЯТЬ ВИДОВ ТЕСТИРОВАНИЯ НА ПРИМЕРЕ ИНТЕРНЕТ-МАГАЗИНА

<https://spreecommerce-demo.herokuapp.com/>

1) Функциональное тестирование. Функциональное тестирование - проверка на соответствие реальных функций продукта заявленным функциональным требованиям. В документации должны быть детализировано описаны функциональные требования работы сайта интернет-магазина. Функции интернет-магазина:

* обзор каталога товаров с описанием и указанием стоимости стоимости
* Поиск товара по наименованию
* Добавление товара/-ов в корзину
* Удаление из корзины
* Регистрация пользователей
* Оформление заказа, выбор способа доставки
* Функция онлайн-оплаты
* и т.д.

2) Нефункциональное тестирование. Юзабилити тестирование. Невозможно формализовать проверку, где пользователям интернет-магазина будет “удобно” и “красиво”, но в общем должно быть максимально доступно, просто/легко в исполнении и интуитивно понятно для широкого круга пользователей, чтобы возникло желание вернуться на этот сайт и продолжить покупки.

3) Нефункциональное тестирование. Безопасность. Защита от злоумышленников на предмет взлома системы, защита персональных данных, вводимых пользователями (пара логин-пароль, поля с платёжными данными и т.д.)

4) Тестирование производительности. Нагрузочное тестирование. Проверка скорости загрузки страницы при постепенном увеличении одновременного посещения пользователей: 10 пользователей - 100 пользователей - 1000 пользователей… и т.д.

5) Тестирование производительности. Стресс-тестирование. Проверка поведения сайта на резкое увеличение нагрузки. Например, в день грандиозных скидок одновременно на сайт вошли десятки тысяч пользователей.

6) По разработке тестовых сценариев. Позитивный сценарий - регистрация пользователя с корректными данными и повторный беспроблемный вход в личный кабинет сайта интернет-магазина с паролем, полученным от системы на email.

7) По разработке тестовых сценариев. Негативный сценарий - регистрация пользователя с данными, по которым ранее была зафиксирована регистрация (логин, email). Система должна отвергнуть такую регистрацию и предложить пользователю либо восстановить пароль, либо зарегистрироваться с указанием другой эл.почты.

2. НАПИШИТЕ В СВОБОДНОЙ ФОРМЕ ОТВЕТ НА УКАЗАННЫЕ ВОПРОСЫ. ЗА КАЖДЫЙ ОТВЕТ МОЖНО ПОЛУЧИТЬ 4 БАЛЛА.

ПРОНУМЕРОВАННЫЕ ОТВЕТЫ ОТПРАВЬТЕ В ВАШ РЕПОЗИТОРИЙ НА GITHUB И СООБЩИТЕ МНЕ ОБ ЭТОМ НА ПОЧТУ ([DRAFTPODIUM@GMAIL.COM](mailto:draftpodium@gmail.com)), ЛИБО В СКАЙП.

1. Опишите своими словами, как вы понимаете уровни тестирования. Что необходимо указать в ответе - какие бывают виды, кто чаще всего выполняет тесты на конкретном уровне, приведите пример одного вида тестирования, которое можно выполнять на каждом из уровней.

Уровни тестирования - это деление всего цикла разработки ПО на последовательные блоки, связанных друг с другом.

В основном количество этих уровней четыре, но м.б. и три и пять.

1. Модульное тестирование (Unit testing). Это тестирование не всего ПО, а некоторых отдельных частей кода. Например, в каком-то ПО требуется регистрация пользователя с указанием города, т.е. в требуемое поле можно ввести только буквы. Небольшая часть программного кода проверяет, какие символы вводит пользователь. Если вводятся буквы, то это верные данные (валидные) и программный код их принимает. Если вводятся цифры, то программный код их не принимает, это ошибка. Таким образом тестируется только эта маленькая часть кода.  
   Unit testing чаще всего пишут разработчики, они не выполняются вручную,Unit тесты - это автотесты.
2. Интеграционное тестирование (Integration testing). Проверка на взаимодействие отдельных Unit-тестов между собой. Например, проверка валидности названия города + проверка валидности эл.почты + проверка валидности номера телефона. Т.е. проверка ранее написанных отдельных частей кода уже в общей системе.

Выполняют как разработчики, так и тестировщики. Тоже часто автотесты.

1. Системное тестирование (System nesting). Тестируется вся разработка целиком, когда продукт готов, проведены модульное и интеграционное тестирование. Проверка функциональных и нефункциональных требований продукта.  
   Выполняет тестировщик - основная активность тестировщика.
2. Приемочное тестирование (Acceptance testing). Проводится по завершении трех предыдущих блоков, когда вся система отлажена. Проверка на соответствие заявленных требований. Прием разработки заказчиком (это не значит, что приемочное тестирование выполняет заказчик, чаще всего, в основном приемочное тестирование выполняется разработчиком).

Модульное тестирование (Unit testing) можно выполнять на каждом из уровней

2. Что такое регрессионное тестирование? В каких случаях его необходимо выполнять? Какие особенности есть у этого вида тестирования?

Регрессионное тестирование - это тестирование новой версии продукта. Проводится, когда была переписана часть кода, написана новая часть кода, добавлен новый функционал в продукт, т.е. проверка того, что новое не сломало старое. Занимает много времени и часто назначают новичкам, начинающим тестировщикам. Чаще всего для регрессии есть готовые, расписанные шаги.

Иногда невозможно перетестировать всё, это может занять очень много времени. Тогда:

- тестируется то, что появилось новое + те части старого кода, которые были задействованы в новой версии (импакт анализ)

- описываем причину невозможности всё перетестировать и обращаемся к заказчику/бизнесу для определения приоритетов, и тестируем обозначенные места (анализ рисков).

Регрессионное тестирование может быть автоматизировано.

3. Что такое юнит-тесты? В чем их преимущества? Почему в большинстве случаев нельзя ограничиться только юнит-тестами?

Unit-тесты - это тестирование отдельных частей кода. Например, в каком-то ПО требуется регистрация пользователя с указанием города, т.е. в требуемое поле можно ввести только буквы. Небольшая часть программного кода проверяет, какие символы вводит пользователь. Если вводятся буквы, то это верные данные (валидные) и программный код их принимает. Если вводятся цифры, то программный код их не принимает, это ошибка. Таким образом тестируется только эта маленькая часть кода.  
Unit testing чаще всего пишут разработчики, они не выполняются вручную,Unit тесты - это автотесты.

Unit-тесты и пишутся и выполняются очень быстро. Их любят в Agile, т.к. они не занимают много времени.

Ограничиться только одними Unit-тестами недостаточно, т.к. необходима проверка на взаимодействие отдельных частей кода в целом продукте (интеграционное тестирование).

4. Предположим, вам предлагают протестировать какой-то проект (программу, сайт). Чем вы будете руководствоваться, выбирая конкретные виды тестирования для этого проекта?

Сначала необходимо с помощью вопросов определить цели и приоритеты заказчика. Какие требования к продукту? Какой должен быть ожидаемый результат? В техническом задании должны быть указаны функциональные требования. На основании полученных данных можно выбрать конкретные виды тестирования для проекта.

5. Что такое тестирование черного ящика (Black Box Testing) и чем оно отличается от тестирования белого ящика (White box testing)? Приведите пример вида тестирования, которое невозможно при тестировании черного ящика.

Черный ящик - неизвестно содержимое. Тестирование черного ящика (Black Box Testing) - мы не знаем устройство системы, в общей картине видим всё, как и другие пользователи. Тестирование путем кликанья мышью, вводом данных с клавиатуры, переходом по ссылкам, нажатием на иконки и т.д. и наблюдением за результатом.

Тестирования белого ящика (White box testing) - мы находимся внутри компании-разработчика, мы видим/знаем, как работает система, механизм разработки прозрачен. На основании знаний системы можно проводить какие-либо виды тестирования.