1. **Что такое тестирование**

**Тестирование программного обеспечения** — это процесс, который позволяет узнать, делает ли программа то, для чего она была создана, и не делает ли она ничего лишнего

**Тестирование** — это процесс проверки программного обеспечения, системы или приложения на соответствие определенным требованиям и оценки их качества.

**Тестирование программного обеспечения (Software Testing)** — проверка соответствия реальных и ожидаемых результатов поведения программы, проводимая на конечном наборе тестов, выбранном определённым образом.

Ожидаемый результат написан в **Требованиях** — это спецификация (описание) того, что должно быть реализовано.

Её берем у **Бизнес аналитик** или **Прожект менеджер**

**Если ее нет** то составляем ее сами (исходя из бюджета, требований пользователей, здравого смысла и др.) или берем что-то похожее у конкурентов

1. **Кто такой Тестировщик ПО**

**Тестировщик ПО** — специалист, который занимается тестированием программного обеспечения

Тестировщик ищет дефекты, оформляет их и передает разработчику для исправления.

Контролирует корректность исправления дефекта

1. **Разница между QC (Quality Control) и QA (Quality Assurance)**

**QC (Quality Control) — Контроль качества продукта**

Анализ результатов тестирования и качества новых версий выпускаемого продукта.

К задачам контроля качества относятся:

* проверка готовности ПО к релизу;
* проверка соответствия требований и качества данного проекта.

Обеспечение качества (Quality Assurance - QA) - это совокупность мероприятий, охватывающих все технологические этапы разработки, выпуска и эксплуатации программного обеспечения (ПО) информационных систем, предпринимаемых на разных стадиях жизненного цикла ПО для обеспечения требуемого уровня качества выпускаемого продукта.

**QA (Quality Assurance) — Обеспечение качества продукта**

Изучение возможностей по изменению и улучшению процесса разработки, улучшению коммуникаций в команде, где тестирование является только одним из аспектов обеспечения качества.

К задачам обеспечения качества относятся:

* проверка технических характеристик и требований к ПО;
* оценка рисков;
* планирование задач для улучшения качества продукции;
* подготовка документации, тестового окружения и данных;
* тестирование;
* анализ результатов тестирования, а также составление отчетов и других документов.

Контроль качества (Quality Control - QC) - это совокупность действий, проводимых над продуктом в процессе разработки для получения информации о его актуальном состоянии в разрезах: "готовность продукта к выпуску", "соответствие зафиксированным требованиям", "соответствие заявленному уровню качества продукта".

1. **Цель тестирования**

**Цель тестирования** — проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении, которые должны быть выявлены до того, как их обнаружат пользователи программы.

* Поиск ошибок
* Уменьшение рисков и затрат
* Удовлетворение потребностей пользователей
* Сохранить имидж компании

1. **Жизненный цикл тестирования**

Описание пути развития По от идеи до окончания ее поддержки

* Общее планирование и анализ требований
* Уточнение критериев приемки
* Уточнение стратегии тестирования
* Разработка тест кейсов
* Выполнение тест кейсов и фиксация найденных дефектов
* Анализ результатов тестирования и отчетность

1. **Жизненный цикл ПО**

* **анализ (сбор требований)**

сбор требований от заказчика, как должен выглядеть и работать конечный продукт

* **проектирование**

использование требований, дизайн, архитектуру

* **реализация (разработка, кодирование)**
* **тестирование продукта**
* **внедрение**

запуск проекта, передача его заказчику

* **Эксплуатация**

**(обслуживание и поддержка)**

**Верификация и валидация — два понятия тесно связаны с процессами тестирования и обеспечения качества. К сожалению, их часто путают, хотя отличия между ними достаточно существенны.**

**Верификация (verification) — это процесс оценки системы, чтобы понять, удовлетворяют ли результаты текущего этапа разработки условиям, которые были сформулированы в его начале.**

**Валидация (validation) — это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, его требованиям к системе.**

1. **Виды тестирования**

**Смок (основные бизнес функции (авторизация, работа с картами, оплата, работа корзины, каталог товаров и информация о них и тд))**

**Регресс проверка тест кейсов написаных в предыдущтк итерациях проекта (билда)**

**(полное тестирование ) проверка старой функциональности после выхода нового билда (чтобы проверить что после ввода нового функционала старый не был затронут)**

**Функциональное**

**Нефункциональное**

* **Ux/ui**
* **Нагрузочное (сколько предполагается системой )**
* **Стресс (больше чем нужно) постепенное увеличение и доходим до предела**
* **Локализация (формат даты,язык,валюта и др)**
* **Тестирование документации и требований к ней**
* **Кроссбраузерное**
* **Кроссплатформенное**
* **Unit (обычно тестирует сам программист функцию)**

1. **Негативное (ввод неверных данных с целью вывода сообщения об ошибке)**
2. **Позитивное**
3. **Ручное**
4. **Авто**

* **Белый ящик (доступ к коду) тестирую код**
* **Черный ящик (без доступа к коду) тестирую бизнес требования**

**Проверка соответствие требованиям**

* **Серый ящик (Тестирую через api )**
* **Инсталяционнре (обновление ,удаление приложения)**

**Альфа(сотрудники компании), бета (тестируют пользователи и мы),**

1. **Уровни тестирвания**

* **Компонентное (Unit)**

Проверяет отдельные модули или компонентное наполнение ПО на корректность их функционирования. Тесты проводятся над отдельными функциями, процедурами или классами.

* **Интеграционное**

Проверяется взаимодействие и взаимосвязь между различными модулями или компонентами программного обеспечения. Цель — убедиться, что они работают вместе и взаимодействуют без ошибок.

* **Системное**

Проводится на уже интегрированной системе в целом. Здесь проверяется функциональность, производительность и поведение системы в соответствии с требованиями и ожиданиями.

* **Приемочное**

Выполняется для проверки соответствия программного обеспечения конечным требованиям заказчика или пользователя. Оно проводится с целью получить подтверждение от заказчика о готовности продукта к использованию.

1. **Тестовая документация**

**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

**Тест план должен отвечать на следующие вопросы:**

* Что необходимо протестировать?
* Как будет проводиться тестирование?
* Когда будет проводиться тестирование?
* Критерии начала тестирования.
* Критерии окончания тестирования.

**Основные пункты тест плана:**

* Идентификатор тест плана (Test plan identifier);
* Введение (Introduction);
* Объект тестирования (Test items);
* Функции, которые будут протестированы (Features to be tested;)
* Функции, которые не будут протестированы (Features not to be tested);
* Тестовые подходы (Approach);
* Критерии прохождения тестирования (Item pass/fail criteria);
* Критерии приостановления и возобновления тестирования (Suspension criteria and resumption requirements);
* Результаты тестирования (Test deliverables);
* Задачи тестирования (Testing tasks);
* Ресурсы системы (Environmental needs);
* Обязанности (Responsibilities);
* Роли и ответственность (Staffing and training needs);
* Расписание (Schedule);
* Оценка рисков (Risks and contingencies);
* Согласования (Approvals).

**Чек-лист (check list) —** это документ, который описывает что должно быть протестировано. Чек-лист может быть абсолютно разного уровня детализации.

**Тестовый сценарий (test case)** — это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

**Атрибуты тест кейса:**

● Идентификатор набора тестов (Test Suite ID): Идентификатор набора тестов, в которых входит этот кейс;

● Идентификатор тестового кейса (Test Case ID): Идентификатор самого кейса;

● Заголовок кейса (Test Case Summary): Краткое и емкое название проводимой проверки;

● Связанное требование (Related Requirement): Идентификатор требования, к которому относится / отслеживается данный тестовый пример;

● Предварительные условия (Prerequisites): Любые предпосылки или предварительные условия, которые должны быть выполнены перед выполнением теста;

● Шаги выполнения (Test Script / Procedure): Шаги выполнения теста;

● Тестовые данные (Test Data): Тестовые данные или ссылки на тестовые данные, которые должны использоваться при проведении теста;

● Ожидаемый результат (Expected Result): результат, который мы ожидаем получить после выполнения шагов теста;

● Статус пройден или не пройден (Status): Другие статусы могут быть «не выполнено», если тестирование не проводится, и «Заблокировано», если тестирование заблокировано;

● Заметки (Remarks): Любые комментарии к тесту или выполнению теста;

● Создано (Created By): Имя автора тестового примера;

● Дата создания (Date of Creation): Дата создания тестового примера (опционально модификации);

● Выполнено (Executed By): Имя человека, выполнившего тест;

● Дата выполнения (Date of Execution): Дата выполнения теста;

● Тестовое окружение (Test Environment): оборудование / программное обеспечение / сеть, в которых выполнялся тест, т.е. все необходимые сведения об окружении, чтобы можно было воспроизвести полученный результат.

**Баг репорт**

Отчет о дефекте (bug report) — это документ, описывающий ситуацию, которая привела к обнаружению дефекта, с указанием причин и ожидаемого результата.

Атрибуты:

* Уникальный идентификатор (ID)
* Тема (краткое описание, Summary)
* Подробное описание (Description)
* Шаги для воспроизведения (Steps To Reproduce)
* Ожидаемый результат (Expected result)
* Фактический результат (Actual result)
* Вложения (Attachments)
* Серьезность (важность, Severity)

Блокирующий (S1 – Blocker) - достиг граничных значений

Критический (S2 – Critical)

Значительный (S3 – Major)

Незначительный (S4 – Minor)

Тривиальный (S5 – Trivial)

* Приоритет (срочность, Priority)

**Высокий (High, P1).** Критически важный для проекта дефект, который требует немедленного исправления.

**Средний (Medium, P2).** Не критичен, но требует обязательного решения в ближайшее время.

**Низкий (Low, P3).** Не является критическим и не требует срочного решения. Может быть исправлен, когда появится возможность.)

* Статус (Status)
* Окружение (Environment)

Класс эквивалентности (спец символы, цифры)

Класс гранич значений