1. **Что такое тестирование.**

Тестирование - процесс, который позволяет специалисту с помощью различных способов, техник, методов, видов проверок находить и фиксировать несоответствия фактического и ожидаемого результата продукта в виде дефектов, и, таким образом, оптимизировать время и ресурсы заказчика, вложенные в достижение ожидаемого результата, для создания конечного программного продукта, который соответствует всем требованиям заказчика и является удобным для конечного пользователя, что позволяет реализовывать бизнес-идею на финансовом уровне.

1. **Что такое QA, QC, Testing.**

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.

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.

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

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.

**3. Качество.**

Качество - совокупность минимально допустимых требований к продукции, обусловливающих её пригодность удовлетворять определённые потребности в соответствии с её назначением.

Иными словами показателем качества является удовлетворенность клиента продуктом, которым он пользуется.

**4. Качество ПО.**

Качество программного обеспечения - набор необходимых характеристик, позволяющих использовать его в целях, для которых он создавался, наиболее эффективно.

**5. Когда нужно начинать тестирование**.

Тестирование начинается с этапа сбора и разработки требований. По сути это вторая стадия в жизненном цикле ПО после этапа возникновения идеи необходимости создания самого продукта. Уже на этапе сбора требований можно начинать их тестировать.

**6. Когда нужно заканчивать тестирование. Эвристика.**

* Пройдены все запланированные тесты, продукт можно выпускать в релиз.
* Нет критических ошибок.
* Когда заказчик отказался от продолжения работы.
* Когда не пройден smoke тест.

Процесс тестирования на основе эвристик – это такая технология тестирования алгоритмов, приложений и программ, при использовании которой стратегия тестирования основывается на предыдущем опыте и данных о вероятности наступления различных событий.

Мнемоника – это набор правил и приёмов, которые помогают эффективно запоминать необходимые сведения (информацию).

Ключевые преимущества тестовых эвристик:

* Они позволяют не забывать контекст, в котором непосредственно и используется тестируемое приложение или продукт;
* Краткость. Все эвристики удобно помещаются в небольшие диаграммы связей, листок бумаги или же в текстовый файл;
* С помощью эвристик можно проводить качественное исследовательское тестирование утилит в более детализированном формате;
* Они помогают не совершать ошибки, что могут быть допущены при аналогичных обстоятельствах во время тестирования похожего программного обеспечения. Философия использования эвристик позволяет разрабатывать специальные «напоминания» на базе предыдущего опыта – личного или наработок других QA специалистов.

Но при всем этом, эвристика – метод крайне несовершенный. Ее базовый недостаток в том, что эвристическая оценка не может предполагать применение пользовательского тестирования и анализа выполнения настоящих сценариев, ее итоги могут быть чрезвычайно субъективными и необоснованными с практической стороны.

**7. За что тестировщику платят деньги?**

Суть работы тестировщика - сделать продукт максимально удобным для клиента, в соответствии с запрашиваемыми требованиями заказчика. Продукт должен соответствовать ожидаемым клиентским либо пользовательским требованиям.

Задача тестировщика заключается в решении проблемы компании. Чем раньше будут найдены ошибки, тем меньше ресурсов будет потрачено на их исправление, тем быстрее будет выпущен продукт, следовательно, компания начнет зарабатывать деньги. В результате, благодаря работе тестировщика, компания экономит деньги, время компании, а также сохраняет репутацию поставщика качественных услуг.

**8. Что такое дефект, баг, ошибка.**

Дефект, баг, ошибка - все эти термины обозначают наличие найденных несоответствий фактического результата выполнения программы ожидаемому результату.

**9. Bug report (все пункты и основная структура).**

Баг-репорт — это документ, содержащий отчёт о любом недостатке в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию.

*Основная структура bug report.*

-title/summary - пишутся по формуле ЧТО, ГДЕ, КОГДА (при каких условиях),

-steps,

-актуальный результат,

-ожидаемый результат.

*Полная структура bug report.*

*-Описание* нашей проблемы. Например, при работе в jira обычно в описании также прописывают шаги воспроизведения данного бага, которые помогут повторить путь и прийти к фактическому результату.

*-Проект* - название проекта, к примеру, интернет-магазин “Калоша”. В этом поле, указывается название проекта “Калоша”. Для того, чтобы в дальнейшем мы могли спокойно проследить - к какому проекту относится баг-репорт.

*-Компонент* - элемент, в котором непосредственно был обнаружен данный дефект.

*-Версия билда -* версия сборки, на которой воспроизводится обнаруженный дефект.

*-Серьезность (severity)* - это критичность бага, степень его влияния на приложение. В баг-трекинговой системе severity обычно обозначает тестировщик.

Виды ***severity***:

*-blocker\*,*

*-critical,*

*-major,*

*-minor,*

*-trivial.*

\*Blocker становится critical, когда есть workaround, обходной путь решения проблемы.

-*Приоритет (priority)* обозначает очередь исправления данного бага. Обычно его выставляет РМ (Product Manager).

*-Автор*, сотрудник, который нашел баг. Это может быть как тестировщик, так и другой участник проекта, либо команды.

-*Назначение*.

Основное отличие тест-кейса от баг репорта - наличие фактического результата в баг репорте.

**10. Классификация по уровням тестирования.**

* «По уровню детализации приложения» = «по уровню тестирования».
* «По (убыванию) степени важности тестируемых функций» = «по уровню функционального тестирования». (*Куликов*)

1. **Unit тестирование (компонентное, модульное)**. Им занимается разработчик программного кода, так как именно Unit testing позволяет протестировать отдельные компоненты исходного кода программы. На примере интернет-магазина к таким модулям можно отнести страницу авторизации, поиск товара, перемещение его в корзину, оплата заказа
2. **Интеграционное** тестирование. Под ним понимают тестинг части системы, состоящей из двух и более модулей, например, как мы можем со страницы корзины произвести оплату посредством платежной системы.

Существует *две разновидности интеграционного тестирования*. Давайте поговорим о каждой из них:

-Тестирование **интеграции компонентов**, т.е. как *отдельные* модули одного приложения взаимодействуют между собой.

**-Системное интеграционное тестирование**, под которым понимают тестирование *взаимодействия* между всеми компонентами системы, либо взаимодействия разных систем между собой, либо тестирование интерфейсов с помощью которых взаимодействуют системы.

1. **Системное тестирование**. Оно выполняется на полной интегрированной системе с целью проверки соответствия системы исходным требованиям. Например, если мы говорим об интернет-магазине, то здесь мы будем смотреть как вообще все модули, которые были разработаны для этого магазина, работают между собой. Можно ли пройтись по всему Business Flow данного приложения, т.е. можем ли мы зарегистрироваться, найти товар, положить его в корзину, оплатить наш заказ, отслеживать этот товар в пути. Т.е. мы тестируем всё приложение целиком.
2. **Приёмочное тестирование**. Это вид тестирования, проводимый на этапе сдачи готового продукта, либо же какой-то его готовой части заказчику. Целью приемочного тестирования является определение готовности продукта и достигается это путем прохода тестовых сценариев, случаев, которые построены на основе спека и требований к продукту. По сути это финальный этап тестирования продукта перед его релизом. Обычно тестируется только основной функционал, который не требует тщательных проверок.

Существует несколько типов приемочного тестирования, которые тоже необходимо знать.

- *Пользовательское приемочное* тестирование (UAT, User acceptance testing). Оно проводится пользователями конечного продукта. Например, когда мы знаем, что наш продукт готов к релизу, тогда мы можем собрать некоторую группу конечных пользователей, которая его протестирует, посмотреть, действительно ли данный продукт соответствует их ожиданиям. Они дадут нам какое-то заключение, мы исправим все критичные баги и сможем спокойно выпустить продукт.

- *Эксплуатационное* тестирование. Обычно данное тестирование выполняется либо пользователями, либо же администратором в среде, которая имитирует реальные условия рабочего окружения. На этом этапе проводится проверка тестирования резервного копирования или, позже, аварийного восстановления системы, также его безопасность.

- Тестирование на *соответствие контракту*. Данное тестирование представляет собой проверку на соответствие спека какому-то другому документу, нормативному акту. Например, в стране существует ГОСТ, который определяет, как наш продукт должен соответствовать данному нормативному правовому акту, т.е. определяет его основные ключевые особенности. Мы должны обязательно разрабатывать продукт в полном соответствии к этому нормативному акту. Либо же один из самых известных документов - определяет требования к защите пользовательских данных. Если данные требования не учитываются при разработке продукта, то компания может получить штрафы в дальнейшем.

- *Альфа-тестирование* - эксплуатационное тестирование, проводится на стороне разработчика, но не в рамках нашей компании.

- *Бета-тестирование* проводится на внешней стороне и без участия разработчиков. Обычно бета-тестирование проводится небольшой группой пользователей, чтобы получить их отзыв до выхода продукта на рынок, это известная практика для компьютерных игр. Т.е. заказчик выбирает некую целевую группу пользователей и выпускает для них версию продукта, на основании их фидбека будет проводиться исправление каких-то критичных багов, либо же мы будем просто понимать, что продукт соответствует их ожиданиям и его можно уже выпускать.