Принципы тестирования:

* Принцип присутствия дефекта, если вы протестировали функционал и не нашли в нем багов, это не значит, что в нем его нет
* Принцип пестицида, способность багов адаптироваться к тестам, это значит, что необходимо модифицировать тесты, по мере развития кода.
* Принцип раннего тестирования, стоимость исправления бага экспоненциально пропорциональна времени его обнаружении, то есть чем раньше и быстрее мы найдем баг и исправим, тем дешевле он обойдется.
* Принцип концентрации багов: 80% багов сконцентрировано в 20% приложения. ( основные главные места концентрация – это платформа apple, платежные системы и интеграция с любимы сторонними системами)

Модели разработки:

* Скрам (scrum) или итерационная модель разработки. Её суть в том, что весь процесс разработки делится на этапы, обычно одинаковой продолжительности (от 2 до 4 недель), которые называются спринтами. В рамках спринта реализуются задачи берущиеся из бэклога относительно их приоритета. За весь спринт происходит полный цикл разработки. Каждый спринт – это маленькая завершенная разработка. После релиза спринта определяется список задач, которые можно либо необходимо реализовать. Этот список задач кладётся в бэклог и на следующий спринт берутся задачи из прошлого по их приоритету. Скрам – это самый распространённая модель при разработке, так как это самый оптимальный способ по времени-результату. «Наилучшая из худших»
* Водопадная модель (waterfall) разработки. Она заключается в том, что весь процесс разработки делится на этапы, и к следующему этапу можно только если прошлый этап был полностью завершён. Если в процессе реализации этапа возникла необходимость изменений на предыдущем этапе, то в этом случае происходит откат до по предыдущего и внесение изменений до полного его завершения. Её преимущество в предсказуемости результата. Минус в увеличенном времени на разработку.
* Гибкая модель (agile). В рамках разработки проекта все этапы выполняются одновременно. При необходимости внесения изменений, они происходят сразу на этом же этапе. Скорость выполнения проекта минимально, но результат при этом непредсказуем. Её используют при условиях очень ограниченного времени.

Жизненный цикл тестирования:

1. Общее планирование и анализ требований
2. Уточнение критериев приемки. Этап на котором мы выясняем, что на самом деле хочет заказчик
3. Уточнение стратегии тестирования
4. Разработка тест-кейсов
5. Выполнение тест-кейсов
6. Фиксация найденных дефектов.
7. Анализ результатов тестирования
8. Отчетность
9. Повтор цикла

В требовании необходимо описывать то, что имеет значение для приложения или проекта.

Источники и пути выявления требований:

* Интервью. Интервью проводится с диктофоном, и все выписывается на документ.
* Работа с фокусными группами, которые описываются в концепции.
* Анкетирование. Необходимо учитывать, что какой-то процент анкетируемых проставит случайные ответы.
* Семинары и мозговой штурм.
* Наблюдение. Очень важно фиксировать результаты наблюдений.
* Прототипирование. Создание быстрого сырого приложения, которое имеет только необходимый функционал.
* Анализ документов.
* Моделирование процессов и их взаимодействие.
* Самостоятельное описание.

Прототипирование: оно проводится для проверки сомнительных частей программы, для которых есть некоторые неопределенности.