**Цели тестирования:**

1) Проверить продукт на соответствие ожиданиям пользователя;

2) Показать разработчику и клиенту, что ПО отвечает заявленным требованиям. Соответствующий данной цели тест называют валидацией (проверкой достоверности). Успешное валидация указывает, что система работает как надо;

3) Поиск ситуаций, когда ПО ведёт себя в несоответствии заданным требованиям описанным в спецификации.

**Тестовые мероприятия (зачем они, время проведения):**

1) Дымовое тестирование (Smoke test) - Проверка самой важной функциональности продукта. Позволяет на начальном этапе выявить основные критические дефекты. Программа не проходящая “дымовой тест” , не имеет смысла в глубоком тестировании. Из-за того, что проверки в смоке практически всегда одинаковы, целесообразно их автоматизировать.

2) Тестирование критического пути (Critical path test) - основной тип тестирования, во время которого проверяются основная масса функций и элементов на предмет правильности работы при стандартном их использовании.

* Позитивный тест критического пути - это проверка работоспособности функций ПО, с которыми пользователь сталкивается ежедневно. (Ввод валидных данных)
* Негативный тест критического пути - это проверка всевозможных вариантов нестандартного использования функциональности.

3) Расширенное (Extended test) - Проверка всей заявленной функциональности. С проверкой нестандартного использования ПО (ввод специальных символов, нелогичное кликанье по кнопкам…)

**Тестирование и отладка:**

1) Тестирование - процесс исследования ПО с целью выявления ошибок и определения соответствия между ожидаемым и фактическим поведением программы.

2) Отладка - это процесс исправления конкретной ошибки из программы, которая была найдена в процессе тестирования.

**Принципы тестирования:**

1) Тестирование показывает наличие дефектов - тестирование снижает вероятность наличия дефектов, находящихся в ПО, но не доказывает, что их нет.

2) Исчерпывающее тестирование недостижимо - полное тестирование с использованием всех комбинаций вводов віполнить невозможно. Нужно использовать техники тест дизайна, анализ рисков, расстановку приоритетов, чтобі более грамотно распределить усилия на тестирование.

3) Раннее тестирование - тестирование необходимо начинать как можно раньше в жизненном цикле разработки ПО, чтобы найти дефекты как можно раньше.

4) Скопление дефектов - усилие затраченное на тестирование должно быть пропорционально плотности дефектов. Большая часть дефектов обнаруживается контретных функциональностях (Принцип Парето: 80% дефектов находится в 20% функциональности). Их и надо тестировать тщательней.

5) Парадокс пестицида - если одни и те же тесты будут выполняться много раз, то этот набор тестовых сценариев больше не будет находить новых дефектов. Нужно периодически модифицировать/дополнять тест кейсы.

6) Тестирование зависит от контекста - выбор методологии, техники и типа тестирования будет напрямую зависеть от природы самой программы.

7) Заблуждение об отсутствии ошибок - процесс тестирования при котором не было обнаружено дефектов, не доказывает, что в ПО отсутствуют дефекты и программа готова к релизу.

**Типы и виды тестирования:**