

Виды тестирования

По доступу к коду

- Метод "белого" ящика - тестер компетентен, ему предоставляется доступ к коду
- Метод "черного" ящика - без доступа к коду, тестируется функционал
- Метод "серого" ящика - смешанный (белый + серый)

По запуску кода на исполнение

- Статическое (по требованиям заказчика)
- Динамическое (на "рабочей" системе)

По уровню детализации приложения

- Модульное тестирование (по методу белого ящика, Unit-тесты, нужен доступ к коду)
- Интеграционное тестирование (проверка взаимодействия нескольких модулей)
- Системное тестирование (как приложение работает в комплексе с ОС - устойчивость, надежность, безопасность)
- Приемочное тестирование (соответствие бизнес-процессам и требованиям пользователя) - может проводить тестировщик и пользователь

По степени автоматизации

- Ручное (мануальное)
- Автоматизированное (Программируются автотесты)

По принципам работы с приложением

- Позитивное тестирование (все действия выполняются в соответствии с инструкцией пользователя без отклонений)
- Негативное тестирование (используются данные для тестов, которые приводят к ошибкам) деление на 0, например

По уровню функционального тестирования

- Дымовое (запуск приложения) + санитарное (основные бизнес функции)
- Тестирование критического пути (Проверка типичных пользовательских функций, обычно применяемых при использовании приложения)
- Расширенное тестирование - как можно более полное тестирование функционала

В зависимости от исполнителей

- Альфа-тесты (закрытое тестирование внутри команды)
- Бета-тестировние (+ реальные пользователи)

В зависимости от целей тестирования

Функциональное

- Модульное
- Дымовое+санитарное
- Регрессионное (новое исправление не нарушило старый функционал)
- Альфа/бета -тестирование
- Системное тестирование
- Интеграционное тестирование
- UAT (Приемочное тестирование) критический путь.
- End-to-end тесты (сквозное тестирование от начала процесса до его завершения)

Нефункциональное

- Производительность (стабильность на ограниченных ресурсах)
- Нагрузочное тестирование (когда система дает сбой - свыше лимитированной нагрузки)
- Стресс-тест (нагрузка превышает расчетный уровень, либо недостаточно ресурсов для устойчивой работы приложения)
- Инсталляционное тестирование (установка, удаление и настройка приложения)
- Тестирование интерфейсов (UI) шрифты, размеры элементов, дизайн, цвета
- Удобство использования (UX) - юзабилити
- Локализация (даты, язык, особенности цветовой гаммы)
- Надежность (стабильность при длительном функционировании при стресс-нагрузке)
- Тестирование безопасности (несанкционированный доступ, защита от вирусов, от атак хакеров)
- Восстановление (возможность восстановления после "падения")
- Кросс-платформенное (на разных браузерах)