

Виды тестирования

По доступу кода

- Метод серого ящика
 - Тестирование, являющееся комбинацией White Box и Black Box подходов, либо как дополненный чёрный ящик. Имея доступ к внутреннему устройству ПО, можно написать более эффективные тест-кейсы, но само тестирование провести с позиции пользователя
- Метод чёрного ящика
 - Тестирование, основанное на требованиях и спецификациях. Внутренняя система ПО неизвестна, тестирование проводится только со внешними интерфейсами. Может быть применено на всех уровнях (модульном, интеграционном, системном, приемочном)
- Метод белого ящика
 - Тестирование, при котором внутренняя структура, устройство, реализация системы известны тому, кто её тестирует. При выборе входных данных, мы знаем, каким должен быть результат обработки.

По запуску кода на исполнение

- Статическое
 - Существуют только требования (нет объекта тестирования)
- Динамическое
 - Существует готовый сайт (объект тестирования)

По уровню детализации приложения

- Модульное тестирование
 - Логически выделенный элемент - модуль. Проводится разработчиками; предполагает полный доступ к коду - метод чёрного ящика (UNIT-тесты)
- Интеграционное тестирование
 - Направлен на проверку коррекции взаимодействия нескольких модулей в единое целое
- Системное тестирование
 - Процесс тестирования системы, в котором проводится не только функциональное тестирование, но и оценка её характеристик. Проверка работы приложения в какой-либо системе
- Приёмочное тестирование
 - Проверяется соответствие системы потребностям, требованиям и бизнес-процессам пользователя

По степени автоматизации

- Ручное
 - Является мануальным тестированием (проводится фактически вручную, поэтому его также называют monkey-test)
- Автоматизированное
 - Автотесты - создание функции с помощью программного продукта, которая сама ищет ошибки

По принципам работы с приложением

- Позитивное
 - Штатное тестирование по нормальному ожидаемому сценарию. При введении валидных данных ожидаемо должны получить правильный результат
- Негативное
 - Отрицательное тестирование (попытка сломать систему). При введении невалидных данных, мы проверяем ожидаемый результат. В итоге получаем отличный от ожидаемого результат.

По уровню функционального тестирования

- Дымовое + санитарное тестирование
 - Проверка возможности запуска ПО и основные функции бизнес-требований
- Критического пути
 - Проверяются основные пользовательские функции, которые используются ими в повседневной жизни
- Расширенное тестирование
 - Направлено на исследование всех заявленных требований по функциональности

В зависимости от исполнителя

- Альфа-тест
 - Закрытое тестирование внутри команды (соглашение о неразглашении ограничивает круг лиц, которые могут иметь доступ к информации)
- Бета-тест
 - Привлечение к тестированию сторонних лиц совместно с командой разработчиков

В зависимости от целей тестирования

- Функциональное тестирование
 - Модульное тестирование
 - Подтема 1
 - Дымовое + санитарное тестирование
 - Подтема 1
 - Регрессионное тестирование
 - После выпуска обновлений программы, необходимо проверить не нарушена ли функциональность программы в целом
 - Альфа/Бета тестирование
 - Тестирование ПО внутри команды, а также с привлечением сторонних лиц
 - Системное тестирование
 - Процесс тестирования системы, в котором проводится не только функциональное тестирование, но и оценка её характеристик. Проверка работы приложения в какой-либо системе
 - Интеграционное тестирование
 - Направлен на проверку коррекции взаимодействия нескольких модулей в единое целое
 - Критического пути
 - Проверяются основные пользовательские функции, которые используются ими в повседневной жизни
 - E2E
 - End to end - тестирование. Тестовые площадки (stage, pre-prodaction) дают возможность провести тестирование функционала и визуала без нарушения работоспособности программы и причинения неудобств для пользователей
- Нефункциональное тестирование
 - Тестирование производительности
 - Определение стабильности и потребления ресурсов в условиях нагрузок (вход разного количества пользователей - изменение производительности ПО; мощная анимация и пр.). Выяснить причины снижения производительности ПО
 - Нагрузочное тестирование
 - Проверка работы ПО сверх лимита и на каком этапе выходит сбой
 - Стресс-тест
 - Проверка стабильности работы ПО на максимальной нагрузке (в пределах заявленных возможностей)
 - Инсталляционное
 - Направлено на проверку успешной установки и настройки обновления и удаления приложения (скачать, обновить, удалить). Подходит для тестирования мобильных приложений
 - Тестирование интерфейса UI
 - Шрифты
 - Размеры
 - Цвета
 - Удобство использования UX
 - Понятность, привлекательность ПО для пользователей
 - Локализация
 - Проверка адаптации ПО для ее аудитории в соответствии с ее культурными особенностями (интернационализация, даты, цвета, текст и др.)
 - Надёжность
 - Проверка работоспособности ПО при длительном тестировании
 - Безопасность
 - Стратегия тестирования, которая направлена на проверку систем безопасности (анализ рисков в защите целостности ПО, атаки хакеров, несанкционированного доступа к информации и пр.)
 - Восстановление
 - Наблюдение за действиями системы, направленными на ее восстановление
 - Кроссплатформенное тестирование
 - Тестирование ПО в условиях работы на различных платформах