

Виды тестирования

По коду доступа

- Метод чёрного ящика
- Метод белого ящика
- Метод серого ящика

По запуску кода на исполнение

- Статическое
- Динамическое

По уровню детализации

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование
- Приёмное тестирование

По степени автоматизации

- Ручное тестирование
- Автоматическое тестирование

По принципам работы с приложением

- Позитивное тестирование
- Негативное тестирование

По уровню функционального тестирования

- Дымовое+Санитарное
- Критического пути
- Расширенное тестирование

В зависимости от исполнителя

- Альфа тесты
- Бета тесты
- Гамма тест

В зависимости от целей тестирования

- функциональное тестирование
 - модульное тестирование
 - дымовое+ санитарное
 - критического пути
 - Альфа/Бета тестирование
 - Системное тестирование
 - Интеграционное тестирование
 - регресс
 - E 2 E
- Нефункциональное тестирование
 - Тестирование производительности
 - Нагрузочное тестирование
 - Стресс тестирование
 - Инсталляционное тестирование
 - Тестирование интерфейса UI
 - Шифр
 - Цвет
 - размер
 - Удобство пользователя
 - Тестирование локализации
 - Тестирование надёжности
 - Тестирование безопасности
 - Восстановление

Тестирование методом «черного ящика» - это стратегия, в которой тестирование основано исключительно на требованиях и спецификациях, при этом мы не знаем, как устроена внутри тестируемая система и работаем исключительно с внешними интерфейсами тестируемой системы или компонента. Тестирование черного ящика может быть применено на всех уровнях - модульном, интеграционном, системном и приемочном.

метод тестирования ПО, который предполагает, что внутренняя структура/устройство/реализация системы известны тому, кто её тестирует. Мы выбираем входные значения, основываясь на знании кода, который будет их обрабатывать. Точно так же мы знаем, каким должен быть результат этой обработки.

это метод тестирования программного продукта или приложения с частичным знанием его внутреннего устройства. Для выполнения тестирования «серого ящика» нет необходимости в доступе тестировщика к исходному коду. Тесты пишутся на основе знания алгоритма, архитектуры, внутренних состояний или других высокоуровневых описаний поведения программы.

Статическое тестирование — процесс тестирования, который проводится для верификации практически любого артефакта разработки: программного кода компонент, требований, системных спецификаций, функциональных спецификаций, документов проектирования и архитектуры программных систем и их компонентов.

Динамическое тестирование - это методика, направленная на проверку функционала программы, во время выполнения кода. То есть, данный тип тестирования подразумевает фактическую эксплуатацию программы и определение того, как работает ее функционал, в соответствии с ожиданиями или нет.

Модульное тестирование (Unit Testing)- это тип тестирования программного обеспечения, при котором тестируются отдельные модули или компоненты программного обеспечения. Его цель заключается в том, чтобы проверить, что каждая единица программного кода работает должным образом. Данный вид тестирования выполняется разработчиками на этапе кодирования приложения. Модульные тесты изолируют часть кода и проверяют его работоспособность.

Интеграционное тестирование (integration testing) — это тестирование, проверяющее работоспособность двух или более модулей системы в совокупности — то есть нескольких объектов как единого блока. В тестах взаимодействия же тестируется конкретный, определенный объект и то, как именно он взаимодействует с внешними зависимостями.

Системное тестирование — процесс тестирования системы, на котором проводится не только функциональное тестирование, но и оценка характеристик качества системы — ее устойчивости, надежности, безопасности и производительности.

Приёмное тестирование — проверяет соответствие системы потребностям, требованиям и бизнес-процессам пользователя.

Ручное тестирование — это процесс поиска ошибок в программе без использования специальных ПО, силами человека. Тестирущик имитирует реальные действия пользователя и старается охватить максимум функций продукта и найти ошибки (на языке QA — «баги»).

Автоматизированное тестирование - это метод тестирования программного обеспечения, который выполняется с использованием специальных программных средств, которые, в свою очередь необходимы для выполнения набора тестовых примеров.

«Позитивное» тестирование — это тестирование на данных или сценариях, которые соответствуют нормальному (штатному, ожидаемому) поведению тестируемой системы. Основной целью «позитивного» тестирования является проверка того, что при помощи системы можно делать то, для чего она создавалась.

Негативное или отрицательное тестирование — это тип тестирования программного обеспечения (ПО), направленный на проверку того, что система или приложение ведут себя должным образом в негативных ситуациях, то есть, когда они получают недопустимые или неожиданные входные данные. Суть в том, что нужно попробовать сломать приложение, новый функционал, раздел, кнопку и так далее.

Дымовое тестирование — это тип тестирования программного обеспечения, которое обычно выполняется в начале цикла разработки, чтобы убедиться, что наиболее важные функции приложения работают правильно.Санитарное относится к виду тестирования которое используется с целью доказательства работоспособности конкретной функции или модуля согласно заявленным техническим требованиям

Тестирование критического пути (critical path test): основной тип тестовых испытаний, во время которого основные элементы и функции приложения проверяются на предмет правильности работы при стандартном их использовании. В общем случае, тестирование критического пути охватывает основные сценарии использования разрабатываемого приложения.

Расширенное тестирование (Extended Testing) — это подход к тестированию программного обеспечения, при котором проводятся тесты, позволяющие оценить качество продукта на всех уровнях и в различных ситуациях. Одно из видов тестирования, оно используется для проверки работы приложения во всех возможных сценариях использования, учитывая все возможные варианты входных данных и условий работы программы.

Альфа-тестирование — это первоначальное тестирование продукта небольшой группой пользователей.

Бета-тестирование (beta testing): Эксплуатационное тестирование потенциальными и/или существующими клиентами/заказчиками на внешней стороне никак не связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы.

Гамма-тестирование (Gamma Testing). Тип тестирования перед выпуском продукта, направленный на исправление незначительных дефектов, обнаруженных в бета-тестировании. Как правило, выполняется с максимальным привлечением конечных пользователей или заказчика. Также может быть формой внешнего приёмочного тестирования.

Регрессионное тестирование (regression testing). Тестирование уже протестированной программы, проводящееся после модификации для уверенности в том, что процесс модификации не внес или не активизировал ошибки в областях, не подвергавшихся изменениям. Проводится после изменений в коде программного продукта или его окружении. (ISTQB).

Автоматическое end-to-end (E2E) тестирование — это процесс автоматического тестирования с подробной эмуляцией действий пользователя: кликаньем мышки, переходами по страницам, заполнения форм и так далее. Цель E2E тестирования — удостовериться, что программа работает именно так, как задумано для конечного пользователя. Проводится в самом конце перед выкаткой изменений или доработок на рабочие версии продукта

Тестирование производительности - это нефункциональный вид тестирования программного обеспечения, используемый для проверки скорости, времени отклика, стабильности, надежности, масштабируемости и использования ресурсов программного приложения при определенной рабочей нагрузке, обычно регрессионным образом, когда в приложение ежедневно или еженедельно вносятся небольшие изменения.

Нагрузочное тестирование-это тип тестирования производительности, который имитирует реальную нагрузку на любое тестируемое программное обеспечение, приложение или веб-сайт. Он направлен на изучение того, как система будет вести себя в нормальных и экстремальных условиях нагрузки, в основном для того, чтобы определить, может ли программное обеспечение или вычислительное устройство выдерживать высокие нагрузки при высоком спросе пользователе

Стресс-тестирование (иногда называемое тестированием пыток) - это форма преднамеренно интенсивного или тщательного тестирования, используемого для определения стабильности данной системы, критической инфраструктуры или объекта. Она включает в себя тестирование сверх нормальной рабочей мощности, часто до предела, чтобы наблюдать за результатами.

Проверяется корректность инсталляции и деинсталляции программного продукта в среде максимально приближенной к эксплуатационной. Позволяет проверить возможность инсталляции/деинсталляции продукта при различных условиях: новая инсталляция, обновление системы (upgrade), инсталляция по умолчанию, полная инсталляция, выборочная инсталляция.

Тестирование интерфейса определяется как тип тестирования программного обеспечения, который проверяет правильность связи между двумя различными программными системами

Юзабилити-тестирование — это метод оценки интерфейса со стороны удобства и эффективности его использования. Чаще всего оно проводится, когда у страницы, сайта, приложения низкая посещаемость или есть жалобы на проблемы в работе интернет-ресурса. Например, пользователи пишут в поддержку: «Я не могу оформить заказ, товар не отображается в корзине», и мы проводим тестирование, чтобы разобраться с этой жалобой

локализационное тестирование — это проверка содержимого приложения или сайта на соответствие лингвистическим, культурным требованиям, а также специфике конкретной страны или региона.

процедура тестирования программного обеспечения, которая определяет, может ли часть программного обеспечения работать без сбоев в течение определенного периода времени в данной среде. Тестирование надежности гарантирует, что программный продукт не содержит ошибок и способен выполнять свои функции по назначению.

процесс, направленный на выявление недостатков в механизмах безопасности информационной системы, которые защищают данные и поддерживают функциональность по назначению. Из-за логических ограничений тестирования безопасности прохождение процесса тестирования безопасности не является признаком того, что никаких недостатков не существует или что система адекватно удовлетворяет требованиям безопасности.

подвид тестирования производительности, проверяет тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться после возможных сбоев, возникших в связи с ошибками ПО, отказами оборудования или проблемами связи/сети.