

Виды тестирования

Исследовательское (exploratory)

- ad hoc
 - интуитивное, импровизационное, проводится без подготовки к тестам, тестирующая просто проверяет всё, что ему хочется
- scripted (сценарное)
 - проводится по заранее подготовленным тест-кейсам
- monkey test
 - беспорядочное, произвольное тестирование ПО с целью его сломать
- Формализованный ad hoc
 - то же интуитивное тестирование, но с определенными правилами/ограничениями/рамками

Функциональное (functional)

- Smoke test (дымовое)
 - проверка, что приложение стартует и выполняет основные функции
- Critical path (критического пути)
 - проверка функциональности при стандартном использовании приложения/сайта типичным пользователем. Бывает положительный и отрицательный
- Sanity testing (тест работоспособности)
 - проверка функционала БЕЗ внесения изменений/ДО внесения изменений. В том числе, это тестирование документации, требований
- Regression testing (регрессионное)
 - проверка функционала ПОСЛЕ внесения изменений. Его рекомендуется автоматизировать

UI (user interface)

- работа с полями (неизменные и поля для ввода)
- графика
- правильность написания текстов, выравнивание текстов
- вёрстка (расположение элементов)
- реакция элемента при хovere на него
- унификация дизайна (единые цвет, шрифт, размер, поля, иконки, изображения итд)
- tooltips (всплывающие уведомления и подсказки при наведении курсора на элемент)
- вид элементов при уменьшении окна браузера - появление скрола
- правильность перемещения фокуса в окне
- желательно НЕ использовать двойной клик

UX (user expirience)

- восприятие
- удобство
- читаемость
- эргономика
- ожидания пользователя

По знанию внутренних систем

- Black box (чёрный ящик)
 - тестировщик не знает код, идеи для тестирования формируются путём предположений о сценариях (паттернах) поведения пользователя
- White box (белый ящик)
 - проводится либо самими программистами, либо тестирующими с программистской квалификацией того же уровня
- Grey box (серый ящик)
 - Сочетает в себе элементы двух предыдущих подходов

По степени автоматизации

- manual testing (ручное)
 - без использования дополнительных программных средств, т. е. «ручным»
- automated testing (автоматизированное)
 - с использованием программных средств (авто-тесты)
- semi-automated testing (полуавтоматизированное)
 - ручное тестирование с частичным использованием средств автоматизации

Безопасности и контроля доступа (security and access control)

- Конфиденциальность (confidentiality)
 - ограничение доступа или сокрытие определенных ресурсов или информации (шифрование данных, сокрытие пароля астерисками"*)
- Целостность (integrity)
 - информация не может быть повреждена ни случайно, ни умышленно каким-либо пользователем или процессом
- Доступность (availability)
 - доступ к информации должен быть предоставлен только авторизованным пользователям
- Отслеживаемость (traceability)
 - система ведет журнал действий пользователей, реагирует на них

По уровню

- Unit-testing (модульное)
 - проверка корректной работы отдельных элементов системы (модулей). Этот вид тестирования могут выполнять сами разработчики
- Integration testing (интеграционное)
 - оно же комплексное, сборочное. Проверяются объединённые элементы (компоненты или подсистемы) общей системы, взаимодействующие между собой группы элементов в соответствии с архитектурой системы.
- System testing (системное)
 - после того, как система собрана из составляющих компонентов, она проверяется на соответствие системным спецификациям: реализованы ли все функциональные и нефункциональные требования, на данном уровне тестируется приложение или система ЦЕЛИКОМ
- Acceptance testing (приёмочное)
 - завершённое приложение (система в целом) тестируется заказчиком/конечным пользователем с целью определения соответствия системы требованиям заказчика и готовности системы к внедрению

- сверху вниз
 - взяли один элемент, проверили, присоединили еще один, проверили и так далее
- снизу вверх
 - взяли несколько элементов, декомпозировали, проверили
- Big bang (большой взрыв)
 - проверили сразу всё! (высокая вероятность пропуска ошибок, дефектов)

- Alfa-testing (альфа-тест)
 - вид приемочного тестирования, которое обычно проводится на поздней стадии разработки продукта и включает имитацию реального использования продукта штатными разработчиками либо командой тестирующих (альфа-тестеры)
- Beta-testing (бета-тест)
 - работа с реально рабочей версией с полным функционалом с целью оценить возможности и стабильность работы программы с точки зрения ее будущих пользователей перед отправкой в релиз. Здесь бета-тестерами часто выступают обычные пользователи (фокус-группа)

По динамичности

- Static (Статическое)
 - code review - поверхностная проверка кода
 - проверка документации, спецификаций, требований
- Dynamic (Динамическое)
 - запуск кода для функциональных и нефункциональных проверок, цель - убедиться, что ПО работает в соответствии с требованиями.

Инсталляционное (installation)

- Установка
- Обновление
- Переустановка
- Апгрейд (повышение) версии
- Даунгрейд (понижение) версии

Совместимости (compatibility)

- Аппаратная платформа
 - проверяем, чтобы ПО одинаково корректно работало на ПК, ноутбуке, планшете, смартфоне
- Кроссплатформенность
 - проверяем, чтобы ПО одинаково корректно работало на всех операционных системах (MacOS, iOS, Android, Linux, Windows)
- Кроссбраузерность
 - Браузеры на движке WebKit: Safari
 - Браузеры на движке Blink: Chromium
 - Google Chrome
 - Яндекс Браузер
 - Opera
 - Vivaldi
 - Microsoft Edge
 - Браузеры на движке Gecko: Mozilla Firefox
 - Браузеры на движке Trident: Internet Explorer
- Разрешение экрана
 - Веб-сайты должны меняться отзывчиво и быстро при любом разрешении (как 360х640, так и 1920х1080)

Производительности (performance)

- Capacity testing (ёмкости)
 - наделено на проверку максимальной ёмкости системы с точки зрения трафика, при этом обеспечивая оптимальное взаимодействие с пользователем в соответствии с SLA
- Load testing (нагрузочное)
 - оцениваем способность системы правильно функционировать при некотором завышении планируемых нагрузок (наплав пользователей => сайт падает/не падает)
- Stress testing (стрессовое)
 - проверяет, что система реагирует адекватно на стрессовые ситуации (обрыв сети, недостаток дискового пространства и тп)
- Scalability testing (масштабируемости)
 - проверка способности системы "приспосабливаться", расти и расширяться в изменяющихся условиях (увеличение объёмов принимаемых данных, трафика и тп)
- Volume testing (объёмное)
 - определение производительности системы при увеличении объемов данных в базе данных (время выполнения операций в зависимости от количества пользователей)
- Soak/Endurance testing (выносливости)
 - Система тестируется с нагрузкой, растущей в течение значительного времени, для оценки поведения системы при длительном использовании. Целью тестирования состоит в том, чтобы убедиться, что приложение способно выдержать расширенную нагрузку без какого-либо ухудшения времени отклика.
- Resilience testing (устойчивости)
 - Эту форму тестирования иногда также называют chaos engineering. Проверяет отказоустойчивость приложения или способность противостоять стрессовым или сложным факторам, а также выявить его способность адаптироваться к соответствию (compliance), выносливость (endurance), нагрузочное тестирование и тестирование восстановления (recovery testing).
- Stability/reliability testing(стабильности/ надёжности)
 - проверка работоспособности приложения при длительном тестировании с ожидаемым уровнем нагрузки - время ап-тайма или момента с начала запуска - утечка ресурсов (память, процессор, загрузка диска и тп)
- Fallover and recovery testing (отказ и восстановление)
 - проверка способности ПО к сопротивлению и восстановлению после сбоев в работе (упал сервер, выкл электричество)
- Benchmark testing (эталонное)
 - Создание рабочей нагрузки, предназначенной для того, чтобы подвергнуть систему стрессу. цель — изучение поведения системы, воспроизведение желаемого состояния системы или испытание на отказ нового оборудования. Эталонное тестирование может помочь понаблюдать за поведением системы под нагрузкой, определить ее пропускную способность, узнать, какие изменения важны, или посмотреть, как приложение работает с разными данными, позволяет создавать вымышленные обстоятельства в дополнение к реальным условиям, с которыми вы можете столкнуться
- Baseline testing (базовой версии)
 - процесс записи показателей производительности программного приложения при продолжении тестирования производительности, проводится для обеспечения того, чтобы производительность приложения не ухудшалась с течением времени с новыми изменениями, и какие меры должны быть приняты для того, чтобы оно соответствовало базовой производительности. Это гарантирует, что пользовательский опыт не потерян во всех сценариях тестирования, и область улучшения будет обнаружена
- Threshold test (пороговый)
 - тест, вставленный в Deployment Pipeline, который отслеживает некоторые измеримое явление, сравнивая значение в текущей сборке с пороговым значением. Если значение текущей сборки превышает пороговое значение, тест завершается неудачно, и сборка не выполняется.
- Storage testing (хранилища)
 - проверка того, как тестируемое ПО хранит данные в соответствующих каталогах и достаточно ли в них места для предотвращения неожиданного завершения работы из-за недостатка места на диске. ПО должно обрабатывать такие исключения и отображать предупреждающее сообщение для пользователя.

Испытания на выносливость — это длительный процесс, который иногда длится даже до года. Это может включать применение внешних нагрузок, таких как интернет-трафик или действия пользователя. Это отличает испытание на выносливость от нагрузочного тестирования , которое обычно заканчивается через пару часов или около того.

Тестирование устойчивости гарантирует, что ПО может продолжать выполнять основные функции и избежать потери данных даже в критических условиях (при сбое сети, отказе базы данных), при необходимости выдавая соответствующие сообщения об ошибках. При тестировании на устойчивость записываемым результатом является надежность (Reliability). Устойчивость также известна как восстанавливаемость (recoverability). Тестирование устойчивости можно рассматривать как часть плана обеспечения непрерывности бизнеса (BCP - business continuity plan) организации.

Зачем оно нужно? Медленное хранилище = медленное время отклика, длительные запросы и более низкая доступность (availability) приложений. Медленное хранилище - это накладные расходы на обслуживание серверной инфраструктуры. Помогает найти практические ограничения хранилища перед развертыванием. Это помогает понять, как система отреагирует на замену или обновление оборудования

Локализации, глобализации, интернационализации (Localization/ globalization/ internationalization) - перевод и культурная адаптация к особенностям страны/региона

- Globalization (глобализация)
 - Глобализированное ПО - это то ПО, которое одинаково качественно функционирует независимо от географической, культурной и национальной среды. Тестирование глобализации концентрируется на выявлении потенциальных проблем в дизайне продукта, которые могут испортить глобализацию.
- Internationalization (интернационализация)
 - проверка того, может ли программный код работать со всей международной поддержкой без нарушения функциональности, что может привести к потере данных или проблемам целостности информации, корректное редактирование логики всех подключенных параметров форматирования
- Localization (локализация)
 - деятельность по модификации ПО в соответствии с определенными региональными настройками (языком, географической территорией, культурными особенностями), выполнение работ по переводу всего контента программного обеспечения для конечного пользователя.

- Единицы измерения (килограмм/фунт, километр/миля, литр/галлон, цельсий/фarenгейт)
- Формат валюты (рубли, доллар, евро, юань)
- Соответствие формата даты (дд мм гггг/мм дд гггг), времени (15:00/3 AM) и календаря (2023 в России/4720r в Китае/5784r в Израиле) выбранному региону
- Формат телефонных номеров и почтового индекса соответствует выбранному региону
- Текст (направление слева направо, справа налево, сверху вниз), перенос слов, расшифровка аббревиатуры
- Цветовые и числовые ассоциации (в западных странах белый цвет - свадебный, в азиатских - цвет траура, в азиатских домах нет этажа 13, нет ряда 13 на самолете)
- Символика и жесты: в разных странах воспринимаются по-разному
- Изображения должны быть абстрактными (силуэт), чтобы пользователь не ассоциировал себя с картинкой, или использовать изображение с людьми разных национальностей для лояльности пользователей
- Языки (лингвистическая проверка: используются одинаковые термины, отсутствуют грамматические, орфографические, пунктуационные ошибки)
- перевод документации (мануалы, гайды, FAQ, helps итд)
- раскладка клавиатуры
- аудио (озвучка, если таковая имеется)
- Законы (возрастной ценз в России 16+, в США 12+, цензура сленга, запрещенных организаций и тп)

По характеру сценария

- Positive testing (позитивный сценарий)
 - тестирование с применением сценариев, которые соответствуют нормальному (штатному, ожидаемому) поведению системы, с его помощью мы можем определить, что система делает то, для чего и была создана
- Negative testing (негативный сценарий)
 - тестирование, при котором выполняются некорректные операции или используются данные, потенциально приводящие к ошибкам