

Виды тестирования

Исследовательское тестирование

- Ad hoc
- Сценарное тестирование
- Monkey testing
- Формализованный Ad hoc

Никаких требований и сценариев не имеется, исследуем так, как понимаем (интуитивное тестирование)

Имеются тест-кейсы, которые помогают протестировать

Производятся любые случайные нажатия кнопок, ввод случайных данных, что помогает оценить даст ли при этом система сбой( рандомное тестирование)

Существуют какие-либо формализованные действия, которые необходимо выполнить

Включает минимальный набор тестов на самые очевидные ошибки, проверяет работоспособность критически важных функциональных частей программы

Основные элементы и функции приложения проверяются на предмет правильности работы при стандартном их использовании (позитивное тестирование, антипод негативного тестирования)

Проверка существующего функционала после внесения изменений.

Тестируется как работает старый функционал до внедрения нового

Функциональное тестирование

- Дымовое тестирование(Smoke)
- Тестирование критического пути
- Регрессионное тестирование
- Санитарное тестирование

Нефункциональное тестирование

UI (User Interface) - обозначает тестирование пользовательского интерфейса

UX (User Experience) - это практика тестирования того, насколько легко использовать дизайн

- шрифт
- графика
- изображения
- цвета
- подсказки/уведомления
- и т.д.
- удобство
- читаемость
- восприятие
- доступность (для людей с ограниченными возможностями)
- и т.д.

все, что можно визуально посмотреть, протестировать

составляющие необходимы для выявления проблемных мест и точек роста интерфейса

По методам тестирования

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

Код виден и его можно изменять

Код не виден

Комбинация метода черного и белого ящика(возможность более сложных сценариев тестирования)

По степени автоматизации

- Автоматизированное (часто при регрессионном)
- Ручное
- Полуавтоматическое

Тестирование безопасности

Выясняется закрывается ли пароль при вводе, шифруются ли данные, оценивается уязвимость к атакам, определяется легко ли неавторизованный пользователь может получить доступ к данным и т.д.

Интеграционное тестирование

- Модульное тестирование
- Сверху вниз
- Снизу вверх
- Большой взрыв

Тестирование отдельных компонентов (модулей) системы

Сначала проверяется работа крупных модулей, спускаясь ниже, постепенно разбирая проверяются уровни ниже( пирамида разбирается)

Все мелкие части модуля собираются в один модуль и тестируются. Далее собираются следующие мелкие модули в один большой и модуль тестируется с предыдущим и т.д.( пирамида растёт)

Большинство разработанных модулей соединены друг с другом, чтобы сформировать полную систему программного обеспечения, или основную часть системы ( проводится тестирование взаимодействия)

Инсталляционное тестирование

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

Тестирование совместимости

- аппаратная составляющая
- кроссплатформенность
- кроссбраузерность
- адаптивность под моб.версию

Системные требования

Способность ПО работать с несколькими операционными системами

Способность веб-ресурса отображаться одинаково хорошо во всех популярных браузерах без перебоев

Работает ли ПО на телефоне, не используя мобильное приложение этого ПО

Тестирование производительности

- Нагрузочное
- Стрессовое
- Масштабируемости
- Отказ и восстановление (в т.ч. перфоманс)

Дается нагрузка больше, чем может в теории выдержать программа

Дается нагрузка на грани возможной теоретически

Проводится для определения способности приложения масштабироваться с точки зрения пользовательской нагрузки, количества транзакций, объема данных и т. д.

Проверяется тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться после возможных сбоев

Тестирование с учетом формата времени, формата даты, языка(правильность перевода интерфейса), учет использования цветов, и т.д.- насколько продукт может быть адаптирован к иностранному рынку