

Виды тестирования

исследовательское

- ad hoc (интуитивное, ид тестирования, который выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.)
- сценарное
- monkey test
- ad hoc формализованный

функциональное

- смоки (проверка программного обеспечения на стабильность и наличие явных ошибок.)
- критического пути (весь функционал обрабатывает верно, без дополнительных условий)
- регресс (проверяется то, что исправление багов, а также любые изменения в коде приложения, не повлияли на другие модули ПО и не вызвали новых багов.)
- санитарное

нефункциональное

- UI user interface
 - шрифты (соответствие дизайну)
 - графика
 - изображения
 - подсказки/уведомления
 - размеры кнопок и тд
- UX user experience
 - восприятие
 - удобство
 - читабельность

по методам

- черного ящика Не видим код
- белого ящика Видим код
- серого ящика Комбинация Ч и Б

по степени автоматизации

- ручное тестирование
- автотесты
- полуавтоматические/гибридные

безопасность

- возможно/невозможно ли вставить вредоносный код в окно логин/пароль, например

интеграционное тестирование

- модульное тестирование (эквайринг, капчи и тд)
- сверху вниз ((Top Down Integration) Вначале тестируются все высокоуровневые модули, и постепенно один за другим добавляются низкоуровневые. Все модули более низкого уровня симулируются заглушками с аналогичной функциональностью, затем по мере готовности они заменяются реальными активными компонентами.)
- снизу вверх (Bottom Up Integration) Все низкоуровневые модули, процедуры или функции собираются воедино и затем тестируются. После чего собирается следующий уровень модулей для проведения интеграционного тестирования. Данный подход считается полезным, если все или практически все модули, разрабатываемого уровня, готовы. Также данный подход помогает определить по результатам тестирования уровень готовности приложения.)
- большой взрыв big bang (Все или практически все разработанные модули собираются вместе в виде законченной системы или ее основной части, и затем проводится интеграционное тестирование)

инсталяционное тестирование

- установка
- обновление
- переустановка
- upgrade повышение версии
- downgrade понижение версии

совместимость

- аппаратная составляющая (работает ли под какой-либо архитектурой наше ПО - системные требования)
- кроссплатформенность (андроид, ios, windows, linux, mac os)
- кроссбраузерность
- адаптивность под мобильную версию (например, у человека нет приложения и он хочет зайти на сайт через телефон)

нагрузочное

- стресс-тест (позволяет проверить насколько приложение и система в целом работоспособны в условиях стресса (например, повышение интенсивности выполнения операций до очень высоких значений или аварийное изменение конфигурации сервера) и также оценить способность системы к регенерации, т.е. к возвращению к нормальному состоянию после прекращения воздействия стресса)
- масштабируемость (измеряет производительность сети или системы, когда количество пользовательских запросов увеличивается или уменьшается.)
- перформанс (определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок.)
- отказ и восстановление (проверяет тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться, т.е. обеспечивать сохранность и целостность данных, после возможных сбоев, возникших в связи с ошибками программного обеспечения, отказами оборудования или проблемами связи (например, отказ сети).)

локализация

- проверка адаптации программного обеспечения для определенной аудитории в соответствии с ее культурными особенностями.

системное

- роверка как функциональных, так и не функциональных требований в системе в целом. При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных пользовательского уровня, несовместимость с окружением, непредусмотренные сценарии использования и т.д., и оцениваются характеристики качества системы — ее устойчивость, надежность, безопасность и производительность.

операционное

- тестирование в среде эксплуатации позволяет выявить и нефункциональные проблемы, такие как: конфликт с другими системами, смежными в области бизнеса или в программных и электронных окружениях и др. О

по исполнению кода

- статическое (Например, путем анализа кода (code review). Анализ может производиться как вручную, так и с помощью специальных инструментальных средств. Целью анализа является раннее выявление ошибок и потенциальных проблем в продукте.)
- динамическое (тестирование проводится на работающей системе, т.е. с осуществлением запуска программного кода приложения.)