

Нефункциональное тестирование системы выполняется для оценки таких характеристик системы и программного обеспечения, как удобство использования, производительность или безопасность. Нefункциональное тестирование – это проверка того, «насколько хорошо работает система». Ниже перечислены виды нефункционального тестирования и какие проверки они подразумевают.

Тестирование установки (инсталляционное тестирование) позволяет удостовериться в том, что ПО корректно устанавливается и настраивается, накат новых версий происходит без ошибок, а также есть возможность деинсталлировать и удалить данное ПО. Тестирование инсталляции необходимо проводить при создании ПО, после появления новой версии, а также при изменении конфигурации стенда.

Юзабилити-тестирование (Usability – удобство) – это проверка программного продукта на соответствие с требованиями в плане удобства использования приложения. Таким образом, с помощью юзабилити-тестирования мы можем определить эргономичность (приспособленность к использованию) программы.

Проверка юзабилити приложения заключается в:

- Оценка соответствия дизайна приложения к его функциональности, заданной заказчиком.
- Анализ используемых графических элементов, цветового оформления с точки зрения восприятия.
- Оценке удобства навигации и ссылочной структуре.
- Анализ текстового наполнения.
- Оценка удобства использования функциями приложения (сервисами, если это сайт).
- Анализ шрифтового оформления текста.

Тестирование на отказ и восстановление (Failover and Recovery Testing) проверяет тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться после возможных сбоев, возникших в связи с ошибками программного обеспечения, отказами оборудования или проблемами связи (например, отказ сети).

Целью данного вида тестирования является проверка систем восстановления (или дублирующих основной функционал систем), которые, в случае возникновения сбоев, обеспечат сохранность и целостность данных тестируемого продукта.

Тестирование на отказ и восстановление очень важно для систем, работающих по принципу “24x7”. Если Вы создаете продукт, который будет работать, например в интернете, то без проведения данного вида тестирования Вам просто не обойтись, так как каждая минута простоя или потеря данных в случае отказа оборудования, может стоить вам денег, потери клиентов и репутации на рынке.

Методика подобного тестирования заключается в симулировании различных условий сбоя и последующем изучении и оценке реакции защитных систем. В процессе подобных проверок выясняется, была ли достигнута требуемая степень восстановления системы после возникновения сбоя.

Для наглядности рассмотрим некоторые варианты подобного тестирования и общие методы их проведения. Объектом тестирования в большинстве случаев являются весьма вероятные эксплуатационные проблемы, такие как:

- Отказ электричества на компьютере-сервере
- Отказ электричества на компьютере-клиенте
- Незавершенные циклы обработки данных (прерывание работы фильтров данных, прерывание синхронизации).
- Объявление или внесение в массивы данных невозможных или ошибочных элементов.
- Отказ носителей данных.

Данные ситуации могут быть воспроизведены, как только достигнута некоторая точка в разработке, когда все системы восстановления или дублирования готовы выполнять свои функции. Технически реализовать тесты можно следующими путями:

- Симулировать внезапный отказ электричества на компьютере (обесточить компьютер).
- Симулировать потерю связи с сетью (выключить сетевой кабель, обесточить сетевое устройство)
- Симулировать отказ носителей (обесточить внешний носитель данных)

- Симулировать ситуацию наличия в системе неверных данных (специальный тестовый набор или база данных).

При достижении соответствующих условий сбоя и по результатам работы систем восстановления, можно оценить продукт с точки зрения тестирования на отказ. Во всех вышеперечисленных случаях, по завершении процедур восстановления, должно быть достигнуто определенное требуемое состояние данных продукта:

- Потеря или порча данных в допустимых пределах.
- Отчет или система отчетов с указанием процессов или транзакций, которые не были завершены в результате сбоя.

Стоит заметить, что тестирование на отказ и восстановление – это весьма продукт-специфичное тестирование. Разработка тестовых сценариев должна производиться с учетом всех особенностей тестируемой системы. Принимая во внимание довольно жесткие методы воздействия, стоит также оценить целесообразность проведения данного вида тестирования для конкретного программного продукта.

Тестирование совместимости - вид тестирования, который позволяет проверить работоспособность программной системы в условиях различных операционных систем, аппаратных и программных конфигураций.

Исходя с определения, можно выделить два уровня проведения тестирования совместимости – клиентский и серверный.

Клиентский. Приложение тестируется с позиции рабочего окружения конечного пользователя. А именно:

- Кроссплатформенное тестирование (типы и версии ОС).
- Кроссбраузерное тестирование (используется, при тестировании веб-приложения).
- Тестирование работы при различных версиях драйверов.
- При тестировании игровых приложений – тестирование видеоадаптера.

Если же приложение клиент-серверное, необходимо протестировать взаимодействие приложение с окружением:

- Аппаратным (тип и количество процессоров, объем памяти, характеристики сети / сетевых адаптеров и т.д.).

- Программным (ОС, драйвера и библиотеки, стороннее ПО, влияющее на работу приложения и т.д.).

Тестирование «интернационализации и локализации» – это процесс придания продукту свойств определенной народности, местности, расположения. Для успешной реализации продукта во все необходимые нам страны, нужно уделять внимание не только техническому переводу элементов интерфейса на язык страны-покупателя. Само значение слов, технологий, размещения кнопок, текстовых полей, изображений не должно хоть как-то затрагивать чью-либо религию или культуру. Не должно возникнуть проблем с восприятием конечным клиентом нашего продукта. И область программного обеспечения попадает под данные правила.

Локализация – процесс адаптации программного продукта к языку и культуре клиента. Данный процесс адаптации включает в себя:

- Перевод пользовательского интерфейса.
- Перевод документации.
- Контроль формата даты и времени.
- Внимание к денежным единицам.
- Внимание к правовым особенностям.
- Раскладка клавиатуры пользователя.
- Контроль символики и цветов.
- Толкование текста, символов, знаков и т.д.

Кроме этого, в некоторых случаях повышенного внимания к соблюдению требований локализации, возможно влияние на проектирование приложения.

Интернационализация – более обобщенное понятие, подразумевающее проектирование и реализацию программного продукта или документации таким образом, который максимально упростит локализацию приложения.

Интернационализация включает в себя:

- Создание продукта с учетом возможности кодировки Unicode (стандарт кодирования, поддерживающий практически все языки мира).
- Создание в приложении возможности поддержки элементов, которые невозможно локализовать обычным образом (вертикальный текст азиатских стран, чтение с права на лево арабских стран и т.д.). Возможность загрузки локализованных элементов в будущем при желании пользователя.

Тестирование производительности – это комплекс типов тестирования, целью которого является определение работоспособности, стабильности, потребления ресурсов и других атрибутов качества приложения в условиях различных сценариев использования и нагрузок. Тестирование производительности позволяет находить возможные уязвимости и недостатки в системе с целью предотвратить их пагубное влияние на работу программы в условиях использования. Необходимые параметры работы системы в определенной среде можно тестировать с помощью:

- Определения рабочего количества пользователей приложения.
- Измерение времени выполнения различных операций системы.
- Определения производительности приложения при различных степенях нагрузки.
- Определения допустимых границ производительности программы при разных уровнях нагрузки.

В зависимости от характеристик, которые нам нужно протестировать, **тестирование производительности** делится на типы:

Нагрузочное тестирование (Loadtesting) – тестирование времени отклика приложения на запросы различных типов, с целью удостовериться, что приложение работает в соответствии с требованиями при обычной пользовательской нагрузке.

Стресс-тестирование (Stresstesting) – тестирование работоспособности приложения при нагрузках, превышающих пользовательские в несколько раз. При стресс-тестировании (зачастую, только при нем) мы можем получить реальные данные границ производительности приложения, исследовать способность программы обрабатывать исключения, ее стабильность и устойчивость. Именно в значительно увеличенной нагрузке на приложение и заключается разница между тестированием производительности и стресс тестированием.

Тестирование стабильности или наработка на отказ (Stability/Reliability testing) исследует работоспособность приложения при длительной работе во времени, при нормальной для программы нагрузке.