О чём это всё

Будучи инженером по тестированию, вы, вероятно, слышали о таких видах тестирования как «дымовое» (smoke), «санитарное тестирование» (sanity), «ре-тест» и регрессионное тестирование. Вполне возможно, многие из этих видов используются вами на ежедневной основе.  
  
В этой статье я хотел бы внести ясность и объяснить разницу между этими видами тестирования и попробовать разобраться, провести границы (хоть и условные) где заканчивается один вид тестирования, и начинается другой.  
  
Для новичков в тестировании (и даже опытных тестировщиков) разделение этих понятий может быть затруднительно. И в самом деле, как отличить где начинается санити-тестирование и заканчивается smoke? Насколько сильно нам надо ограничить проверку части функциональности системы или её компонентов, чтобы назвать это «дымовым» тестированием? Является ли ввод логина/пароля в пользовательскую форму входа на сайт дымовым тестом, или сам факт её появления на странице сайта уже является пройденным тестом?  
  
Строго говоря, вы всё равно сможете проводить тестирование, даже при том что не сможете точно сказать, в чём же разница. Можно даже не задумываться о разграничении, каким именно видом тестирования вы сейчас заняты. Но всё же, чтобы расти над собой в профессиональном смысле, нужно знать что вы делаете, зачем, и насколько правильно вы это делаете.

Ликбез

Ниже приведены краткие определения видов тестирования, которые мы сегодня сравниваем:

* **Дымовые тесты**: выполняются каждый раз, когда мы получаем новый билд (версию), проекта (системы) на тестирование, при этом считая её относительно нестабильной. Нам нужно убедиться что критически важные функции AUT (Application Under Test) работают согласно ожиданиям. Идея данного вида тестирования заключается в том, чтобы выявить серьёзные проблемы как можно раньше, и отклонить этот билд (вернуть на доработку) на раннем этапе тестирования, чтобы не углубляться в долгие и сложные тесты, не затрачивая тем самым время на заведомо бракованное ПО.
* **Санитарное тестирование**: используется каждый раз, когда мы получаем относительно стабильный билд ПО, чтобы определить работоспособность в деталях. Иными словами, здесь проходит валидация того, что важные части функциональности системы работают согласно требованиям на низком уровне.

Оба эти вида тестирования нацелены на то, чтобы избежать потерь времени и усилий, чтобы быстрее определить недостатки ПО и их критичность, а так же то, заслуживает ли оно перехода в фазу более углублённого и тщательного тестирования или же нет.

* **Ре-тест**: проводится в случае, если фича/функциональность уже имела дефекты, и эти дефекты были недавно исправлены
* **Регрессионные тесты**: собственно то, что занимает львиную долю времени и для чего существует автоматизация тестирования. Проводится регрессионное тестирование AUT тогда, когда нужно убедиться что новые (добавленные) функции приложения / исправленные дефекты не оказали влияния на текущую, уже существующую функциональность, работавшую (и протестированную) ранее.

Для лучшего понимания ниже представлена сравнительная таблица этих понятий и области применения:

|  |  |  |  |
| --- | --- | --- | --- |
| **Дымовые (Smoke)** | **Санити (Sanity)** | **Регрессионные (Regression)** | **Ре-тест (Re-test)** |
| Исполняются с целью проверить что критически важные функциональные части AUT работают как положено | Нацелено на установление факта того, что определённые части AUT всё так же работают как положено после минорных изменений или исправлений багов | Подтверждают, что свежие изменения в коде или приложении в целом не оказали негативного влияния на уже существующую функциональность/набор функций | Перепроверяет и подтверждает факт того, что ранее заваленные тест-кейсы проходят после того, как дефекты исправлены |
| Цель — проверить «стабильность» системы в целом, чтобы дать зелёный свет проведению более тщательного тестирования | Целью является проверить общее состояние системы в деталях, чтобы приступить к более тщательному тестированию | Цель — убедиться что свежие изменения в коде не оказали побочных эффектов на устоявшуюся работающую функциональность | Ре-тест проверяет что дефект исправлен |
| Перепроверка дефектов не является целью Smoke | Перепроверка дефектов не является целью Sanity | Перепроверка дефектов не является целью Regression | Факт того что дефект исправлен подтверждает Re-Test |
| Дымовое тестирование выполняется **перед** регрессионным | Санитарное тестирование выполняется **перед** регрессионным и **после** smoke-тестов | Проводится на основании требований проекта и доступности ресурсов (закрывается автотестами), «регресс» может проводиться в параллели с ре-тестами | — Ре-тест выполняется перед sanity-тестированием — Так же, приоритет ре-теста выше регрессионных проверок, поэтому должно выполняться перед ними |
| Может выполняться автоматизированно или вручную | Чаще выполняется вручную | Лучший повод для автоматизации данного вида тестирования, т.к. ручное может быть крайне затратным по ресурсам или времени | Не поддаётся автоматизации |
| Является подмножеством регрессионного тестирования | Подмножество приёмочного тестирования | Выполняется при любой модификации или изменениях в уже существующем проекте | Ре-тест проводится на исправленной сборке с использованием тех же данных, на том же окружении, но с различным набором входных данных |
| Тест-кейсы часть регрессионных тест-кейсов, но покрывающие крайне критичную функциональность | Санитарное может выполняться без тест-кейсов, но знание тестируемой системы обязательно | Тест-кейсы регрессионного тестирования могут быть получены из функциональных требований или спецификаций, пользовательских мануалов, и проводятся вне зависимости от того, что исправили разработчики | Используется тот же самый тест-кейс, который выявил дефект |

Ну а по существу?

Приведу пример разграничения понятий на моём текущем проекте.  
  
Пример: у нас есть веб-сервис с пользовательским интерфейсом и RESTful API. Будучи тестировщиками, мы знаем:

* Что у него есть 10 точек входа, для простоты, в нашем случае расположенных на одном IP
* Мы знаем все они принимают GET-запрос на вход, возвращая какие-либо данные в формате json.

Тогда можно сделать ряд утверждений о том, какие типы тестов нужно использовать в какой момент времени:

* Выполнив один простой GET-запрос к одной из этих точек входа, и получив ответ в формате json, мы уже убеждаемся что дымное тестирование пройдено.  
  Если же одна из этих точек входа так же возвращает данные из БД, тогда как первая — нет, нужно дополнительно выполнить ещё один запрос, чтобы убедиться что приложение  
  верно обрабатывает запросы к базе. И на этом «дымный» тест закончен.  
    
  То есть мы выполнили запрос — от сервиса пришёл ответ, и он не «задымился», то есть не вернул ошибку 4хх или 5хх, и что-то невнятное, вместо json. На этом можно сказать что «дымный» тест пройден. Для проверки того, что работает так же и UI достаточно просто один раз открыть страницу в браузере.
* Санитарное тестирование в данном случае будет состоять из выполнения запроса ко всем 10 точкам входа в api, сверкой полученного json с ожидаемым, а так же наличием требуемых данных в нём.
* Регрессионные тесты будут состоять из smoke + sanity + UI выполняемые вместе в одной куче. Цель: проверить что добавление 11-ой точки входа не поломало, к примеру, восстановление пароля.
* Ре-тест в данном примере это точечная проверка что, к примеру, сломавшаяся точка входа в api в следующем билде отрабатывает как задумывалось.

При этом, если это api принимает так же post-запросы, то очевидно что в другой набор тестов sanity нужно включить именно эти запросы. По аналогии с UI мы будем проверять все страницы приложения.

Подведём итог

Надеюсь, что после чтения данной статьи, у вас появится ясность в определении какой вид тестирования вы используете на каком этапе, и в чём разница между этими видами тестирования. Как и было упомянуто вначале, граница между этими понятиями весьма условная и остаётся на ваше усмотрение в рамках проекта.  
  
**UPD**:  
Часто «тестирование согласованности» или «тестированием на вменяемость», называют термином «санитарное тестирование». Думаю что это пошло из-за фонетических свойств английского слова sanity, схожего по звучанию с чем-то «санитарным». [Гугл транслейт вносит ясность](https://translate.google.ru/?hl=ru#en/ru/sanity). В интернете встречаются оба варианта. Относительно данной статьи прошу считать «санитарное» тестирование как «тестирование на согласованность».

* **Чек-лист** — это список того, что нужно проверить. Например, можно составить чек-лист для проверки сайта или отдельного его компонента — скажем, личного кабинета или корзины.
* **Тест-кейс** — это пошаговое описание того, как мы будем тестировать ту или функцию. Например, если это личный кабинет на сайте, в тест-кейсе будут прописаны конкретные действия: зайти на сайт фирмы «Рога и копыта», ввести логин и пароль, нажать кнопку «Войти» и так далее.