



stachevm

29 окт в 20:17

Как мне захотелось систематизировать виды тестирования

Простой 11 мин 12K

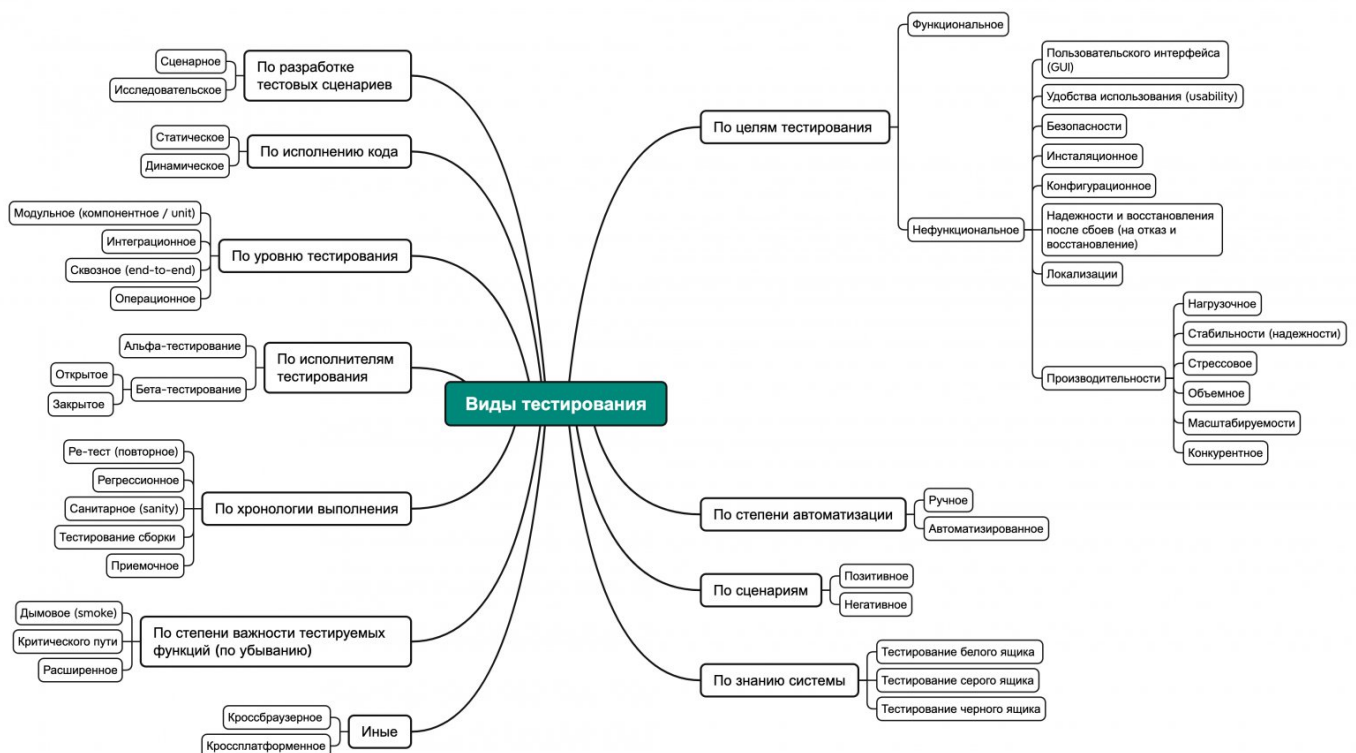
Тестирование IT-систем*, Тестирование веб-сервисов*, Тестирование мобильных приложений*

Из песочницы

Привет, Хабр! Я, как и все люди, которые начали изучать тестирование, столкнулся с темой "Виды тестирования". Погружаясь в эту тему, я обнаружил множество различных классификаций и схем, которые иногда сильно отличались друг от друга. Это вдохновило меня на идею систематизации видов тестирования и создания общей схемы.

В процессе подготовки, я дал определение для каждого вида, а также добавил краткое описание и примеры, чтобы облегчить понимание сути каждого вида и выделить их отличия друг от друга.

В этой статье я собрал различные фрагменты информации по теме видов тестирования из разных источников в интернете, иногда переформулировал определения и теперь готов поделиться этим всем с вами.



Виды тестирования

Итак, ниже приведены классификации...

По целям тестирования

1. **Функциональное** — вид тестирования, при котором проверяются функциональные требования ПО, то есть способность ПО решать возложенные на него задачи. Направлено на проверку корректности работы функциональности приложения. Например, в приложении интернет-магазина при нажатии на кнопку "Добавить в корзину", товар добавляется в корзину пользователя.

2. **Нефункциональное** — вид тестирования, который проверяет нефункциональные аспекты ПО.

- **Пользовательского интерфейса (GUI)** — тестирование интерфейса ПО на соответствие требованиям (размер, шрифт, цвет и т. д.).
- **Удобства использования (usability)** — тестирование, направленное на установление степени удобства использования, понятности и привлекательности для пользователей разрабатываемого продукта. При тестировании нужно ответить на вопросы:
 - Как быстро пользователь достигнет цели?
 - Размер кнопок (удобно ли попадать)?
 - и др.
- **Безопасности** — тестирование защищенности системы, анализ рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным. Например, тестировщик пытается взломать приложение, чтобы проверить, получится ли обойти защиту и получить данные пользователей.
- **Инсталяционное** — тестирование установки, обновления и удаления приложения.

Например, нужно протестировать будет ли обновляться приложение с 1-й версии сразу на 3-ю версию, а не только со 2-й версии на 3-ю.

- **Конфигурационное** — тестирование, направленное на проверку работы ПО при различных конфигурациях системы (заявленных платформах, поддерживаемых драйверах, при различных конфигурациях компьютеров и т. д.).
- **Надежности и восстановления после сбоев (на отказ и восстановление)** — тестирование 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, работая в текстовом редакторе, при потере сети, отключении электричества на стороне клиента / сервера, данные должны сохраниться.

- **Локализации** — тестирование программного обеспечения на соответствие лингвистическим, культурным требованиям, а также специфике конкретной страны или региона (язык, валюта, система мер и т. д.). Например, в Китае в играх не должно быть изображений мертвых тел или крови.
- **Производительности** — определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок.
 - **Нагрузочное** — тестирование работы системы под разными уровнями нагрузки до предельного значения, которое она должна выдерживать и которые содержатся в требованиях.

При таком тестировании измеряется время отклика системы и скорость обработки запросов от пользователей (например, как быстро открываются и прогружаются страницы сайта, как быстро система выполняет расчеты, выдает результаты поиска и т. д.), а также сколько ресурсов "съедает" система — сетевых, процессорных, памяти.

- **Стабильности (надежности)** — тестирование работоспособности приложения со средним уровнем нагрузки, но при длительном времени работы.
- **Стрессовое** — тестирование системы при нагрузке выше нормы, описанной в требованиях.

Например, сайт рассчитан на одновременную работу 5 тысяч пользователей, стрессовое тестирование предполагает, что на сайте будет работать более 5 тысяч пользователей одновременно.

- **Объемное** — тестирование, которое проводится для получения оценки производительности при увеличении объемов данных в базе данных приложения.

Объемное тестирование предназначено для прогнозирования того, может ли система обрабатывать большой объем данных в плане проверки объема данных, обрабатываемых базой данных. Например, при записи большого объема данных нет сообщений об ошибках / крашах системы, а данные записываются в БД.

- **Масштабируемости** — тестирование производительности системы с точки зрения способности масштабироваться (приспосабливаться), реагируя на увеличение количества пользователей, запросов, и других динамических параметров системы.

Цель такого тестирования — оценить, насколько хорошо масштабируется система в ответ на различные уровни нагрузки. Например, компания ожидает шестикратного увеличения нагрузки на серверы в течение следующих двух месяцев. Им может потребоваться увеличить производительность сервера и сократить время обработки запроса, чтобы лучше обслуживать посетителей. Если приложение масштабируемое, вы можете сократить это время, обновив оборудование сервера, например, вы можете увеличить частоту процессора и добавить больше оперативной памяти.

- **Конкурентное** — тестирование поведения системы в момент, когда одновременно происходят два или более событий, или выполняется одновременный вход нескольких пользователей в систему с выполнением одного и того же действия.

Например, при одновременной авторизации нескольких пользователей в систему, все они должны авторизоваться, а система не выдать ошибок.

По степени автоматизации

1. **Ручное** — тип тестирования, в котором проверки выполняются тестировщиком вручную, без использования инструментов автоматизации.
2. **Автоматизированное** — тип тестирования, при котором проверки выполняются с использованием программных средств для выполнения тестовых сценариев. Например, тестировщик пишет код на JavaScript для автоматизации процесса авторизации на сайте.

По сценариям

1. **Позитивное** — тестирование, при котором ПО или его элементы реагируют корректно (согласно требованиям) на совершенные действия при использовании корректных тестовых данных. Например, в окне регистрации на сайте в поле "имя" можно ввести от 2-х до 20-ти букв. Позитивной проверкой будет ввод имени из 4-х букв — Иван и система даст пройти дальше для заполнения следующих полей.
2. **Негативное** — вид тестирования ПО, направленный на проверку того, что система или приложение ведут себя должным образом (согласно требованиям) в негативных

ситуациях, то есть, когда они получают недопустимые или неожиданные входные данные.

Например, при заказе товара в интернет-магазине в поле "номер телефона" можно ввести только цифры. При вводе других символов система должна выдавать ошибку и не дать оформить заказ.

По знанию системы

1. **Белого ящика** — метод тестирования ПО, который предполагает полный доступ тестировщика к коду системы.
Тестировщик изучает реализацию кода поля ввода на веб-странице, определяет все предусмотренные (как правильные, так и неправильные) и не предусмотренные пользовательские вводы, и сравнивает фактический результат выполнения программы с ожидаемым. При этом ожидаемый результат определяется именно тем, как должен работать код программы.
2. **Серого ящика** — метод тестирования ПО, который предполагает частичный доступ к коду проекта, комбинация методов белого и черного ящиков.
Например, тестирование API с помощью Postman или работа с базой данных.
3. **Черного ящика** — метод тестирования ПО, основанный на спецификации, который не предполагает доступа (полного или частичного) к системе, т. е. основывается на работе исключительно с внешним интерфейсом тестируемой системы.
Например, тестирование приложения по макетам или по описанию функциональности в требованиях.

По разработке тестовых сценариев

1. **Сценарное** — тестирование по заранее подготовленной тестовой документации. Такое тестирование проводится если:
 - требования к продукту подробно зафиксированы;
 - у тестировщика достаточно времени, чтобы подготовить тестовую документацию заранее.
2. **Исследовательское** — вид тестирования, при котором тестовую документацию составляют по ходу проверки сервиса или приложения, а не заранее. Исследовательское тестирование применяют, если:
 - требования к продукту не прописаны или много серых зон;
 - недостаточно времени, чтобы составить тестовую документацию заранее.

По исполнению кода

1. **Статическое** — тип тестирования, который предполагает, что программный код во время тестирования не будет выполняться.
Например, вычитка документации приложения, проверка синтаксиса кода.
2. **Динамическое** — тип тестирования, который подразумевает запуск кода для проведения функциональных и нефункциональных проверок ПО.

По уровню тестирования

1. **Модульное (компонентное / unit)** — вид тестирования, при котором проверяется отдельная часть приложения.

Например, в приложении такси нужно проверить функциональность "Поделиться поездкой" — это модульное тестирование.

2. **Интеграционное** — вид тестирования, направленный на проверку корректного взаимодействия нескольких компонентов системы между собой.

Например, в приложении такси нужно протестировать, подтягивается ли адрес в поле "Место назначения" по клику в любую точку на карте.

3. **Сквозное (end-to-end)** — вид тестирования, при котором проверяется система целиком.

Например, пользователь хочет купить билет в театр в онлайн-кассе. Если нужно провести сквозное тестирование, нужно проверить, что он может сделать всё: от выбора места до отправки чека о покупке на электронную почту.

4. **Операционное** — процесс проверки системы на удовлетворение всех потребностей пользователя и соответствия бизнес-требованиям.

Суть заключается в том, чтобы проверить функционал в его естественной среде, где он должен быть. Например, если какое-то приложение предназначалось для бухгалтера, то на этом уровне тестирования — приложение будет тестироваться непосредственно бухгалтерами. То есть это самый заключительный этап проверки функционала непосредственно теми людьми, кто будет его использовать.

По исполнителям тестирования

1. **Альфа-тестирование** — тестирование ПО, проводимое на ранней стадии разработки, внутри организации-разработчика, которое имитирует реальное использование продукта.
2. **Бета-тестирование** — тестирование практически готового ПО, выпускаемое

для ограниченного количества пользователей, разрабатывается в первую очередь для тестирования конечными пользователями и получения отзывов клиентов о продукте для внесения соответствующих изменений.

- **Открытое** — попадают все желающие.
- **Закрытое** — для ограниченного количества пользователей, попадают по приглашениям или иным способам.

По хронологии выполнения

1. **Ре-тест (повторное)** — проведение повторной проверки, при которой ранее был выявлен дефект и отправлен на исправления с целью проверить, что дефект исправлен.

Например, на новостном сайте добавили новый раздел, но кнопка для перехода в него не работает, и пользователь не может попасть в этот раздел. Тестировщик заводит баг-репорт, затем разработчик исправляет проблему, и после этого тестировщик проверяет, что ошибка действительно устранена.

2. **Регрессионное** — тестирование после внесения изменений в код приложения, для подтверждения факта, что эти изменения не внесли ошибки в областях, которые не подверглись изменения, то есть проверка, что ничего из старой функциональности не сломалось.

Например, в сервис для построения маршрутов для разных видов транспорта добавили новый вид транспорта — вертолёт. В процессе регресса обнаружилось, что:

- поехала вёрстка у старых видов транспорта;
- сломалась возможность их переключать;
- сбросились расчёты для старых видов транспорта.

Регрессионное тестирование помогает обнаружить ошибки вовремя.

3. **Санитарное (sanity)** — узконаправленное тестирование отдельных функциональных элементов системы.

Является подвидом регрессионного тестирования. При санитарном тестировании проверяется небольшой объем кода, отвечающий за одну функцию и тщательно исследуется. Smoke-тесты, в отличие от санитарных, проверяют только критически-важный функционал.

Например, при регистрации на сайте добавили новое поле "Семейное положение",

санитарное тестирование предполагает тестирование этого добавленного поля.



Место санитарных тестов

4. **Тестирование сборки** — вид тестирования направленный на определение соответствия, выпущенной версии ПО, критериям качества перед релизом. Например, команда разработчиков внесла ряд изменений в код приложения, чтобы добавить новые функции и исправить ошибки. После того, как все изменения были внесены, была создана сборка приложения. Теперь перед выпуском релиза необходимо провести тестирование сборки.
5. **Приемочное** — вид тестирования, проводимый на этапе сдачи готового продукта (или готовой части продукта) заказчику.

Оценивается соответствие продукта бизнес-требованиям и требованиям пользователей. Это финальный этап тестирования продукта перед его релизом. При этом, он не является сверх тщательным, всеохватывающим и полным — тестируется, в основном, только основной функционал. Приемочное тестирование проводится либо самим заказчиком, либо группой тестировщиков, представляющих интересы заказчика, либо тестировщиками компании-разработчика. Зависит от предпочтений компании-заказчика.

По степени важности тестируемых функций (по убыванию)

1. **Дымовое (smoke)** — тестирование только критически важного функционала, неработоспособность которого делает бессмысленной саму идею использования приложения.

Например, в приложении интернет-магазина можно зарегистрироваться, авторизоваться под уже зарегистрированным аккаунтом, добавить товар в корзину, сделать заказ товара и т. д.

2. **Критического пути** — тестирование, направленное на исследование функциональности, используемой типичными пользователями в типичной повседневной деятельности.

Если дымовое тестирование прошло успешно, то на уровне критического пути должны быть проверены все функции, необходимые пользователю для успешной работы с приложением и достижения поставленных целей. Обычно занимает много времени.

Например, в приложении интернет-магазина можно выбрать разные способы оплаты, варианты доставки, проверка сортировки товаров, поиск по ключевым словам и т. д.

3. **Расширенное** — тестирование, всей заявленной в требованиях функциональности, в том числе той, у которой низкий приоритет и незначительная важность. Например, открытие ссылок в подвале сайта с разных страниц.



Схема для лучшего понимания

Иные

1. **Кроссбраузерное** — вид тестирования, направленный на поддержку и корректное отображение программного продукта в разных браузерах.
2. **Кроссплатформенное** — тестирование ПО при котором проверяется, что тестируемое ПО одинаково корректно работает на разных платформах, под разными операционными системами.

Подводя итог статьи, хочу подчеркнуть, что её создание было вдохновлено желанием собрать и систематизировать информацию о различных видах тестирования из разных источников. Надеюсь, что данная статья окажется полезной для всех, кто занимается изучением и практикой тестирования.

Приглашаю вас поделиться вашими мыслями, вопросами и обратной связью по данной статье в комментариях!

Теги: [тестирование](#), [виды тестирования](#), [классификация](#)

Хабы: [Тестирование IT-систем](#), [Тестирование веб-сервисов](#),
[Тестирование мобильных приложений](#)

 +25

 176



 17



10 22

КармаРейтинг

Mark Stachev @stachevm


Начинающий тестировщик

 Комментарии 17

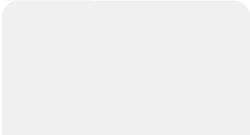
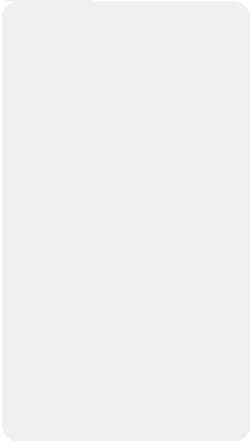
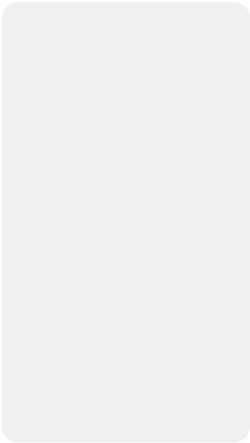
Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ

- 
- 

ИСТОРИИ



РАБОТА

Тестировщик программного обеспечения

55 вакансий

Тестировщик приложений

26 вакансий

Тестировщик мобильных игр

26 вакансий

[Все вакансии](#)

Ваш аккаунт

[Войти](#)

[Регистрация](#)

Разделы

[Статьи](#)

[Новости](#)

[Хабы](#)

[Компании](#)

[Авторы](#)

[Песочница](#)

Информация

[Устройство сайта](#)

[Для авторов](#)

[Для компаний](#)

[Документы](#)

[Соглашение](#)

[Конфиденциальность](#)

Услуги

[Корпоративный блог](#)

[Медийная реклама](#)

[Нативные проекты](#)

[Образовательные
программы](#)

[Стартапам](#)

[Спецпроекты](#)



[Настройка языка](#)

[Техническая поддержка](#)

© 2006–2023, Habr

