
Лекция 3. Виды тестирования ПО
NetCracker®
© 2014 NetCracker Technology Corp. Confidential.

Классификация видов тестирования

Виды тестирования – различные подходы к исполнения тестирования, в зависимости от следующих признаков:

- имеем ли мы доступ к исходному коду программы;
- какие именно требования мы тестируем (какой объект тестирования);
- кто именно тестирует программу;
- на каком этапе жизненного цикла ПО происходит тестирование;
- являются ли тестовые сценарии «позитивными»;
- тестируются ли данные компоненты в изоляции от других;
- тестирование происходит вручную или автоматически;
- откуда берутся ожидаемые результаты

Black box testing (Тестирование чёрного ящика или **поведенческое тестирование) -** В этом методе программа рассматривается как чёрный ящик. Целью тестирования ставится выяснение обстоятельств, в которых поведение программы не соответствует спецификации. Для обнаружения всех ошибок в программе необходимо выполнить исчерпывающее тестирование, то есть тестирование на всевозможных наборах данных. Для большинства программ такое невозможно, поэтому применяют разумное тестирование, при котором тестирование программы ограничивается небольшим подмножеством всевозможных наборов данных. При этом необходимо выбирать наиболее подходящие подмножества, подмножества с наивысшей вероятностью обнаружения ошибок.

White box testing («Белый ящик») — тестирование кода на предмет логики работы программы и корректности её работы с точки зрения компилятора того языка, на котором она писалась.

Техника тестирования по принципу Белого ящика, также называемая техникой тестирования, управляемой логикой программы, позволяет проверить внутреннюю структуру программы. Исходя из этой стратегии тестировщик получает тестовые данные путем анализа логики работы программы.

Техника Белого ящика включает в себя следующие методы тестирования:

- покрытие операторов
- покрытие решений
- покрытие условий
- покрытие решений и условий
- комбинаторное покрытие условий

Grey box testing – тестирование, при котором известна структура объекта, но не известны качественные значения параметров. Т.е., код может быть не доступен, однако известен алгоритм.

В компании Netcracker зачастую выполняется black box тестирование.

Functional testing (функциональное тестирование) — это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает.

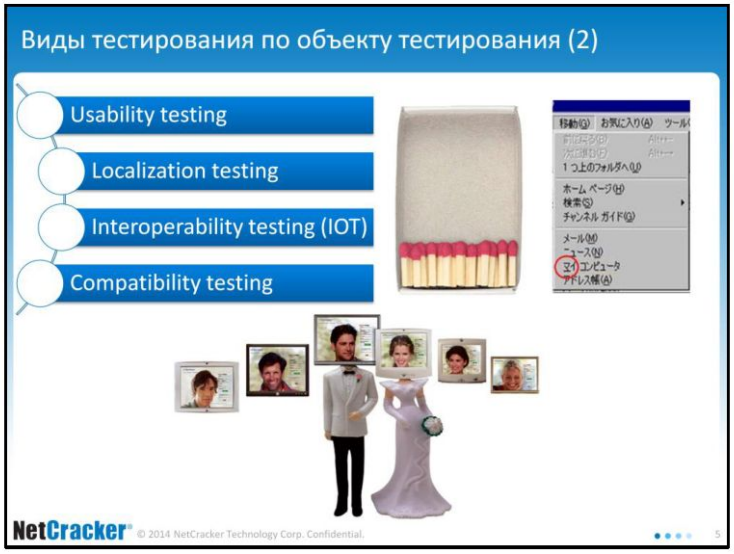
Non-functional testing (Нефункциональное тестирование):

- **Load testing (нагрузочное тестирование)** — определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).

- Для исследования времени отклика системы на высоких или пиковых нагрузках производится **стресс-тестирование**, при котором создаваемая на систему нагрузка превышает нормальные сценарии её использования.

- **UI testing (тестирование интерфейса пользователя)** предполагает проверку соответствия приложения требованиям к графическому интерфейсу, профессионально ли оно выглядит, выполнено ли оно в едином стиле.
  В большинстве случаев, функциональное тестирование приложения осуществляется вместе со следующими видами тестирования графического интерфейса пользователя:
- Тестирование на соответствие стандартам графических интерфейсов;
- Тестирование с различными разрешениями экрана;
- Тестирование в ограниченных условиях, например, в условиях нехватки памяти;
- Совместимость с различными Интернет-браузерами;
- Тестирование локализованных версий: точность перевода, проверка длины

названий элементов интерфейса и т.д.;
- Тестирование графического интерфейса пользователя на целевых устройствах (для КПК-совместимых приложений).



- **Usability testing (юзабилити тестирование, тестирование удобства пользования)** - это метод тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.

Исследование, выполняемое с целью определения, удобен ли некоторый искусственный объект (такой как веб-страница, пользовательский интерфейс или устройство) для его предполагаемого применения. Таким образом, проверка эргономичности измеряет эргономичность объекта или системы. Проверка эргономичности сосредоточена на определённом объекте или небольшом наборе объектов, в то время как исследования взаимодействия человек-компьютер в целом — формулируют универсальные принципы.

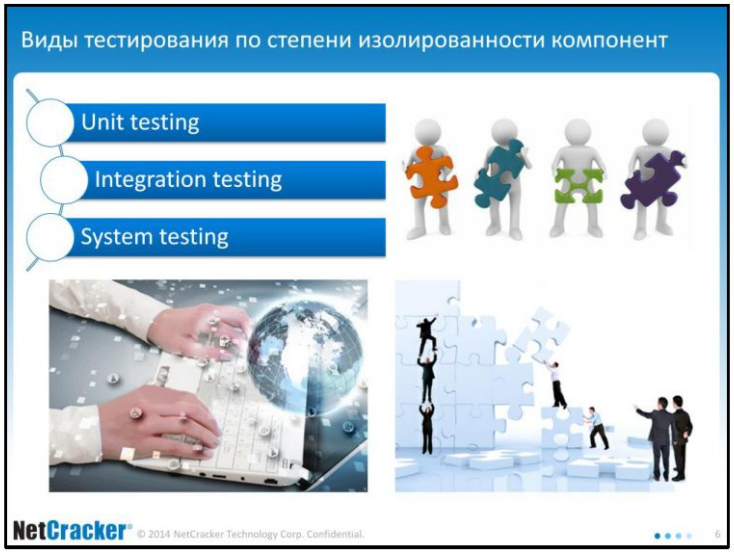
- **Localization testing (тестирование локализации)** – тестирование, которое проводится чаще всего после перевода языка ПО на другой. Под локализацией понимают процесс адаптации ПО к культуре какой-либо страны.

  При таком тестировании выполняется проверка не только правильность перевода интерфейсов , а и:

  корректность работы в операционной системе другой страны;
  - Вывод на экран символов языка;
  - Стандарты страны: формат числа, даты, валюта и т.п.
  - Учет менталитета, традиций;
- **Interoperability testing (IOT) (тестирование взаимодействия с внешними системами)** – это функциональное тестирование, проверяющее способность

приложения взаимодействовать с одним и более компонентами или системами и включающее в себя тестирование совместимости (compatibility testing) и интеграционное тестирование (integration testing).

- **Compatibility testing (тестирование совместимости)** — вид нефункционального тестирования, основной целью которого является проверка корректной работы продукта в определенном окружении. Окружение может включать в себя следующие элементы:
    - Аппаратная платформа;
    - Сетевые устройства;
    - Периферия (принтеры, CD/DVD-приводы, веб-камеры и пр.);
    - Операционная система (Unix, Windows, MacOS, ...)
    - Базы данных (Oracle, MS SQL, MySQL, ...)
    - Системное программное обеспечение (веб-сервер, файрвол, антивирус, ...)
    - Браузеры (Internet Explorer, Firefox, Opera, Chrome, Safari)



Unit testing (модульное тестирование) – тестирование отдельных компонент, часто выполняется разработчиками.
Такое тестирование позволяет проверить на корректность отдельные модули исходного кода программы.
Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.
Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны. Этот тип тестирования обычно выполняется программистами.

Integration testing (интеграционное тестирование) - одна из фаз тестирования программного обеспечения, при которой отдельные программные модули объединяются и тестируются в группе. Обычно интеграционное тестирование проводится после модульного тестирования и предшествует системному тестированию.

Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования.

Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным и требованиям надежности. Тестирование этих проектируемых единиц — объединения, множества или группы модулей — выполняется через их интерфейс, с использованием тестирования «чёрного ящика».

System testing (системное тестирование) - это тестирование программного обеспечения (ПО), выполняемое на полной, интегрированной системе, с целью

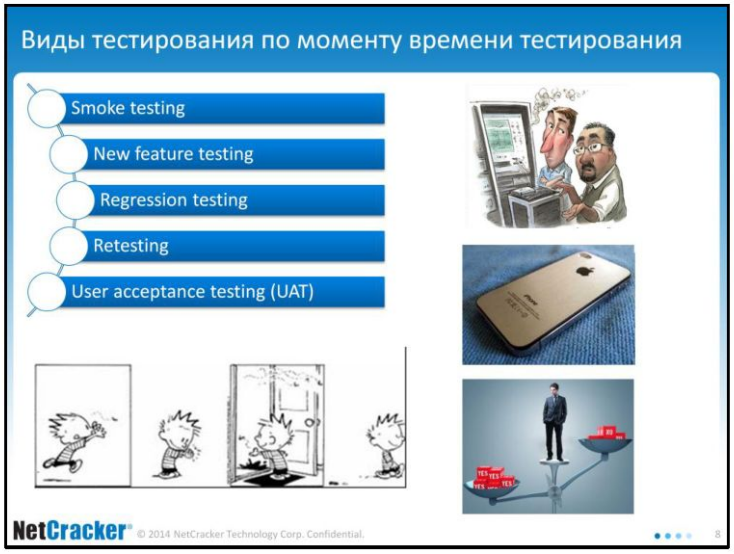
проверки соответствия системы исходным требованиям. Системное тестирование относится к методам тестирования чёрного ящика, и, тем самым, не требует знаний о внутреннем устройстве системы. ***Альфа-тестирование и бета-тестирование*** *являются* ***подкатегориями системного тестирования.***

Alpha testing (Альфа-тестирование) - имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования. Иногда альфа-тестирование выполняется под отладчиком или с использованием окружения, которое помогает быстро выявлять найденные ошибки. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования в окружении, подобном тому, в котором будет использоваться ПО.

Beta testing (Бета-тестирование) - в некоторых случаях выполняется распространение предварительной версии (в случае проприетарного ПО иногда с ограничениями по функциональности или времени работы) для некоторой большей группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей.

Альфа-тестирование (alpha testing):

1. Дымовое тестирование (smoke testing)
2. Тестирование новой функциональности (new feature testing)
3. Перетестирование (retesting)
4. Регрессионное тестирование (regression testing)
5. Приёмочное тестирование (acceptance testing)



Smoke testing (дымовое тестирование) – рассматривается как короткий цикл тестов, выполняемый для подтверждения того, что после сборки кода (нового или исправленного) устанавливаемое приложение, стартует и выполняет основные функции.

Вывод о работоспособности основных функций делается на основании результатов поверхностного тестирования наиболее важных модулей приложения на предмет возможности выполнения требуемых задач и наличия быстронаходимых критических и блокирующих дефектов. В случае отсутствия таковых дефектов дымовое тестирование объявляется пройденным, и приложение передается для проведения полного цикла тестирования, в противном случае, дымовое тестирование объявляется проваленным, и приложение уходит на доработку.

New feature testing (тестирование новой функциональности) – тестирование производится впервые.

Regression testing (регрессионное тестирование) - это вид тестирования направленный на проверку изменений, сделанных в приложении или окружающей среде (починка дефекта, слияние кода, миграция на другую операционную систему, базу данных, веб сервер или сервер приложения), для подтверждения того факта, что существующая ранее функциональность работает как и прежде.

Retesting (перетестирование) – тестирование функциональности после исправления в ней дефекта.

User acceptance testing (UAT) (приемное тестирование или приемо-сдаточное испытание) – тестирование, зачастую происходящее на стороне заказчика. Формальный процесс тестирования, который проверяет соответствие системы требованиям и проводится с целью:
определения удовлетворяет ли система приемочным критериям;
вынесения решения заказчиком или другим уполномоченным лицом принимается приложение или нет.
Фаза приемочного тестирования длится до тех пор, пока заказчик не выносит решение об отправлении приложения на доработку или выдаче приложения.

Positive testing ("Позитивное" тестирование) – это тестирование на данных или сценариях, которые соответствуют нормальному (штатному, ожидаемому) поведению системы.
Основной целью "позитивного" тестирования является проверка того, что при помощи системы можно делать то, для чего она создавалась.
Negative testing ("Негативное" тестирование) – это тестирование на данных или сценариях, которые соответствуют нештатному поведению тестируемой системы – различные сообщения об ошибках, исключительные ситуации, "запредельные" состояния и т.п.
Основной целью "негативного" тестирования является проверка устойчивости системы к воздействиям различного рода, валидация неверного набора данных, проверка обработки исключительных ситуаций (как в реализации самих программных алгоритмов, так и в логике бизнес-правил).
Boundary testing (граничное тестирование)

Manual testing – ручное тестирование
Automated testing – автоматизированное тестирование
Semi-automated testing – полуавтоматизированное тестирование

Formal testing (тестирование по документации) – тестирование по документации, по сценарию.

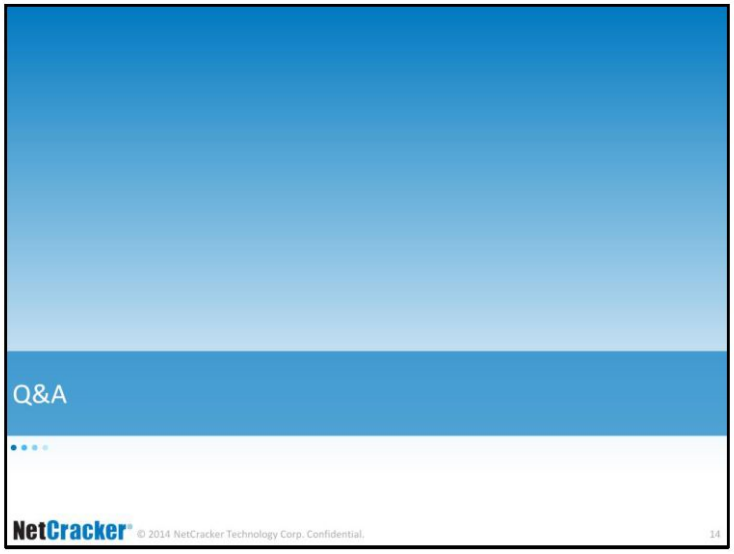
Ad hoc testing (тестирование ad hoc или интуитивное тестирование) – при полном отсутствии документации, тестирование «методом тыка», однако используя собственный опыт и здравый смысл.

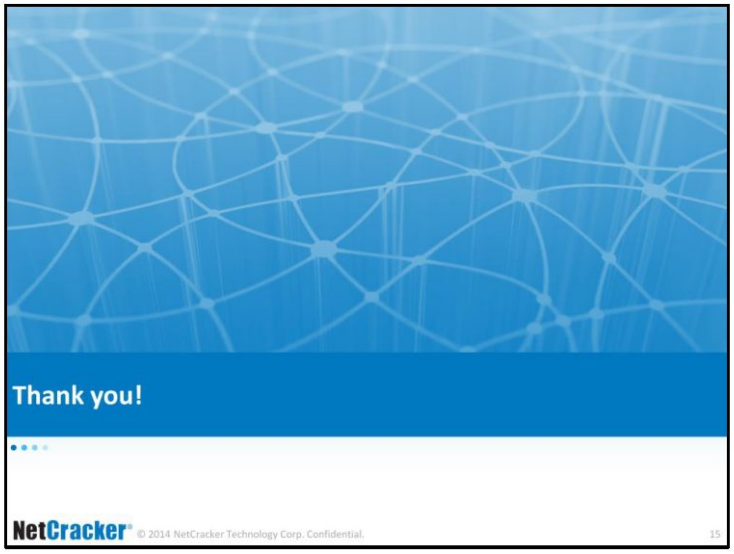
Exploratory testing (исследовательское тестирование) – тестирование при полном отсутствии документации, методом свободного поиска, наобум. Эффективно работает в условиях неопределенности.


Виды тестирования (summary)
По объекту тестирования
• Functional testing
• Non-functional testing
• Load, stress testing
• Usability testing
• UI testing
• Security testing
По времени проведения тестирования
• Alpha testing
• Smoke testing
• New feature testing
• Regression testing
• Acceptance testing
• Beta testing
По степени изолированности компонентов
• Component / Unit testing
• Integration testing
• System / end-to-end testing
По знанию системы
• Black-box testing
• White-box testing
• Grey-box testing
По степени автоматизации
• Manual testing
• Automated testing
• Semi-automated testing
По степени подготовленности к тестированию
• Formal testing
• Ad hoc testing
• Exploratory testing
По признаку позитивности сценариев
• Positive testing
• Negative testing
NetCracker
© 2014 NetCracker Technology Corp. Confidential.
12

Материалы для самостоятельного изучения

- Роман Савин. Тестирование dot com (с.10-35):
  http://adm-lib.ru/books/4/testirovanie_dot-com.pdf
- http://ru.wikipedia.org/wiki/Тестирование_программного_обеспечения
- http://google.com ☺


Q&A
NetCracker®
© 2014 NetCracker Technology Corp. Confidential.
14


Thank you!
NetCracker®
© 2014 NetCracker Technology Corp. Confidential.
15