IT-Academy

Основы тестирования ПО

Виды и методы тестирования.

В этом разделе:

В этом разделе:

- Уровни тестирования;
- Виды и направления тестирования;
- Методы тестирования;
- Базовая терминология.

Уровни тестирования (по степени детализации приложения).

Компонентное тестирование

Компонентное тестирование (component testing, unit testing) – тестирование отдельного модуля.

Интеграционное тестирование (integration testing) – проверка взаимодействия модулей.

Системное тестирование (system testing) – полная проверка приложения.

Компонентное → **Интеграционное** → **Системное**

Компонентное тестирование

Компонентное тестирование (Component Testing/Unit testing/ Module testing): “Тестирование отдельных компонентов программного обеспечения”

Компонент (Component): “Наименьший элемент программного обеспечения, который может быть протестирован отдельно”

Компонентное тестирование

Компоненты для тестирования:

- **Отдельный метод**/функция **в коде** программы;
- **Отдельный компонент в программе** (модуль логина, аккаунт пользователя);
- **Отдельная программа** из сложной интеграционной системы.

Интеграционное тестирование

Интеграционное тестирование (Integration Testing) - тестирование части системы, состоящей из двух и более частей.

Основная задача – поиск дефектов, связанных с ошибками в реализации и интерпретации взаимодействия между модулями.

Так же, как и модульное тестирование, оперирует интерфейсами модулей и подсистем и требует создания тестового окружения.

ВикипедиЯ
Свободная энциклопедия

IT-Academy

Немного терминологии

Интерфе́йс (англ. *interface)* — совокупность возможностей, способов и методов взаимодействия двух информационных систем, устройств или программ.

Интерфейс программирования приложений (API) — набор методов, которые можно использовать для доступа к функциональности другой программы.

Интерфейс командной строки (CLI): инструкции компьютеру даются путём ввода с клавиатуры текстовых строк (команд).

Графический интерфейс пользователя (GUI): программные функции представляются графическими элементами экрана.

Интеграционное тестирование

Заглушки (stab) и **драйверы** (driver):

- Используются для эмуляции недостающих компонентов:
  - Внешние компоненты/системы (регистрация из соц. сетей без подключения к ним);
  - Подсистемы/неготовые модули (регистрация без БД);

- Может понадобиться специально написать их для тестируемой системы (**SUT**).

Интеграционное тестирование

Тестирование интеграции компонентов (component integration testing): тестирование, взаимодействия между несколькими интегрированными компонентами **одного приложения**.

Системное интеграционное тестирование (system integration testing): тестирование, взаимодействия между всеми компонентами системы, между различными приложениями объединенными в систему, **интерфейсами** связи с внешними системами (интернет и т.д.).

Системное тестирование

Системное тестирование (System Testing):

“Процесс тестирования системы в целом с целью проверки того, что она соответствует установленным требованиям”.

Тестирование полной системы:

- Может быть последним шагом в интеграционном тестировании в узком смысле;
- Может быть первый раз, когда из компонентов появляется рабочая система.

В идеале – проводится независимой тестовой командой.

Приемочное тестирование

Приёмочное тестирование (acceptance testing): тестирование по отношению к потребностям и требованиям пользователя, проводимое с целью дать возможность пользователям, заказчикам определить, принимать систему или нет.

Как правило заключительный этап тестирования, осуществляемый перед передачей продукта заказчику и/или конечным пользователям.

Приемочное тестирование

Типичные **формы** приемочного тестирования:

- Пользовательское **приемочное** тестирование (UAT);
- Эксплуатационное **приемочное** тестирование (OAT);
- Альфа- и бета- тестирование.

Формы приемочного тестирования.

Пользовательское приемочное тестирование
(User acceptance testing):

- Тестирование конечного продукта **проводят пользователи**;
- Может проходить как на оборудовании производителя или пользователей.

Эксплуатационное приемочное тестирование
(Operational acceptance testing):

- Тестирование резервного копирования/восстановление;
- Аварийное восстановление;
- Задачи **технической поддержки**;
- Периодические проверки уязвимостей безопасности.

Формы приемочного тестирования.

Альфа-тестирование (alpha testing): тестирование потенциальными пользователями/заказчиками или независимой командой тестирования **на стороне разработчиков**, но вне разрабатывающей организации.

Бета-тестирование (beta testing): тестирование потенциальными и/или существующими клиентами/заказчиками **на внешней стороне никак не связанными с разработчиками**.

Это форма внешнего приёмочного тестирования готового программного обеспечения для того чтобы получить отзывы рынка

Уровни тестирования

IT-Academy

Резюме

- Существует много разных подходов к **компонентному** тестированию.
- Интеграционное тестирование проверяет, как компоненты и/или системы **взаимодействуют** друг с другом.
- Системное тестирование позволяет проверить **систему в целом** согласно спецификаций, требований и бизнес процессов.
- Участие **пользователя** в приемочном тестировании является критически важным.

IT-Academy

Есть вопросы? Давайте обсудим!

Исторически так сложилось,
что как минимум «тип тестирования»
(testing type) и «вид тестирования»
(testing kind) давно стали
синонимами.

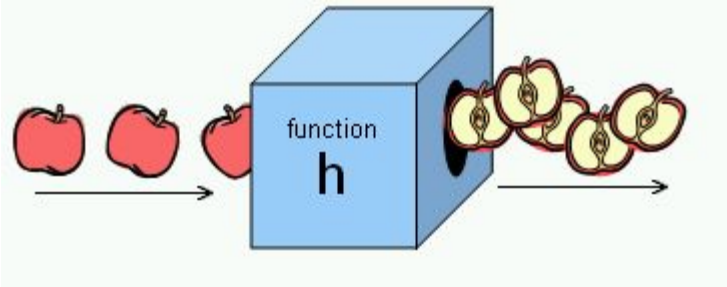
С. Куликов.

Виды/направления тестирования по целям и задачам.

Виды тестирования

Функциональное тестирование (functional testing): тестирование, основанное на анализе функциональности компонента или системы.

- Функции – это то, **ЧТО ДЕЛАЕТ** система;
- Функциональное тестирование **может проводиться на всех уровнях тестирования.**



Виды тестирования

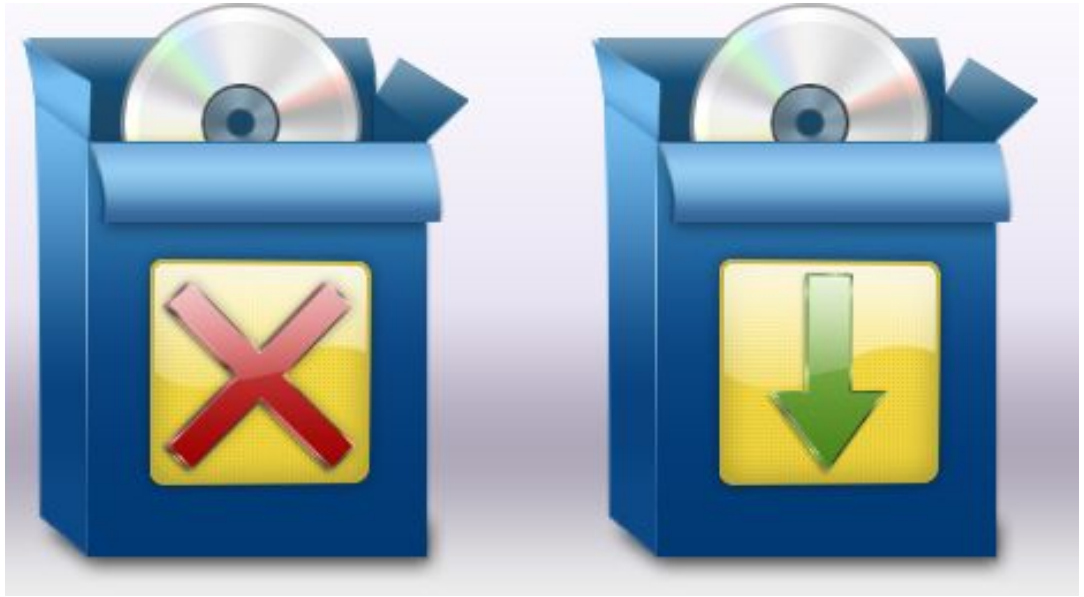
Нефункциональное тестирование (non-functional testing): Тестирование атрибутов компонента или системы, не относящихся к функциональности, то есть:

- надежность,
- внешний вид,
- практичность,
- устойчивость..

Проверяет, **КАК** система **РАБОТАЕТ**.

Может быть представлено на всех уровнях тестирования.

Виды тестирования



Инсталляционное тестирование (installation testing) – «установка и удаление»

Виды тестирования

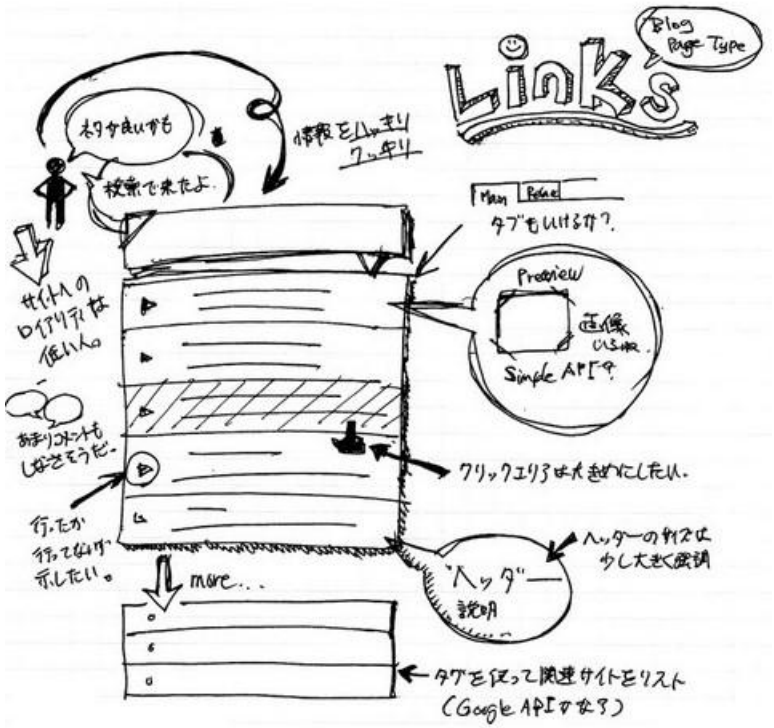
Конфигурационное тестирование (configuration testing) – «разное оборудование и настройки»

Виды тестирования

Тестирование совместимости
(compatibility testing) –
«разный софт вокруг»

Виды тестирования

Тестирование графического интерфейса (GUI testing) – «как расположены кнопочки».



Виды тестирования

Тестирование удобства использования (usability testing) – «не заставляй меня думать»

Суровая правда жизни (C) bash.org.ru
«Поражаюсь старательности подрастающего поколения... Сегодня по дороге на работу заметил появление большого количества скворечников на деревьях. Дабы не долбить деревья гвоздями, скворечники примотаны скотчем. Многие прямо поперек кругленького отверстия...»

Виды тестирования

Тестирование удобства

проводится с целью определения, **удобна ли программа для** ее предполагаемого применения и основанное на привлечении **группы пользователей** в качестве тестировщиков и суммировании и анализе полученных от них выводов.

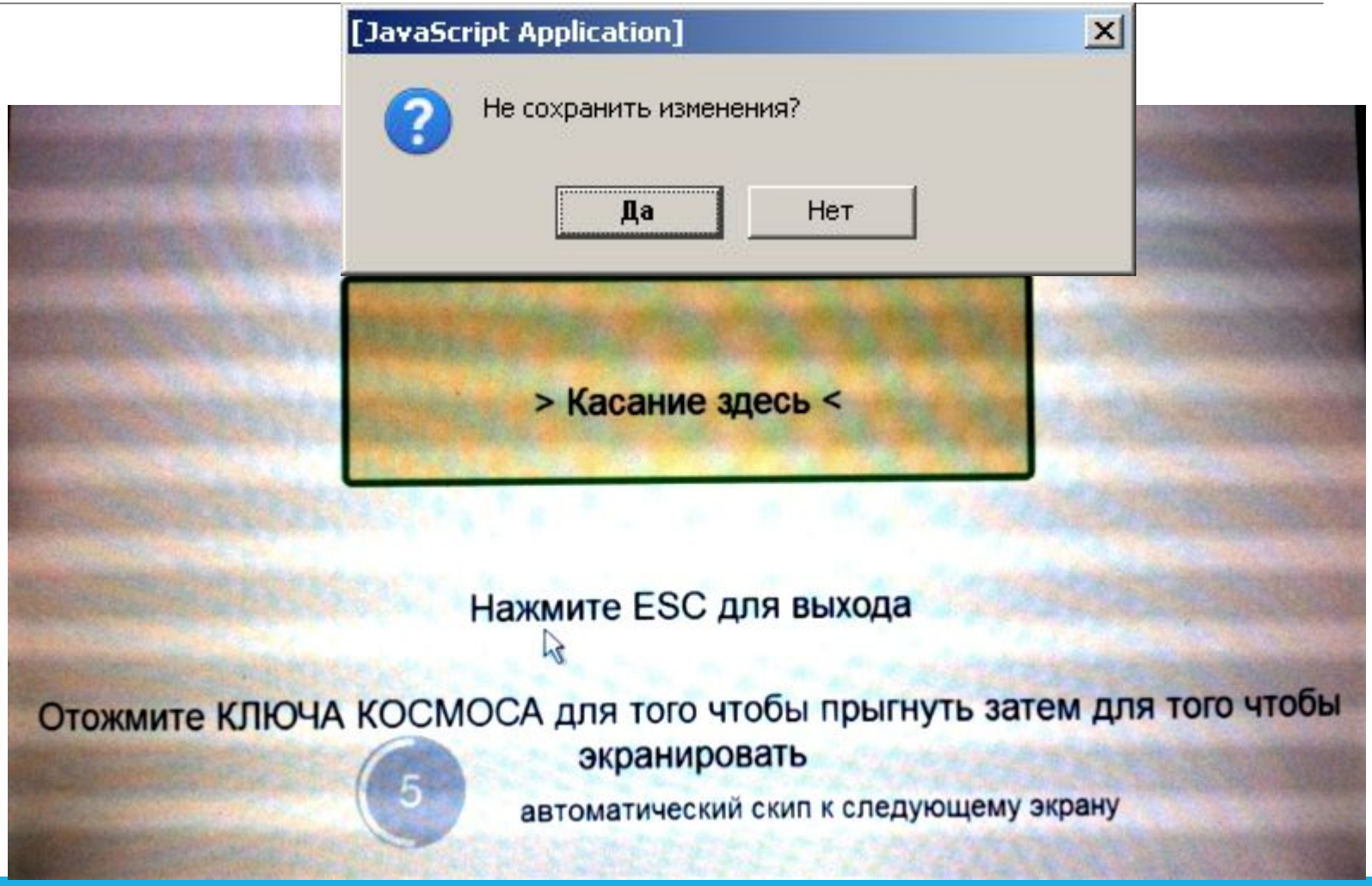
- Бета-тестирование может быть дешевым способом провести тестирование практичности.

Виды тестирования

Тестирование интернационализации (internationalization testing, i18n) – «а если перевести на японский?»

Виды тестирования

Тестирование локализации (localization testing, l10n) – «как перевели?»



Виды тестирования

Тестирование безопасности

(security, penetration testing) – «а если ломать?»

Виды тестирования

Тестирование безопасности (security testing): Тестирование с целью оценить защищенность программного продукта.

- Насколько просто неавторизованному пользователю получить доступ к системе?
- Насколько просто постороннему лицу получить доступ к данным

Виды тестирования

Тестирование доступности (accessibility testing) – «а если я плохо вижу».

Виды тестирования

Тестирование производительности (performance test): тест, проводимый с целью оценить поведение системы под нагрузкой.

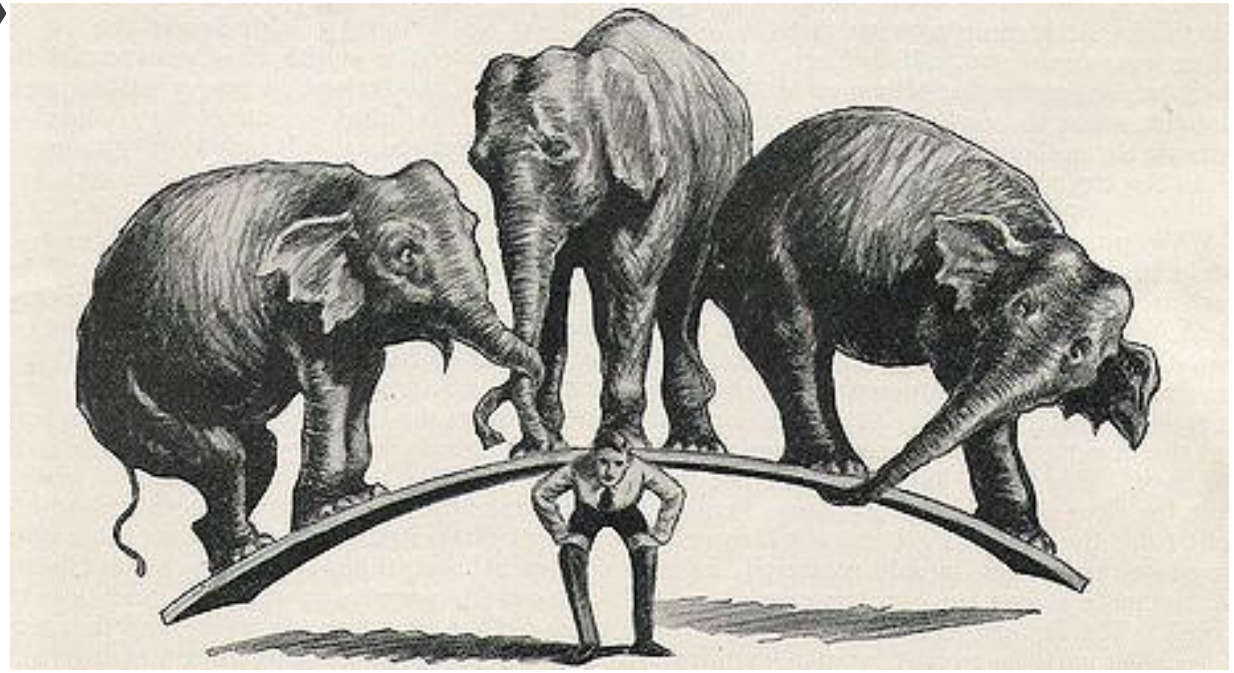
Направления тестирования производительности

Нагрузочное тестирование (load test): «а если много пользователей?».

Направления тестирования производительности

IT-Academy

Стресс тестирование (stress test): «а если гораздо больше, чем может выдержать?»



IT-Academy

По «тестированию изменений».

Виды тестирования

Тестирование нового функционала (new feature testing) – «всё обещанное на месте и работает»

«Крайне важно успеть скомпилировать библиотеку до того момента, как придёт новая отличная идея.»
(C) bash.org.ru

Виды тестирования

Повторное тестирование (confirmation testing, re-testing, bug-fix verification) – *«что было исправлено, то стало работать»*

Виды тестирования

Регрессионное тестирование (regression testing) – «нет ухудшений», *т.е. «что работало, то продолжает работать»*

Регрессионное тестирование

Регрессионное тестирование:

- **Объем** регрессионного тестирования **основан на риске** обнаружения дефектов в ранее работавшем ПО;
- Может проводиться **на всех уровнях** тестирования;
- Применимо к **функциональному и нефункциональному** тестированию;
- Регрессионные тесты должны запускаться и **при изменениях в ПО, и при изменениях в окружении**;

Регрессионное тестирование – очень важная техника тестирования!

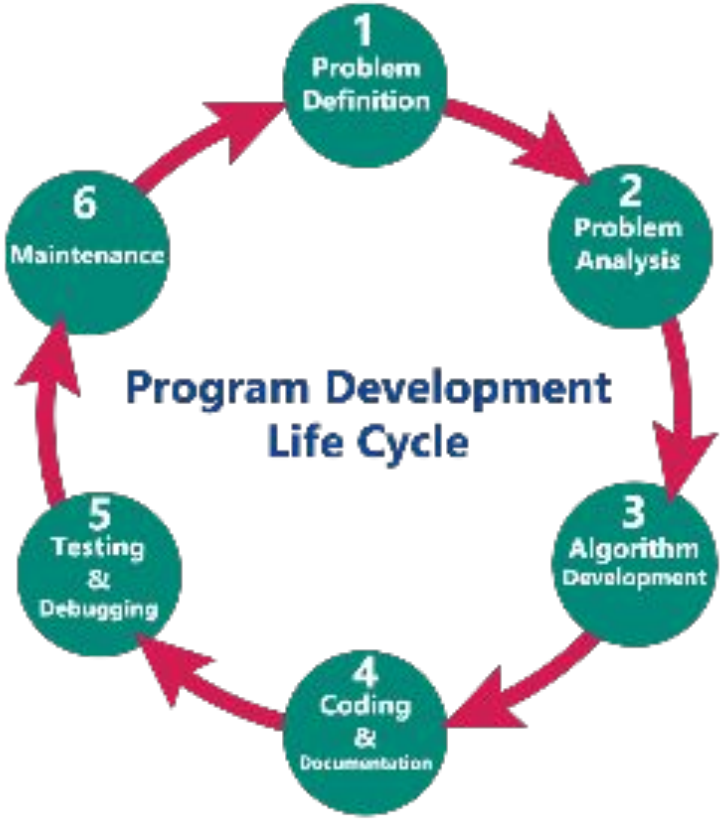
Регрессионное тестирование

Резюме:

- **Исправлена** ошибка – программа ДОЛЖНА быть **протестировано повторно**.
- **Исправления** старых – могут **появляться** новые.
- **Обещали справить – не факт**, что исправили.
- Тесты должны разрабатываться с целью их повторного использования.
- **Повторное тестирование** – это перезапуск «провалившихся» тестов после исправления ошибок, чтобы убедиться, что исправленное работает.
- **Регрессионное тестирование** - это запуск более широкого набора тестов, чтобы проверить, не появились ли неожиданные ошибки в коде, который не меняли.

IT-Academy

Типы тестирования в зависимости от фазы разработки.



В зависимости от фазы разработки

Тестирование разработки (development testing): тестирование, проводимое во время разработки системы, обычно в рабочей среде разработчиков. ***Основная цель*** *– вызвать максимально возможное количество отказов, чтобы они были* ***обнаружены и исправлены***;

Приёмочное тестирование (acceptance testing): тестирование по отношению к потребностям пользователя или заказчика. ***Основная цель*** *– подтвердить, что система работает, как ожидается, чтобы* ***решить принимать или нет***;

В зависимости от фазы разработки

Тестирование в период сопровождения (maintenance testing): тестирование изменений в действующей системе или влияния изменений в окружении на действующую систему. ***Основная цель*** *– проверить, что при разработке изменений* ***не было произведено новых дефектов***;

Тестирование в период сопровождения

Главные мотивы:

- **Изменение** программных продуктов;

  Включает запланированные улучшения, поправки и аварийные изменения, изменения окружения (обновление БД, патчи для уязвимостей ОС).

- **Миграция** программных продуктов (из одной программы в другую);

  Следует включить эксплуатационное тестирование в новых условиях.

- **Вывод** из эксплуатации ПО или систем;

  Может включать тестирование миграции данных или архивирования (если требуется длительное хранение данных).

Тестирование в период сопровождения

Особенности:

- Изменения в приложении - как правило незначительные;
- Система должна быть протестирована **быстро и эффективно**, так как:
  - Ранняя остановка тестирования – ошибки могут быть пропущены;
  - Затянутое тестирование – упущенные возможности бизнеса;
- Эти системы могут быть установлены и работать годами:
  - Жизненно важные для бизнеса;
  - Могут быть в использовании **24/7**.

Тестирование в период сопровождения

Что делать:

- **Регрессионное тестирование для частей системы**, которые подверглись изменениям;
- Объем тестов связан с риском изменения, размером существующей системы и размером изменения.

Тестирование в период сопровождения

Основные сложности для тестировщика:

- **Отсутствие документации**/требований к приложению;
- **Старая/не** актуальная документация.

Выход:

- уже **имеющаяся** система (для эталонного тестирования),
- **руководство** пользователя,
- профессиональные знания **специалиста**.

IT-Academy

Есть вопросы? Давайте обсудим!

IT-Academy

По степени важности тестируемых функций и приоритету выполнения тестов.

По степени важности и приоритету выполнения

Смоук тест (smoke test) – «только самое важное».

Тест критического пути (critical path test) – «повседневное».

Расширенный тест (extended test) – «всё остальное».

Внимание! Возможна путаница!

Единой классификации не существует, и две катигории имеют в обиходе профессионалов похожие названия:

Уровни тестирования (по степени детализации приложения)
= компонентное, интеграционное, системное.

Уровни функционального тестирования (по важности функций)
= smoke, critical path, extended.

IT-Academy

По принципу работы с приложением.

IT-Academy

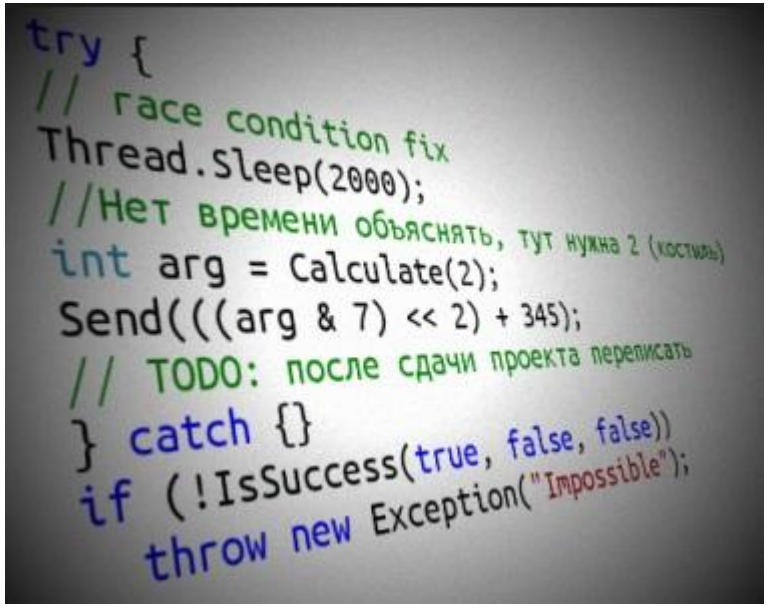
По принципу работы с приложением

Позитивные тесты (positive test) – все «строго по инструкции».

Негативные тесты (negative test) – делаем так, как нельзя.

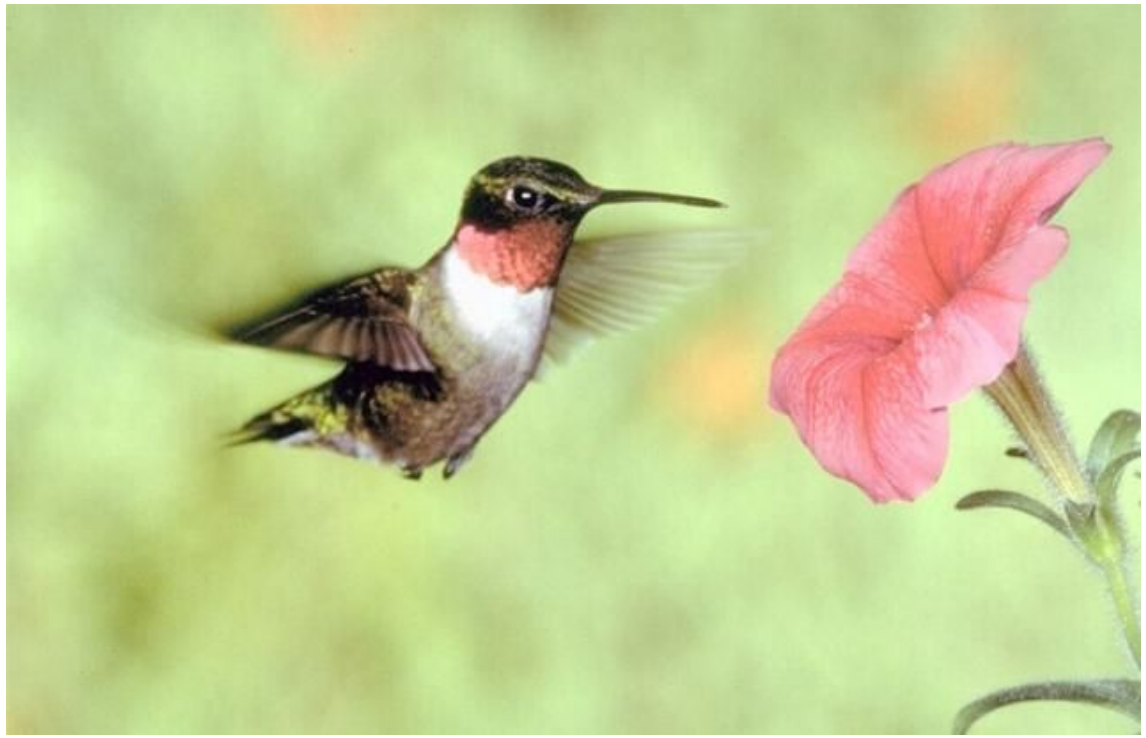
IT-Academy

По запуску кода на исполнение.



Статическое тестирование

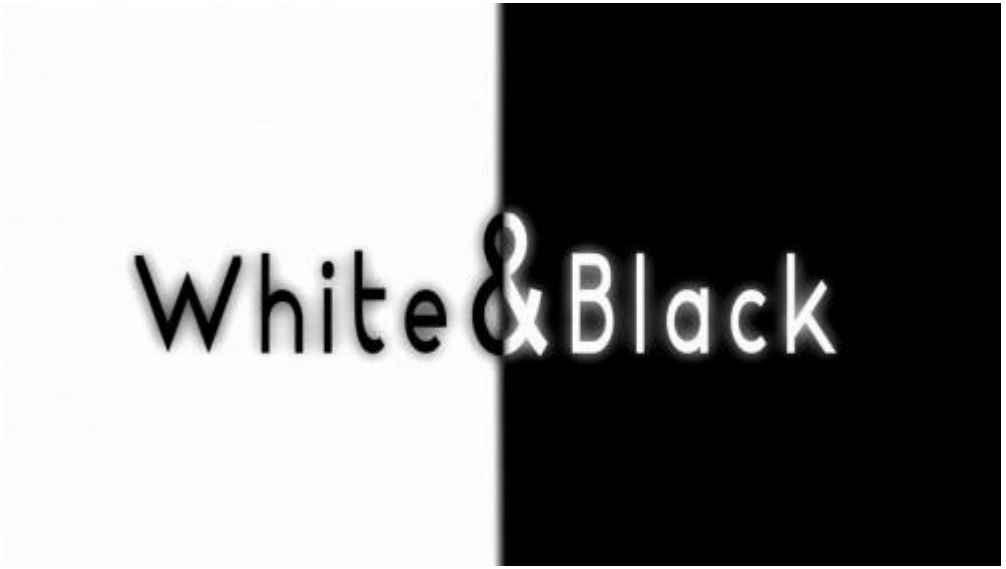
(static testing) – без запуска программы



Динамическое тестирование (dynamic testing) – с запуском программы

IT-Academy

По «доступу к коду».



Метод белого ящика

(white-box testing, glass-box testing)

«тестировщик опирается на код».

Метод чёрного ящика

Тестирование методом черного ящика (black-box testing, specification based, behavioral based): без знания внутренней структуры компонента или системы.

- тестировщик опирается на требования и интерфейс приложения
- «работаем так, как работает конечный пользователь»

Метод серого ящика
(gray box testing)

IT-Academy

совокупность подходов из методов белого и чёрного ящика

Какие плюсы и минусы есть у каждого из этих методов тестирования?

IT-Academy

IT-Academy

По «уровню формализации».

По «уровню формализации»

На основе тест-кейсов (scripted testing, test case based testing) – тестирование производится на основе заранее подготовленных тест-кейсов.

На основе чек-листов (checklist-based testing) – тестирование производится на основе чек-листов.

Исследовательское тестирование (exploratory testing) – по тест-кейсу/сценарию, который дорабатывается в процессе выполнения самих тестов.

Свободное (интуитивное) тестирование (ad hoc testing) – ни тест-кейсов, ни чек-листов, ни сценариев, полностью опираемся на свой профессионализм и интуицию.

IT-Academy

Есть вопросы? Давайте обсудим!

Задание.

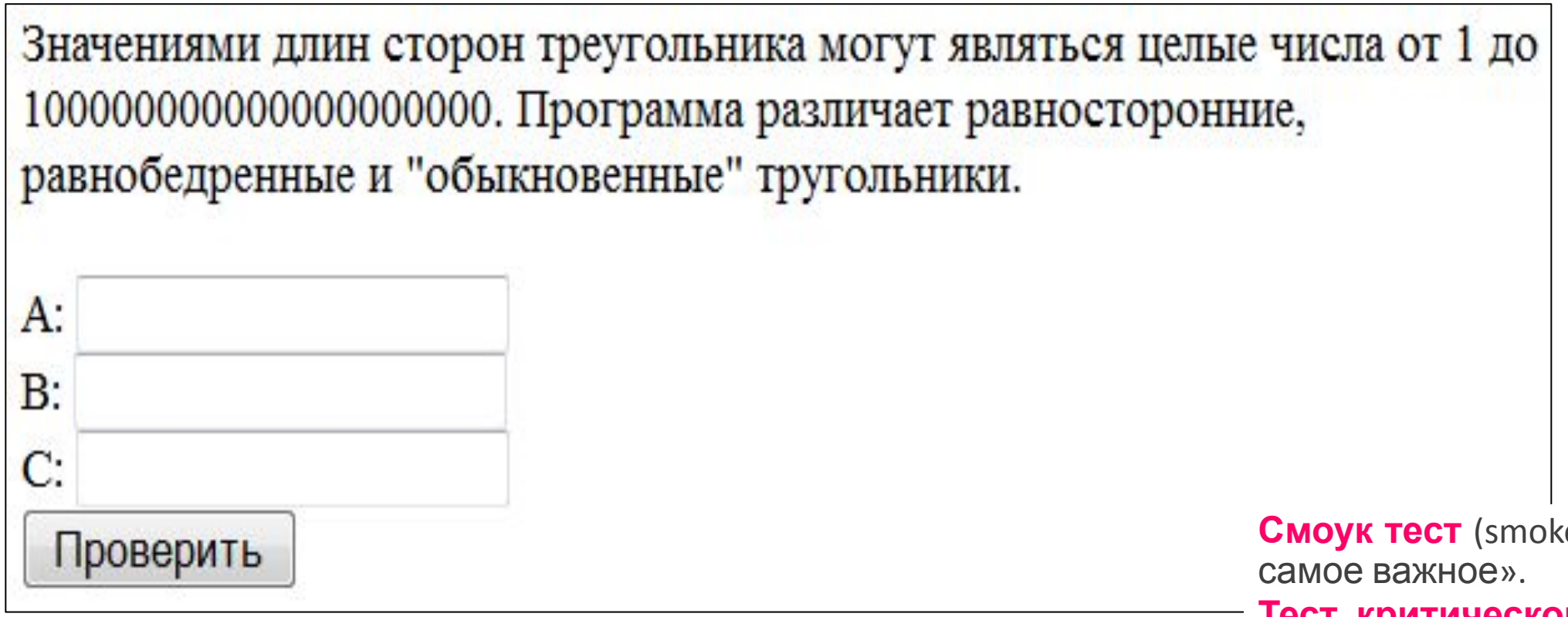
Итак, мы разрабатываем ПО для мобильного телефона. Приведите по 3-5 примеров тестов для **смоук и экстендед** тестов.

Смоук тест (smoke test) – «только самое важное».

Тест критического пути (critical path test) – «повседневное».

Расширенный тест (extended test) – «всё остальное».

Задание.



Смоук тест (smoke test) – «только самое важное».

Тест критического пути (critical path test) – «повседневное».

Расширенный тест (extended test) – «всё остальное».

Задание.

And now, requirements for new feature on your web site. And again, smoke and extended level tests.

IT-Academy

User Story #1

Summary: As an end user, I want to send files via chat.

Description:

- There is possibility to send *.jpg and *.doc format;
- User can click on “Browse” button and select file in file system;
- User can drag-and-drop file into chat field;
- In case of slow internet connection (< 1 Mbps), sending is canceled, user is notified with “File can not be sent due to slow Internet. Please, try later”.

IT-Academy

Есть вопросы? Давайте обсудим!

IT-Academy

Небольшое задание на дом:

Представьте ситуацию: вы хотите у меня уточнить, как правильно оформить домашку на завтра. У вас есть любые известные вам средства коммуникации для связи со мной.

Задание: подумать, где и как вы будете у меня уточнять и написать текст того, как вы это сделаете.