Тестирование ПО
лек. 3

А. Задорожный

2018

Контрольные вопросы

1. Какие подходы к подготовке тестовых данных рассмотрены к настоящему моменту?
2. Что такое “парное тестирование”? В чем заключаются его достоинства?
3. Зачем оценивать качество тестирования?
4. Как измеряется покрытие кода?
5. Как измеряется покрытие требований?
6. Почему важно тестирование с самых ранних этапов разработки?

Содержание

1. Классификация тестирования
   a) По видам
   b) По глубине
   c) …
2. Автоматизация тестирования
   a) Определение
   b) Варианты автоматизации
   c) Достоинства и недостатки
   d) Инструменты автоматизации

Классификация тестирования

Тестирование ПО имеет множество аспектов - важна классификация.

Таких классификаций несколько.

Важнейшая – по видам тестирования

Классификация тестирования

Виды тестирования:

- Функциональное
- Нефункциональное
  - *Безопасности*
  - *Производительности*
  - *Usability, Accessibility*
  - *Совместимости*
  - *…*

Классификация тестирования
Функциональное тестирование

Функциональное тестирование – проверка соответствия ПО функциональным требованиям.

Более общее определение:

Функциональное тестирование – проверка того, что ПО решает задачи для которых оно предназначалось.

Один из важнейших видов тестирования.

Функциональное тестирование

Упомянутая ранее система мониторинга транспортных средств должна:

- *Отображать ТС на карте;*
- *Отображать актуальность информации о ТС;*
- *Позволять строить отчеты по телеметрии ТС.*

Проверка выполнения каждого этого требования является функциональным тестом.

Функциональное тестирование

Процесс состоит из 4 видов деятельности:

1. Подготовка тестовых данных;
2. Определение ожидаемого результата;
3. Выполнение теста с подготовленными данными;
4. Оценка на основе сравнения фактического и ожидаемого результата;

Функциональное тестирование

Концентрируется на следующих аспектах:

1. Основных функциях тестируемого ПО;

2. Основных вопросах *usability*: навигация, доступность для людей с ограниченными возможностями;

3. Поведение при ошибках (*ошибочных данных или сбоях в системе*) - *содержание сообщений об ошибках, возможность выйти из ошибочной ситуации, отсутствие побочных эффектов ошибок*

Функциональное тестирование

Делятся на *Позитивные* и *Негативные* тесты.

Позитивные тесты основываются на правильных входных данных и проверяют правильность исполнения функции ПО.

Негативные тесты основываются на некорректных входных данных или несоответствующих условиях выполнения и проверяют поведение ПО в таких условиях.

Функциональное тестирование

Возвращаясь к примеру с мониторинговой системой.

Позитивный тест - проверка того, что при задержке телеметрических данных ПО отражает, что данные “*устарели*”.

Негативный тест – проверка того, что если время телеметрических данных оказывается впереди текущего времени.

Функциональное тестирование

Модульное (unit) тестирование – один из видов функционального тестирования.

Интеграционное и системное тестирование – совместное тестирование модулей, также, как правило, функциональное тестирование.

Функциональное тестирование во многих случаях занимает основное время (ресурсы) тестирования.

Важна автоматизация, применительно к функциональному тестированию.

Контрольные вопросы

1. Что такое “функциональное тестирование”?
2. Приведите примеры не функционального тестирования.
3. Перечислите действия, выполняемые при подготовке и проведении функционального тестирования.
4. Каким вопросам уделяется внимание при функциональном тестировании?
5. Что проверяется при негативном тестировании?

Автоматизированное тестирование

Автоматизированное тестирование – *запуск теста, инициализация, выполнение, анализ* и *выдача результата* выполняются (все или частично) при помощи программных инструментов.

Для успешной автоматизации тестирования нужны:

- Дополнительные ресурсы;
- Архитектурные решения; (компонентная архитектура ПО, а иногда некоторые специализированные решения)
- Средства автоматизации;

Автоматизированное тестирование

Модульные тесты – элемент автоматизированного тестирования.

Тестирование приложений без UI – как правило автоматизируется.

Нагрузочное и стресс тестирование – как правило автоматизируется.

Здесь фокус на *функциональном тестировании* приложений с UI.

Автоматизация тестирования

Разработка автоматизированных тестов, в отличие от модульных тестов, ложится на плечи тестировщиков.

Все инструменты предполагают написание программного кода (*test scripts*) на специализированном языке с простым синтаксисом.

Требует изучения средств автоматизации и основ программирования.

Важна “логистика” автоматизированных тестов:

- Где хранятся;
- Инструкции по выполнению;
- Связь с ПО (что тестируют);
- Поддержка актуальности (меняются при изменении UI или устаревают);

Достаточно высока “*стоимость разработки и стоимость владения*” ☺.

Инструменты автоматизированного тестирования

Существует ряд платных сред для разработки автоматизированных тестов:

- **HP QuickTest Professional** (QTP) – одна из наиболее полных сред;
- **Test Complete** – windows, web, Android, iOS;
- **IBM Rational** (*Functional, Performance, TestStudio*) – линейка продуктов, в том числе для автоматизации тестирования.

Обладают широкими возможностями, документацией, обновления.

Далее рассмотрены 2 бесплатных инструмента.

Инструменты автоматизированного тестирования

Тестирование *web*-приложений.

Лидером популярности являются продукты Selenium.
http://selenium2.ru/

Selenium IDE – простенькие тесты (Fire Fox)

Selenium WebDriver – создание сложных тестов для любых браузеров, в том числе мобильных.

Selenium Server – запуск удаленных сценариев с нескольких машин

Инструменты автоматизированного тестирования

Selenium IDE – простенькие тесты (Fire Fox)

- Сценарии записываются по действиям пользователей.
- В записанные сценарии можно вносить коррективы.
- Сценарии можно экспортировать в программы на нескольких языках.

Инструменты автоматизированного тестирования

Selenium IDE будет рассмотрен на семинарах. Как и в *unit*-тестах есть средства проверки равенства величин. В целом синтаксис скрипта тривиален.

Расширяется использование Java Script.

Много интернет ресурсов. Например, http://www.guru99.com/first-selenium-test-script.html

Существует несколько расширений. Например, Flow Control позволяет создавать циклы.

Инструменты автоматизированного тестирования

Selenium WebDriver – создание сложных тестов для любых браузеров, в том числе мобильных.

Набор библиотек, позволяющих создавать программы на некоторых языках для наиболее популярных браузеров.

Java, C#, python, ruby, php, perl.

Selenium WebDriver пример

```
using System;
using OpenQA.Selenium.IE;
using System.Drawing.Imaging;

namespace TestUIExercises
{
    class Program
    {
        static void Main(string[] args)
        {
        ...
        }
    }
}
```

Selenium WebDriver пример

```
using (var drv = new InternetExplorerDriver()) {
        drv.Navigate().GoToUrl("http://test.lockey.ru/LockeyTech");

        var userNameField = drv.FindElementById("UserName");
        var userPasswordField = drv.FindElementById("Password");
        var loginButton =
drv.FindElementByXPath("//input[@value='Вход']");

        userPasswordField.Clear();
    userNameField.SendKeys("az_tech");
        userPasswordField.Clear();
        userPasswordField.SendKeys("123456");
        loginButton.Click();

    …
}
```

Selenium WebDriver пример

```
 try
{
      // Здесь можно делать произвольные проверки
            string res = drv.FindElementByXPath("//p").Text;
            Console.WriteLine(res);
}
catch (Exception ex)
{
      drv.GetScreenshot().SaveAsFile
          ("screen.png", ImageFormat.Png);
          Console.WriteLine
          ("Текст не найден. Скриншот в screen.png");
          Console.WriteLine(ex);
}
```

Примеры XPath

Поиск внутри любого элемента xml.

./tr или **tr** – все элементы tr текущего контекста

/body – элемент body документа

//p – все элементы p документа

/body/table – все элементы table дочерние для body

***[@class]** – все элементы, имеющие атрибут class

//input[@type = “hidden”] – все input типа hidden

Инструменты автоматизированного тестирования

Тестирование настольных приложений

Среди бесплатных лидер AutoIt.

http://www.autoitscript.com/site/autoit

Изначально создавался как инструмент массового развертывания приложений.

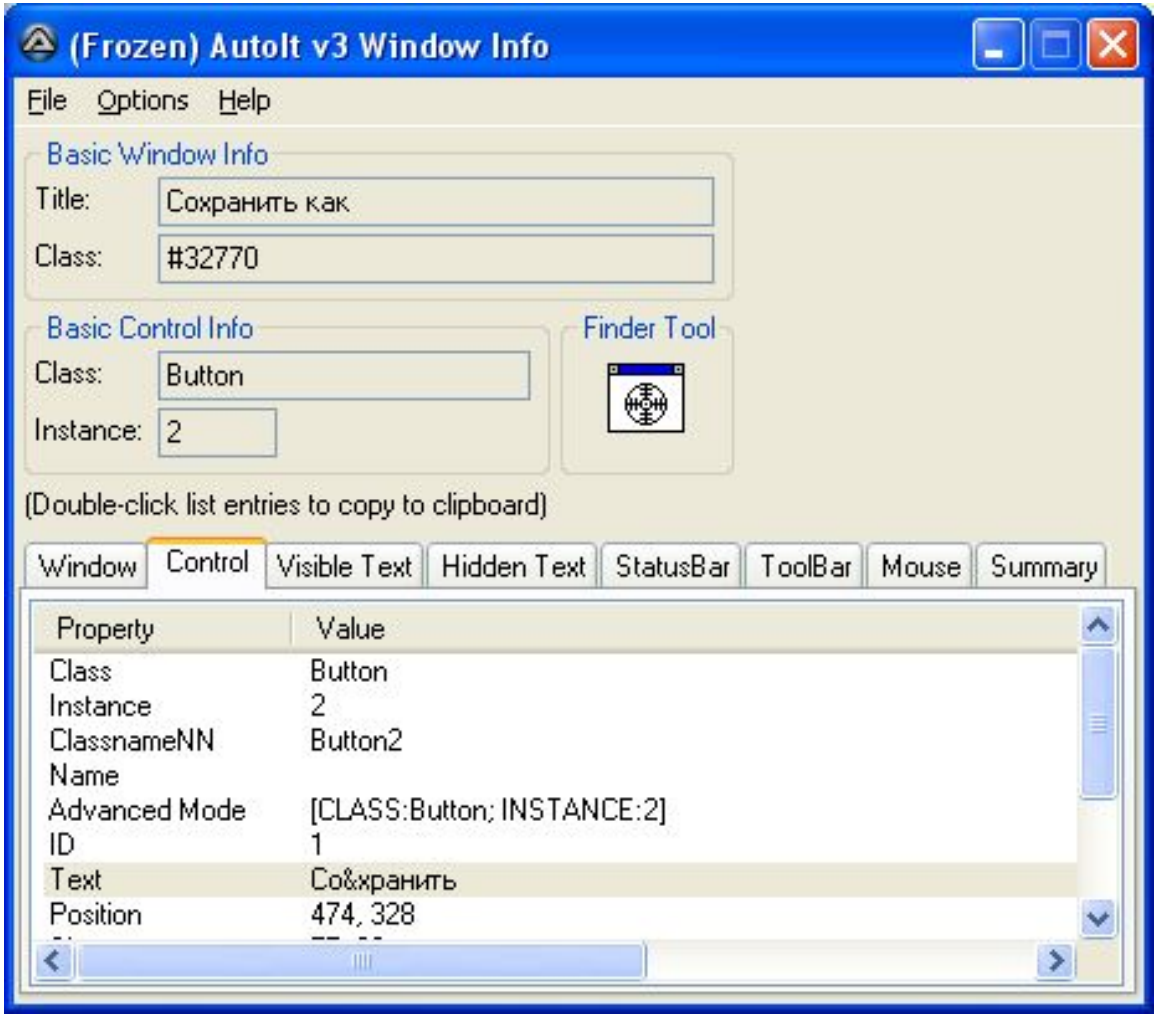
Подход AutoIt

В основе скритповый язык, подобный Basic.

Window Spy - позволяет получать атрибуты окон и контролов приложения.

Язык позволяет искать окна и контролы и отправлять им сообщения.

AutoIt “шпион”



Подход AutoIt

```
Run("notepad.exe")
WinWaitActive("Безымянный - Блокнот")
Send("Проверяем сохранение файла.")
WinClose("Безымянный - Блокнот")
WinWaitActive("Блокнот", "Да")
Send("!д")

WinWaitActive("Сохранить как", "*.txt")
Send("1.txt")
WinWaitActive("Сохранить как", "Со&хранить")
Send("!х")
```

Подход AutoIt

Кроме работы с GUI способен работать и с COM.

COM – базовая технология Windows.

Фактически она позволяет тестировать объекты и их свойства. Ближе к модульному тестированию.

Проблемы AutoIt

Скрипты привязываются к типам окон и текстам в окнах.

В процессе разработки и то и другое может меняться – тесты перестанут работать.

Требует сложной поддержки совокупности тестов.

Сильные стороны автоматизации

- Повторяемость – все написанные тесты всегда будут выполняться однообразно и безошибочно, то есть исключен «человеческий фактор».
- Быстрое выполнение.
- Эффективность – когда автоматические скрипты уже написаны, на их поддержку и анализ результатов требуется, как правило, меньшее время чем на проведение того же объема тестирования вручную.
- Отчеты – рассылаются автоматически.
- Выполнение без вмешательства – может выполняться в нерабочее время.

Проблемы автоматизации

- Узость тестирования – проверяет только то, на что запрограммирован.
- Затраты на поддержку – при изменениях (даже не функциональных) необходимо менять скрипты.
- Большие затраты на разработку – разработка автоматизированных тестов это сложный процесс, требующий времени и квалификации.
- Стоимость инструмента для автоматизации – в случае использования платного ПО, его стоимость может быть достаточно высока.

Контрольные вопросы

1. Какие основные шаги автоматизируют при автоматизации тестирования?
2. Из чего складываются затраты на автоматизацию тестирования?
3. Какие приложения позволяет тестировать Selenium?
4. Какие 3 уровня разработки тестов имеется в Selenium?
5. Что такое XPath?
6. Какая из систем автоматизации тестирования рассмотрена применительно к настольным приложениям Windows?
7. В чем заключаются сильные стороны автоматизации?
8. В чем заключаются проблемы автоматизации?

Решение по автоматизации

Решение об автоматизации не должно приниматься спонтанно. Это приведет к:

- Неэффективному расходованию ресурсов;
- Не будут достигнуты цели;

Необходимо ответить на 3 вопроса:

- Зачем автоматизировать?
- Что автоматизировать?
- Как автоматизировать?

Зачем, что и как

- Сократить время на сложные операции
- Сократить время на многочисленные повторы (регрессионное)
- Гарантировать правильную работу критически важных функций

Зачем, **что** и как

- Где принесет наибольший эффект
- Где потребуются наименьшие затраты на разработку
- Где потребуются наименьшие затраты на поддержку

Что автоматизировать (кандидаты)

Сложно проверить:

- Появление записей в БД
- Форматы создаваемых файлов
- Расчеты

Сложно, но важно провести:

- Ключевые сложные функции UI
- Рутинные, требующие большого количества данных

Что автоматизировать (кандидаты)

- Циклические сценарии (end-to-end), например:
  - Создать, прочесть, изменить, удалить сущность БД.
  - Авторизоваться, создать письмо, отправить письмо получателю и выйти.

Такие сценарии автоматически могут повторяться и не влияют на другие тесты.

Зачем, что и **как**

- Какие инструменты использовать? (*часто стандарт компании*)
- Что можно сделать без UI (*модульные тесты*)?
- Насколько велика вероятность изменений в проекте?
- Как можно снизить стоимость поддержки автоматизированных тестов?
- Нельзя ли применить частичную автоматизацию, на которой мало скажется изменение UI?

Управление тестами

- Объем регрессионного тестирования – один из важных показателей к автоматизации.
  - *С этой целью целесообразно для каждого релиза, который передается на тестирование выполнять все автоматизированные тесты.*

- Нужно продумать и организовать хранилище и документирование разработанных автоматизированных тестов.
  - В случае внесения изменений в реализацию важно понять какие из автоматизированных тестов должны быть изменены.

- Нужен регламент выполнения автоматизированных тестов. Система тестов требует регулярного применения.
  - На начальном этапе автоматизированные тесты выполняются в ручном режиме. Таким образом сами проходят проверку.

Заключение
Тестирование ПО и автоматизация

Более широкий взгляд на автоматизацию тестирования – создание оснастки для повышения эффективности тестирования.

Речь об узкоспециализированных утилитах, который облегчают работу тестировщиков.

Примеры.

- Система собирает online-данные с оборудования (медицинского или иного) и отображает их пользователю. Тестируется именно функция такого сбора и online-отображения. *Естественным шагом* автоматизации будет разработка программного имитатора источника данных.

- Тестируется восстановление Системы (recovery) после “краха”. *Естественным шагом* будет подготовка скрипта для восстановления БД и конфигурации Системы;

> Автоматизация для повышения эффективности тестирования важна и шире, чем рассматривается в академическом разрезе.

Контрольные вопросы

1. Назовите 3 ключевых вопроса, рассматриваемые при автоматизации тестирования.
2. Приведите вариант мотива к автоматизации тестирования.
3. Приведите пример кандидата к автоматизации.
4. Приведите пример решений о способе и глубине автоматизации тестирования.
5. Какие управленческие решения должны быть приняты при автоматизации?