Automated integration tests for AJAX applications

Overview

CEE-SECR 2011
October 31 – November 3, Moscow

AURIGA®

Elite Software R&D Services
Since 1990

Justification

- Who is part of development team?

Who is using …

- automated integration (functional) testing
- unit-testing
- manual testing

… to track quality?

- Why or how?

Agenda

- Motivation
- Challenges
- Solutions & tricks
- Experience
- Conclusion

Motivation – goals

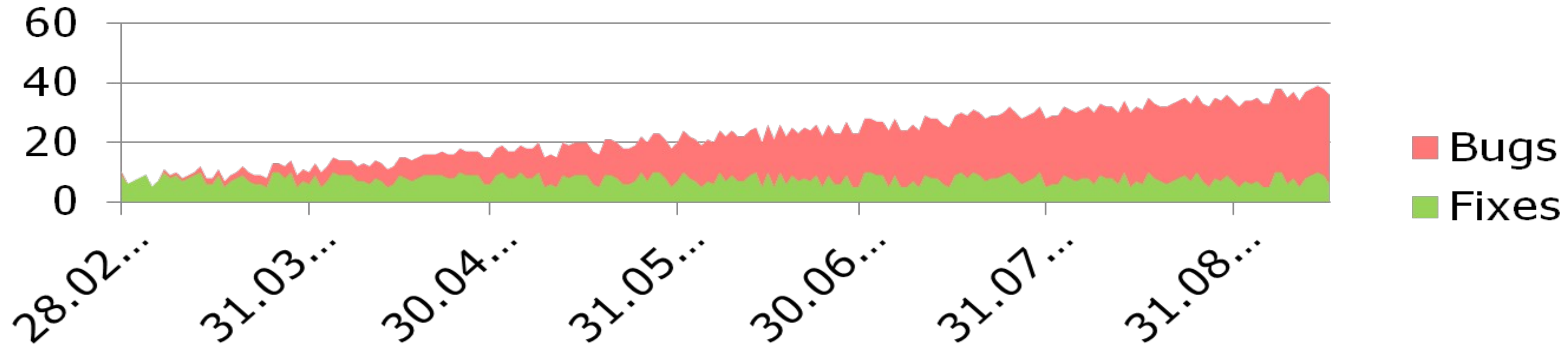
- **Reliability** of the product **at any moment**
- Less time for bugs – more **time for development**
- **Feel free** to add new features – it's safe

Motivation – issues

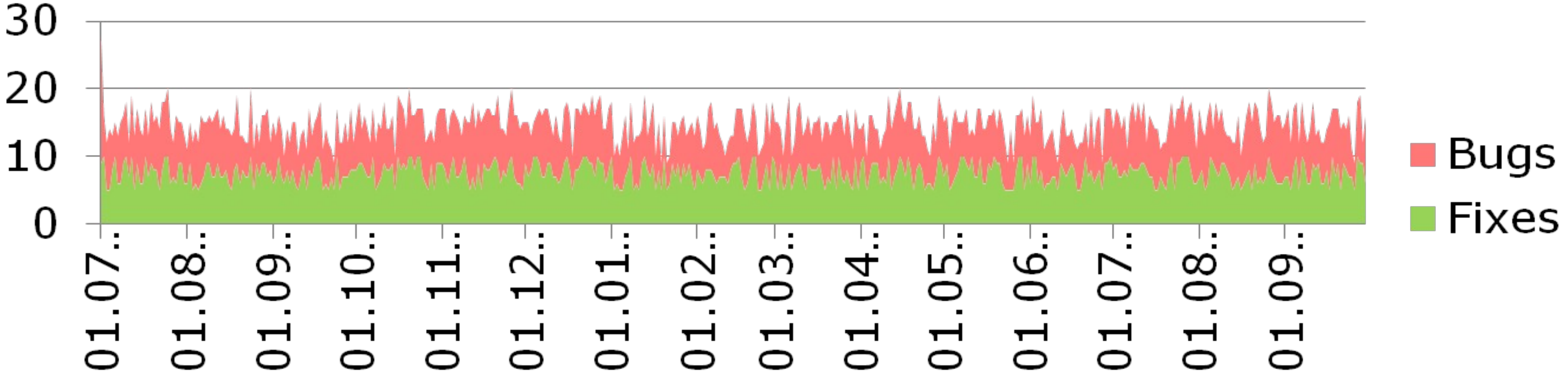
- Manual testing is **slow**
- **Hard to determine** guilty commit

Reliability at any moment

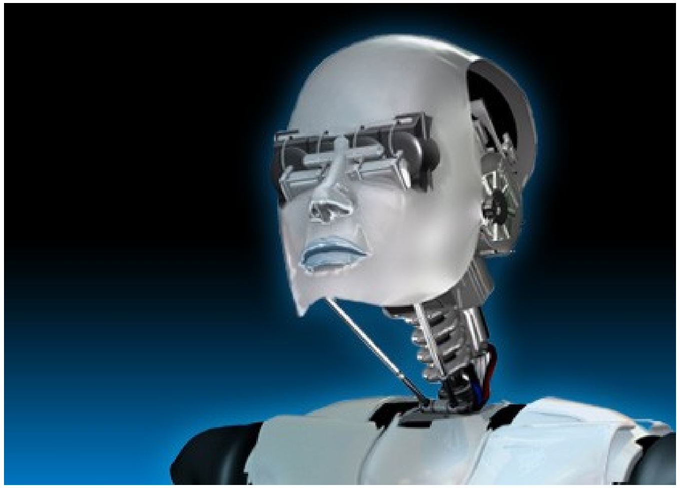
Worst case is:



The goal is:



Automated tests for AJAX web-applications



Challenges - I

- **Recent framework** compared to unit testing
- **Full-featured container** required
- **New approach** (API)
- Lack of tools
- Relatively **faster**, but …. relatively **slow**

Challenges - II

- **Asynchronous behavior**
- Tricky to configure **Hudson/Jenkins**
- Complicated **staging environment** configuration
- Lack of **experienced QA** engineers

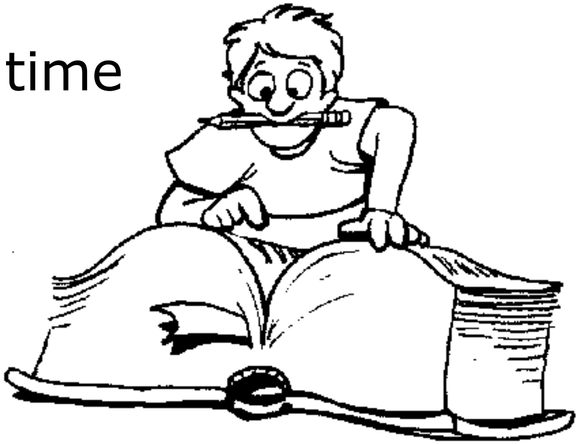
Solutions

Challenges

- Recent framework compared to unit testing
- Tricky to configure Hudson/Jenkins
- Lack of experienced QA engineers

Solution

- Well… Everything happens for the first time
- Research, study and teach



Solutions

Challenges

- Complicated staging environment configuration

Solution

- Test driven application architecture – planning in advance is a key
- Use IoC based approach for applications

Solutions

Challenge

- Requires full-featured container
- Requires new approach (API)

Solution

- Use Selenium, jWebUnit

Solutions

Challenge

- Tools… Just few of them

Solution

- jWebUnit
  - In case of simple ajax
  - Turn off javascript if not needed (save 30 seconds)
- Selenium
  - Need to control items visibility
  - Need to test against real browser

Solutions

Challenge

- Relatively faster, but… relatively slow

Solution

- Still much faster than manual tests
- Define AllInOne suite for quick check
- Run full set of tests on CI
- Use distributed technologies (i.e. Selenium Grid)

Solutions

Challenge

- Hard to determine guilty commit
- Intend to add new features safely

Solution

- Test all before commit

- or at least -

- Email notification from CI that commit produced regression

Solutions

Challenge

- Asynchronous behavior

Solution

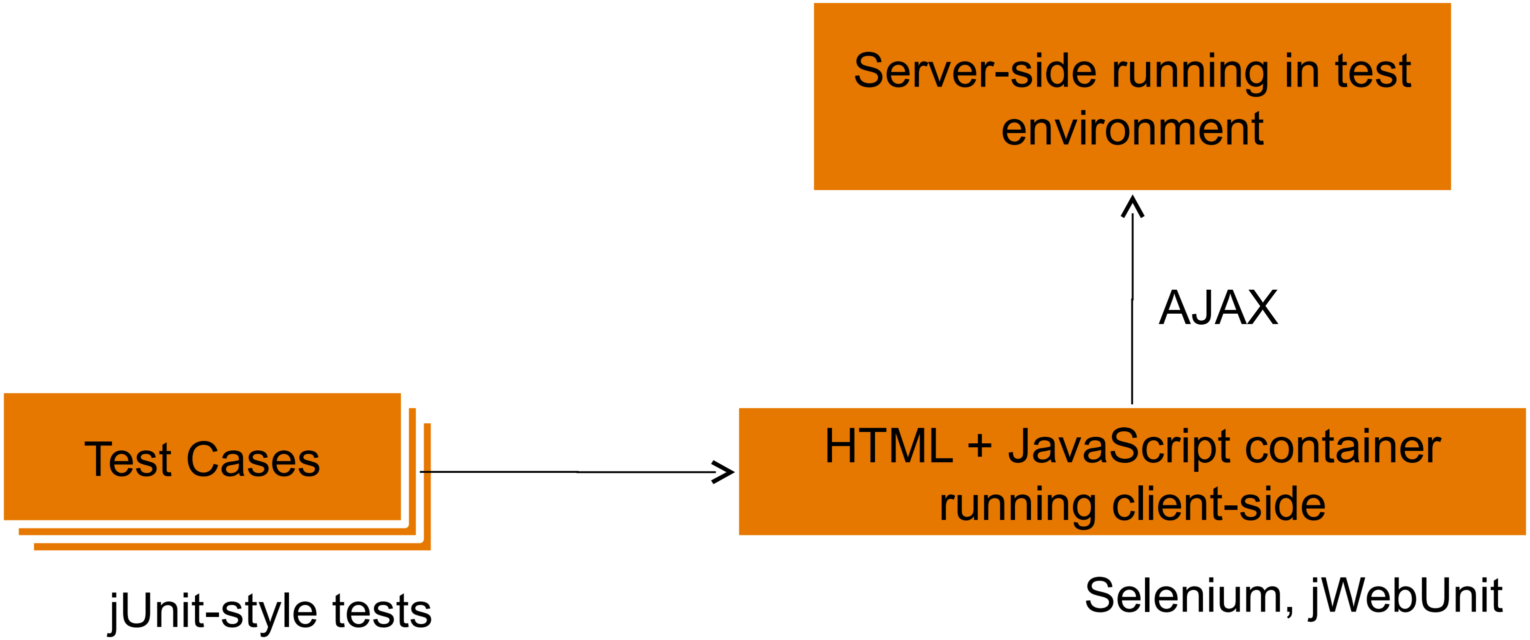
- Use wait() approach
- Use Thread.sleep() after each action
- Be aware of DOM inconsistency

Experience

Our experience with AJAX apps testing:

- **GWT** application (100% AJAX)
  - **GWTTestCase is not recommended for functional testing**
- Regular **WEB 2.0** application

The architecture


Server-side running in test environment
AJAX
Test Cases
HTML + JavaScript container running client-side
jUnit-style tests
Selenium, jWebUnit

Source code example (jWebUnit) – 1/3

```
public class TestShoppingCart extends BaseTest {
  @Test
  public void test() {
    beginAt("/cart/index");
    tester.assertTextPresent("Your cart is empty.");
    tester.setScriptingEnabled(true);
    beginAt("/album/list?sort=albumName&max=20");
    tester.assertElementPresentByXPath("//div[@class='list']");
    tester.assertElementPresentByXPath("//div[@id=‘c' and .='0 items']");
    tester.clickElementByXPath("//div[@class='album-
buy']/input[contains(@value,'Buy Album')]");
    tester.assertElementPresentByXPath("//div[@id=‘c' and .='1 item']");
    tester.clickElementByXPath("//div[@id='cart']//a");
    tester.assertElementPresentByXPath("//div[@id='ribbon-single-
line']/h1[contains(., 'Shopping Cart')]");
```

Source code example (jWebUnit) – 2/3

```
public class TestShoppingCart extends BaseTest {
  @Test
  public void test() {
    beginAt("/cart/index");
    tester.assertTextPresent("Your cart is empty.");
    tester.setScriptingEnabled(true);
    beginAt("/album/list?sort=albumName&max=20");
    tester.assertElementPresentByXPath("//div[@class='list']");
    tester.assertElementPresentByXPath("//div[@id=‘c' and .='0 items']");
    tester.clickElementByXPath("//div[@class='album-
buy']/input[contains(@value,'Buy Album')]");
    tester.assertElementPresentByXPath("//div[@id=‘c' and .='1 item']");
    tester.clickElementByXPath("//div[@id='cart']//a");
    tester.assertElementPresentByXPath("//div[@id='ribbon-single-
line']/h1[contains(., 'Shopping Cart')]");
```

Source code example (jWebUnit) – 3/3

```
public class TestShoppingCart extends BaseTest {
  @Test
  public void test() {
    beginAt("/cart/index");
    tester.assertTextPresent("Your cart is empty.");
    tester.setScriptingEnabled(true);
    beginAt("/album/list?sort=albumName&max=20");
    tester.assertElementPresentByXPath("//div[@class='list']");
    tester.assertElementPresentByXPath("//div[@id=‘c' and .='0 items']");
    tester.clickElementByXPath("//div[@class='album-
buy']/input[contains(@value,'Buy Album')]");
    tester.assertElementPresentByXPath("//div[@id=‘c' and .='1 item']");
    tester.clickElementByXPath("//div[@id='cart']//a");
    tester.assertElementPresentByXPath("//div[@id='ribbon-single-
line']/h1[contains(., 'Shopping Cart')]");
```

Source code example (Selenium) – 1/2

```
public class TestManageChart extends CatalogMerchTestBase {
    @Test
    public void testManageChart() throws Throwable {
        driver.get(baseUrl + "#top.store[1].manage_charts");
        waitV(genId("btn-add-chart"));
        WebElement elm = driver.findElement(By.id("btn-add-chart"));
        click(elm);

        Alert alert = getAlert();
        alert.sendKeys(newChartName);
        alert.dismiss();
        waitInvAjax();
```

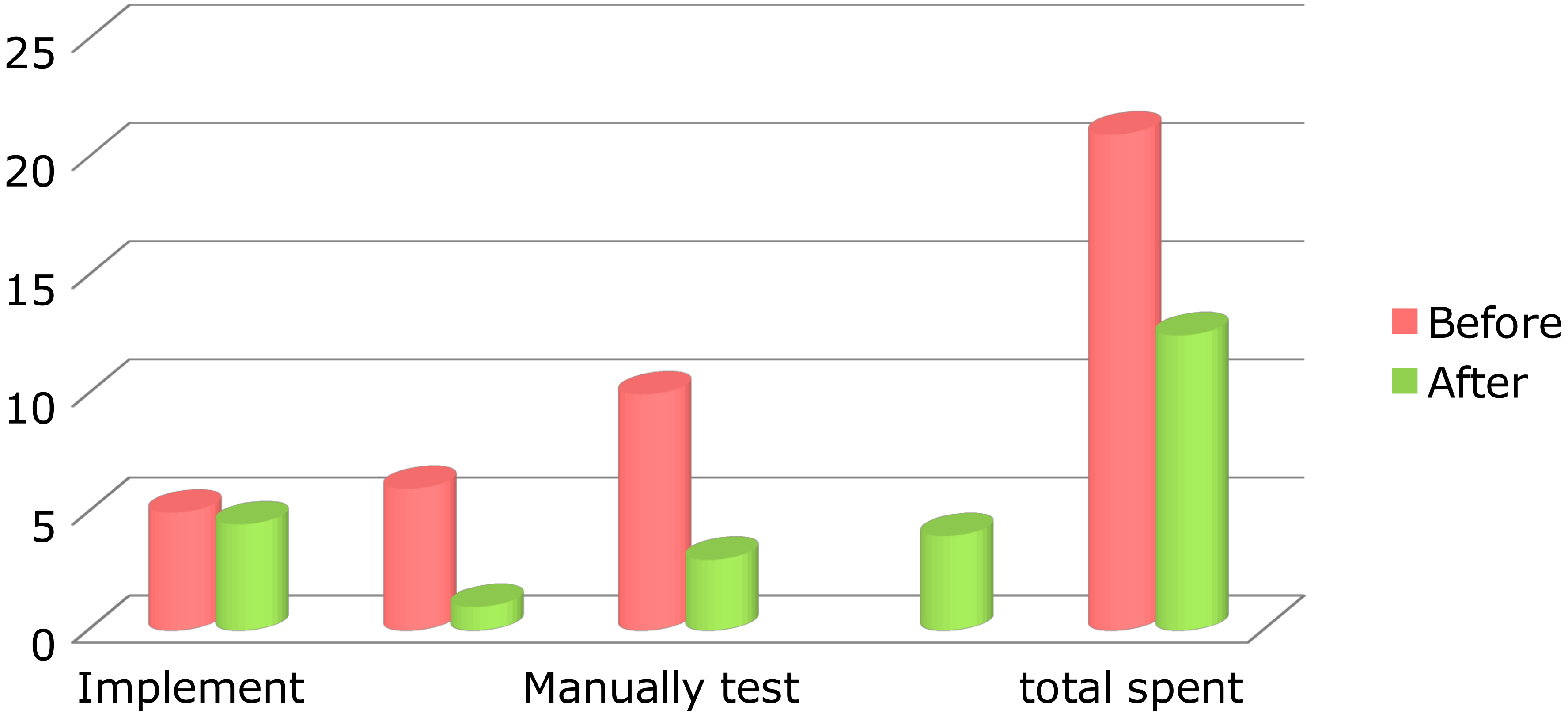
Source code example (Selenium) – 2/2

```
public class TestManageChart extends CatalogMerchTestBase {
    @Test
    public void testManageChart() throws Throwable {
        driver.get(baseUrl + "#top.store[1].manage_charts");
        waitV("btn-add-chart");
        WebElement elm = driver.findElement(By.id("btn-add-chart"));
        click(elm);

        Alert alert = getAlert();
        alert.sendKeys(newChartName);
        alert.dismiss();
        waitInvAjax();
```

Experience

Time spent. Grouped by task type



Conclusion & Questions

- It’s functional testing
- It’s worth it!

Note:

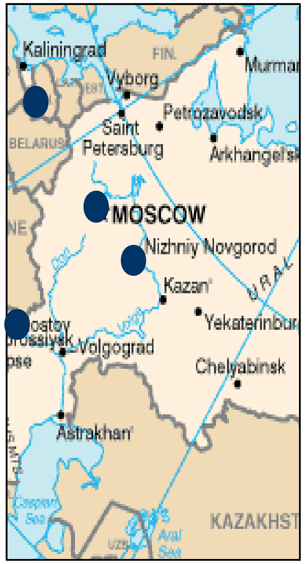
- Write code with low TCO!
- Perform code review!
- Don’t let user story overstep the sprint!

Auriga At a Glance

- **World leader** in R&D outsourcing
  - **#1 Engineering Services Outsourcing (ESO)** provider
  - **#15** in **Top-20 Global Outsourcing Vendors**
  - **Global Services 100** and **Global Outsourcing 100** company
  - **Top 10** Offshore **Eastern and Central Europe**
  - **Top 10 Software R&D** Service Providers
- **In business since 1990**
  - First in Russia to do offshore software services
- **300 employees**
- **Headquartered in the U.S.**
  - Incorporated in the U.S. in 1993
- **Engineering centers in**
  - **Russia**: Moscow, Nizhny Novgorod, Rostov-on-Don
  - **EU**: Vilnius, Lithuania
- **Main Domains: Technologies and Verticals**

- System Level, Embedded
- Mobile
- Document Management
- Portals
- Social Web
- Media and Web

- Technology (s/w and h/w)
- Finance
- Healthcare
- Telecom
- Logistics
- Government
- Security



Great Lakes
Boston
New York
Philadelphia
WASHINGTON, D.C.
Jacksonville
NORTH ATLANTIC OCEAN
Miami
BAH.
CUBA
DOM. REP.
Puerto Rico (U.S.)
JAM.
HAITI

Contacts

Thank You!

Sergey.Karpushin@auriga.com

Arkadiy.Hachikyan@auriga.com