Ajax Debugging and Quality Assurance

AjaxWorld

Dave Johnson

Oct 3, 2006

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
   I. Unit Testing
   II. Functional Testing
   III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

Who is this Guy?

Dave Johnson

(that's me)

Co-founder + CTO
Nitobi Software

(formerly eBusiness Applications)

http://blogs.nitobi.com/dave

Nitobi Enterprise AJAX Podcast

Enterprise AJAX – Addison Wesley

nitobi
built for people

Imperial College
London

Who is this Guy? Who is nitobi?

nitobi
built for people

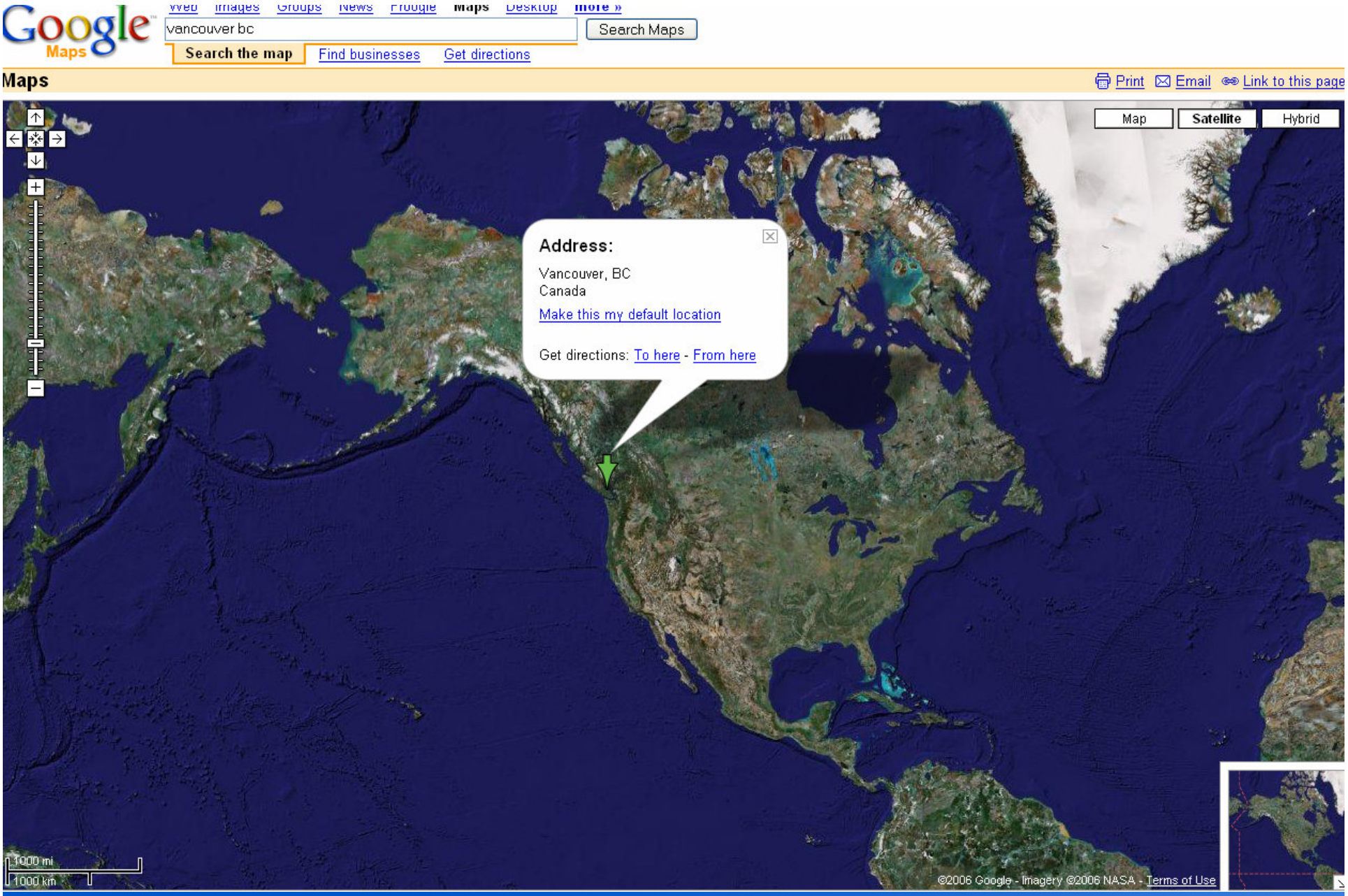
Who is this Guy? Who is nitobi?

nitobi
built for people

Who is nitobi?

Ajax User-Interface Components for the Enterprise

Who is this Guy? Who is nitobi?



Who is this Guy? Who is nitobi?

nitobi
built for people

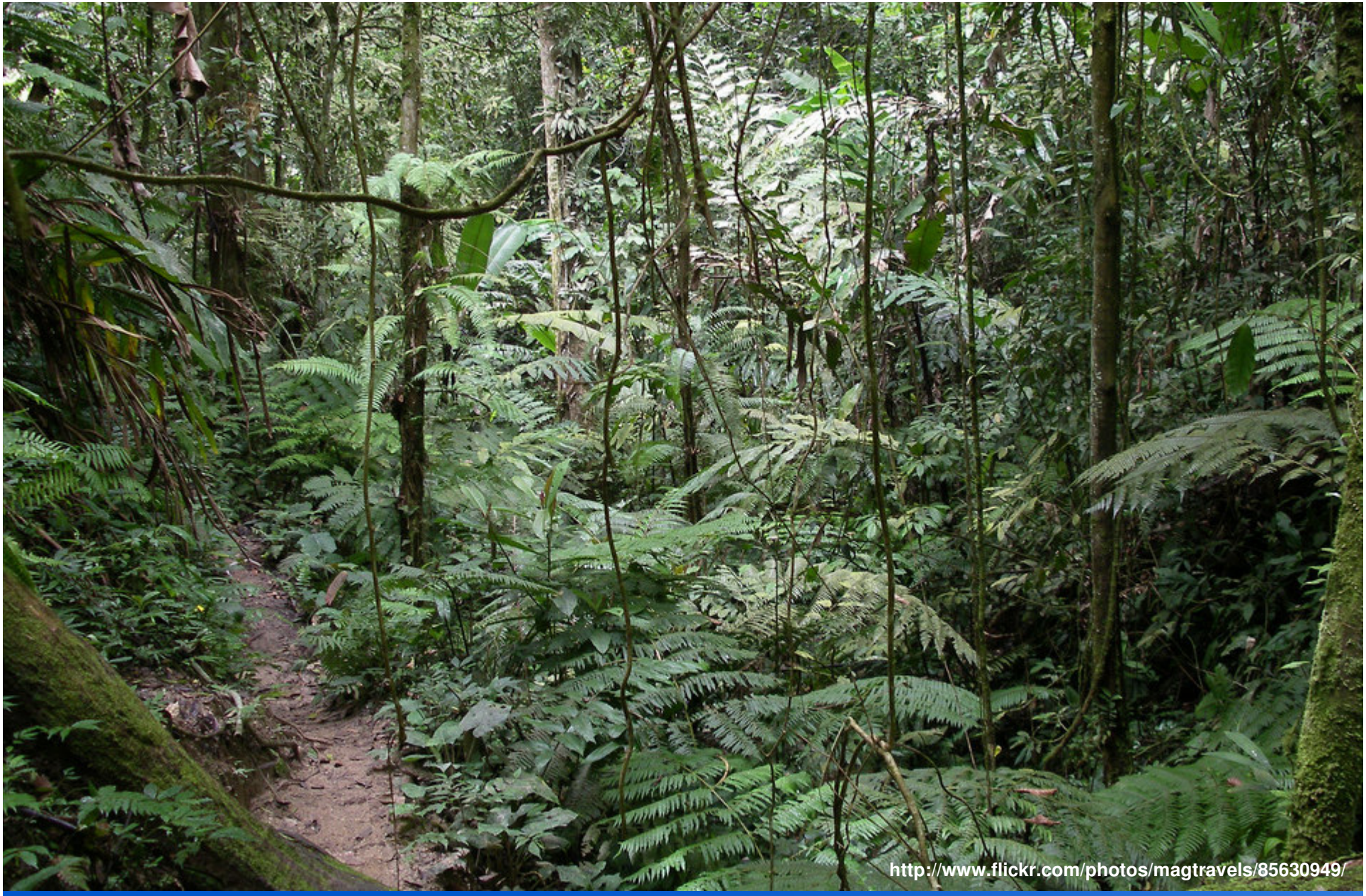
Our Customers

Who is this Guy? Who is nitobi?

nitobi
built for people

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
    I. Unit Testing
    II. Functional Testing
    III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions



Testing in Brief

nitobi
built for people

Quality Assurance

- Widely introduced during WWII
- Munitions industry demanded better quality and more reliable “products”
- Quality systems evolved to ISO 9001

“rely on prevention rather than cure*”

*we will get to curing later

Why Bother

- Riches?
- Fame?
- Becoming a Jedi?

It’s About the Bugs Stupid

- Testing is about finding bugs
- Added bonus!

nitobi
built for people

Not _All_ About Bugs

- Performance (later)
- Usability (some other time)
- Accessibility (read the book)

Development Philosophy

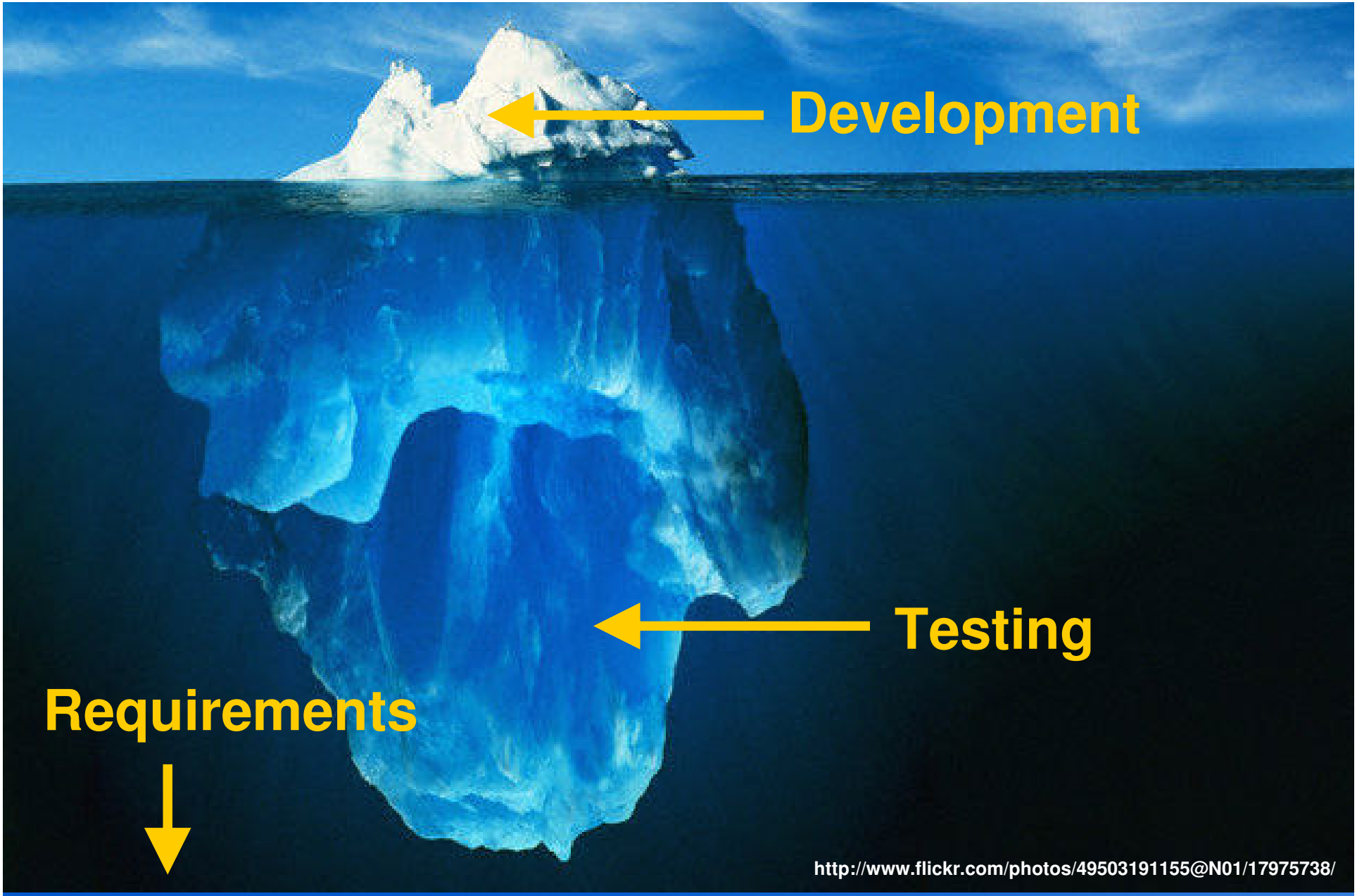
- Lots of development philosophies
  - Waterfall
  - Iterative
  - Agile / Extreme
  - <!-- insert cool new technique here -->

Testing in Brief

nitobi
built for people

Many Stages of Testing

- Requirements
- Design
- Test Planning
- Test Development
- Test Execution
- Test Reporting



nitobi
built for people

Don’t Forget

- Things as simple as bad documentation, or difficult tools can sink a product

nitobi
built for people

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
    I. Unit Testing
    II. Functional Testing
    III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

Unit Testing

nitobi
built for people

Tools

- JSUnit
- JSMock
- HTTPUnit

nitobi
built for people

JSUnit

```
function setUpPage()
{
    setUpPageStatus = 'complete';
}

function setUp()
{
    //…
}

function testDoSomething()
{
    assertEquals(“a”, ”b”);
}
```


nitobi
built for people

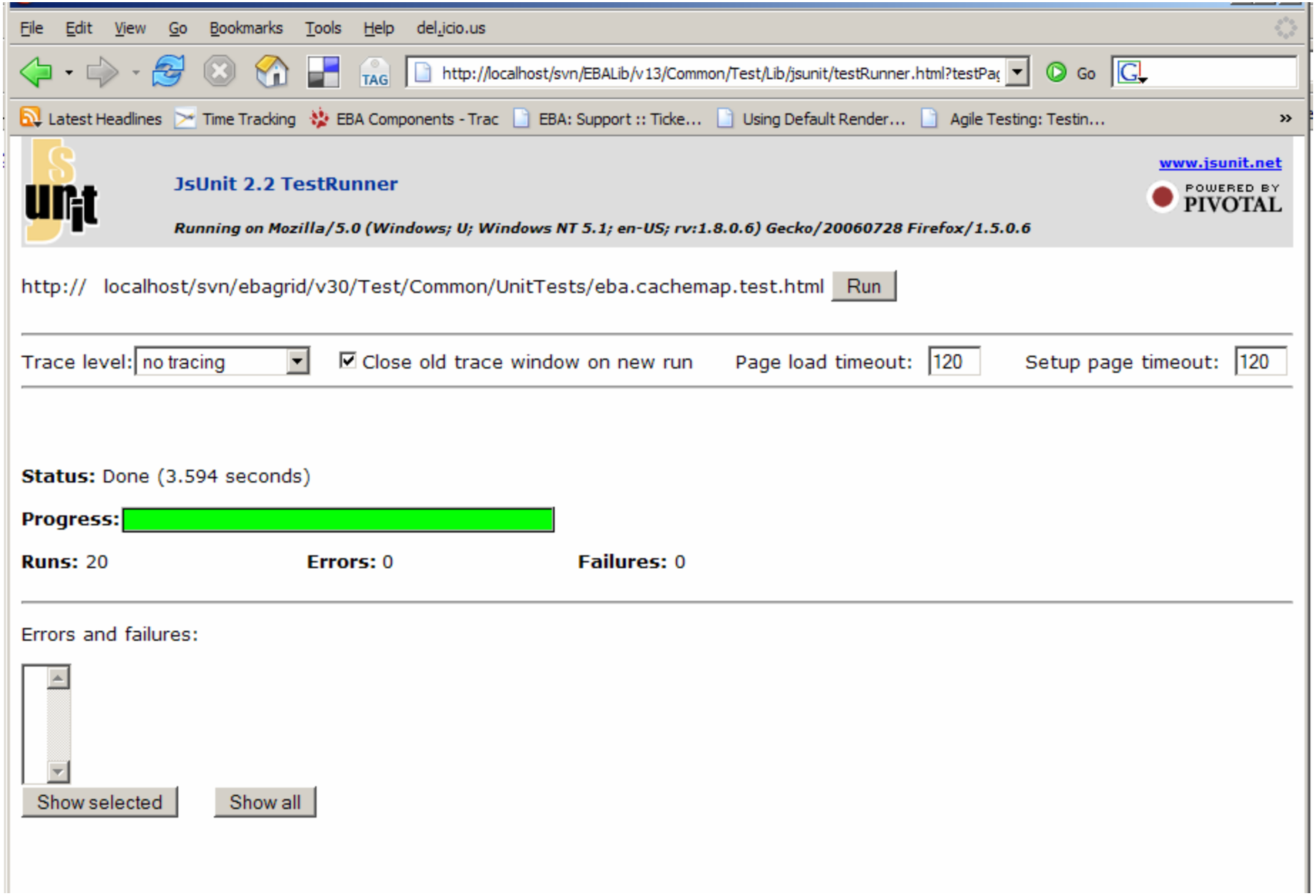
Method Overview

- setUpPage()

- setUp()
- tearDown()

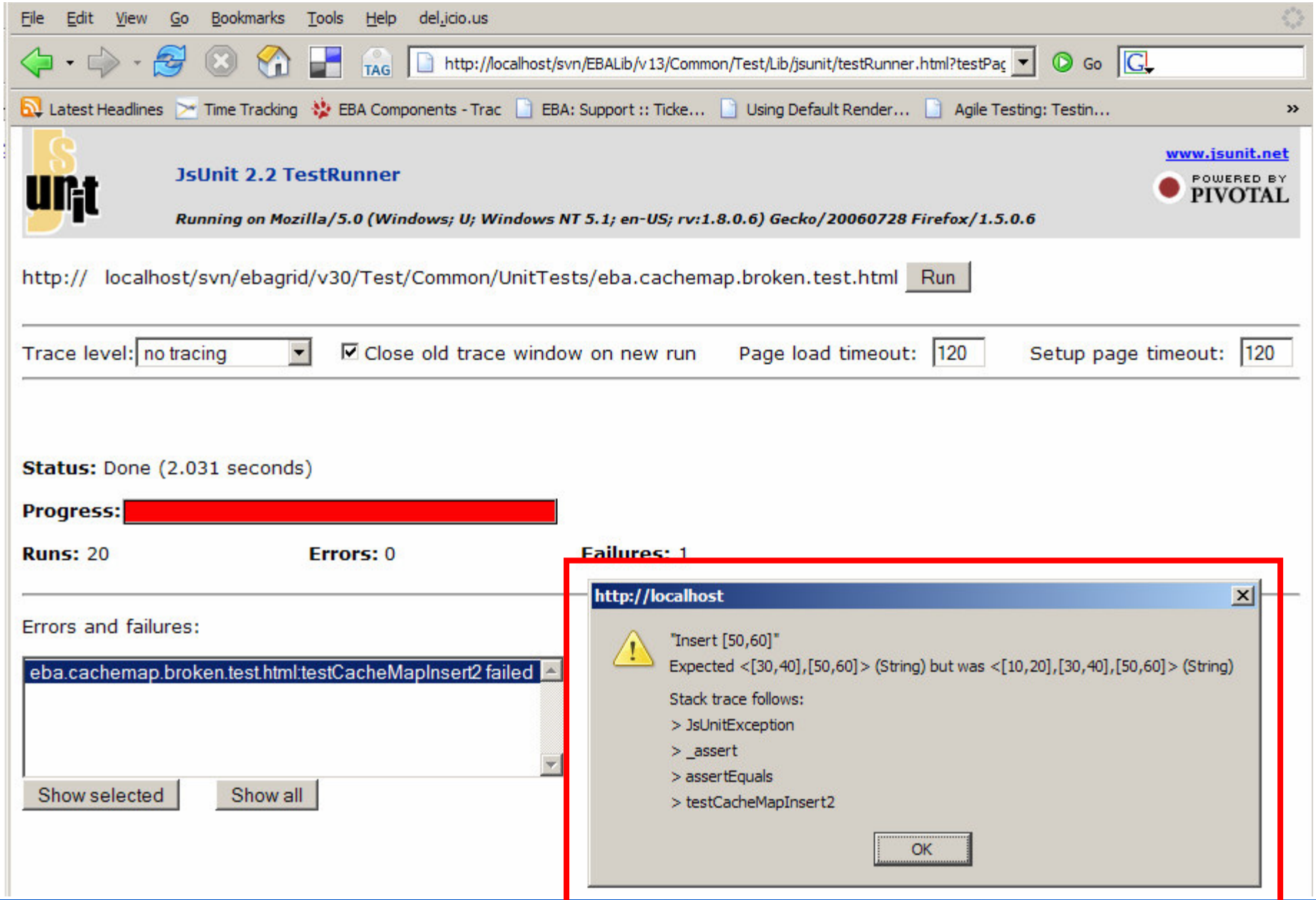
- warn(message, [value])
- inform(message, [value])
- debug(message, [value])

- addTestSuite(filename)



Unit Testing

nitobi
built for people



Unit Testing

nitobi
built for people

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
    I. Unit Testing
    II. Functional Testing
    III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

Functional Testing

nitobi
built for people

Functional Testing

- Test the software according to the specification
- Functional testing steps:
  - Identification of functions that the software is expected to perform
  - Creation of test data that will exercise those functions

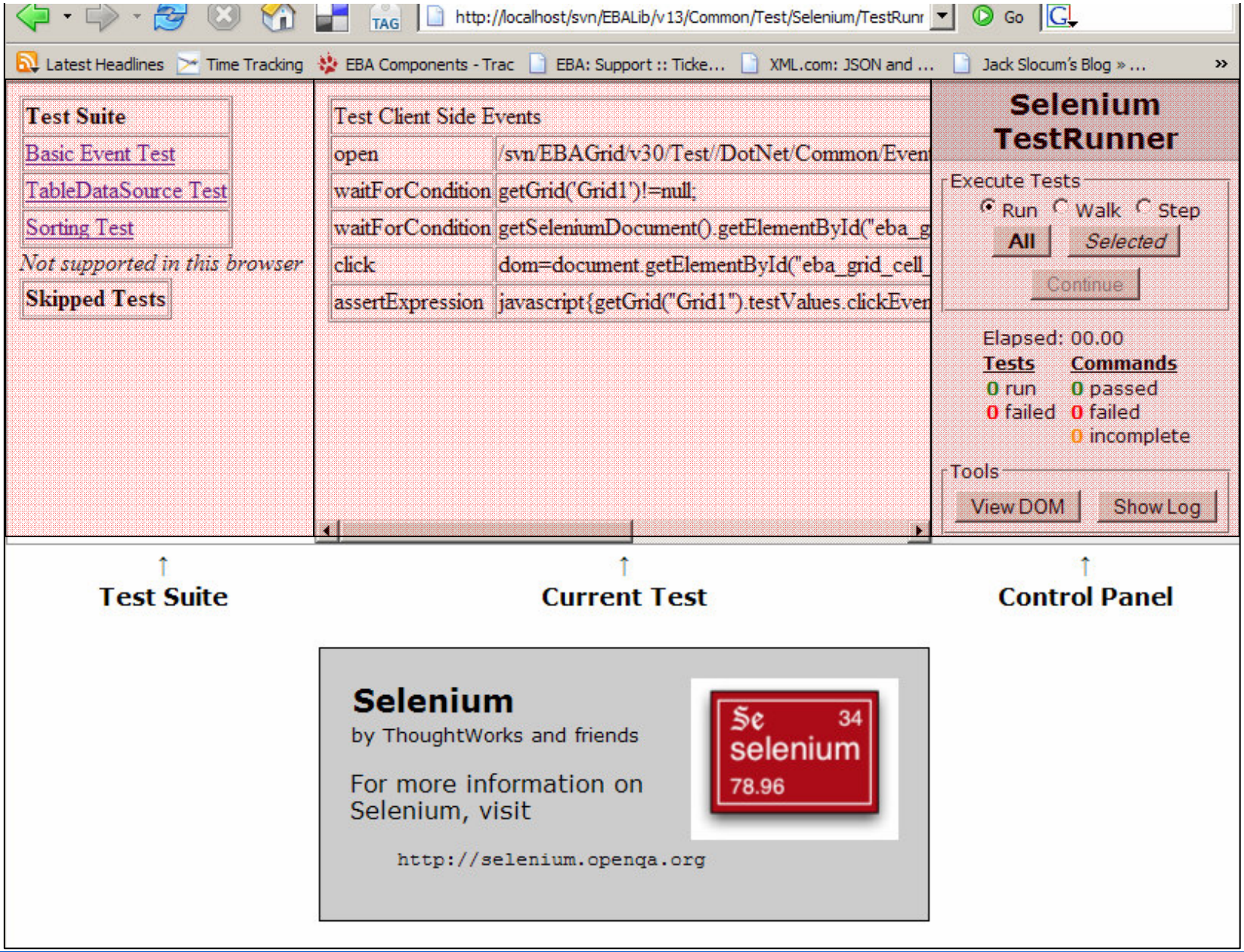
nitobi
built for people

Rule of Thumb

- For every tenfold increase in defects discovered an additional tester must be added

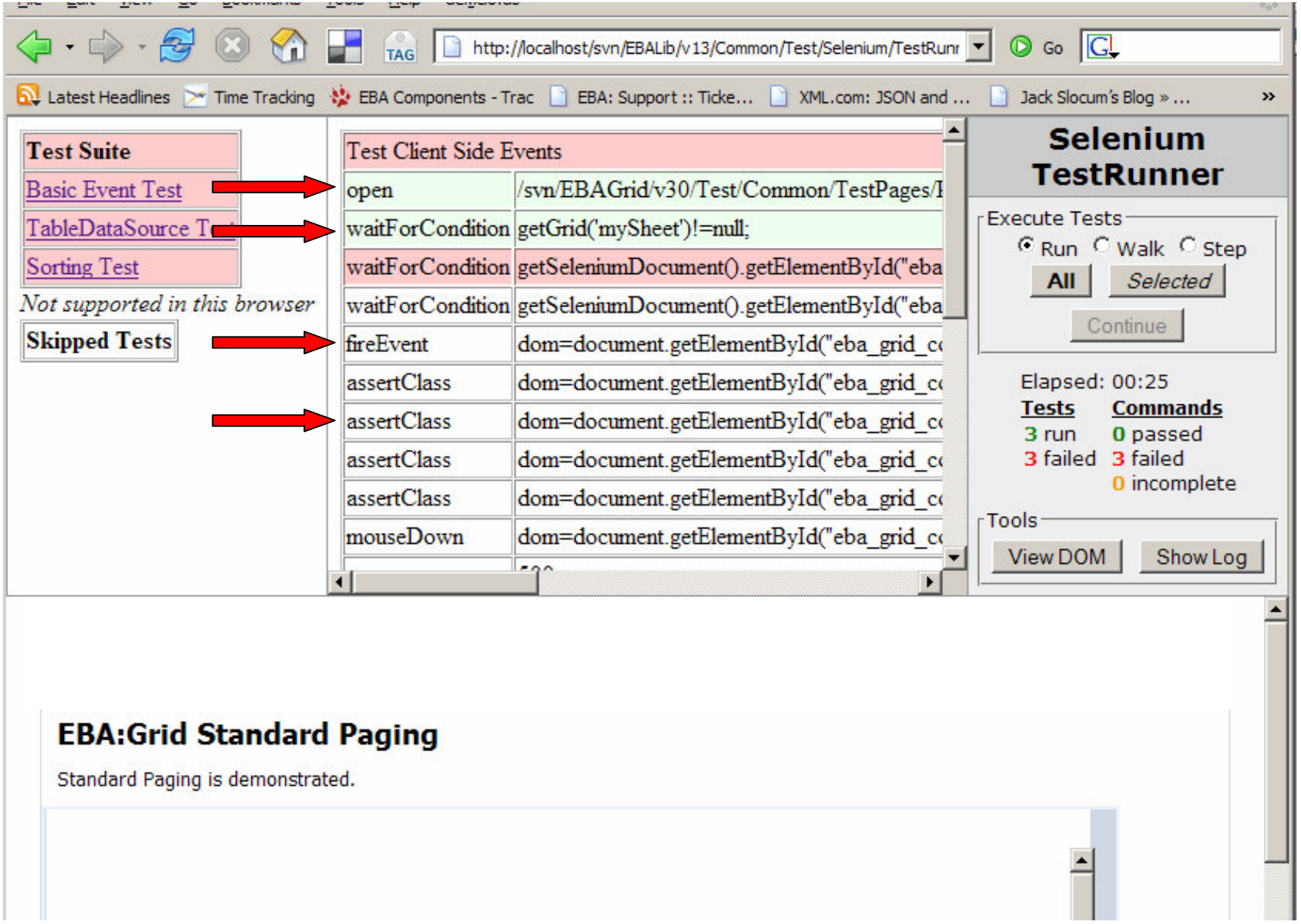
Tools

- Selenium
  - Core
  - IDE
  - Remote Control
- Watir
- jWebUnit



Functional Testing

nitobi
built for people

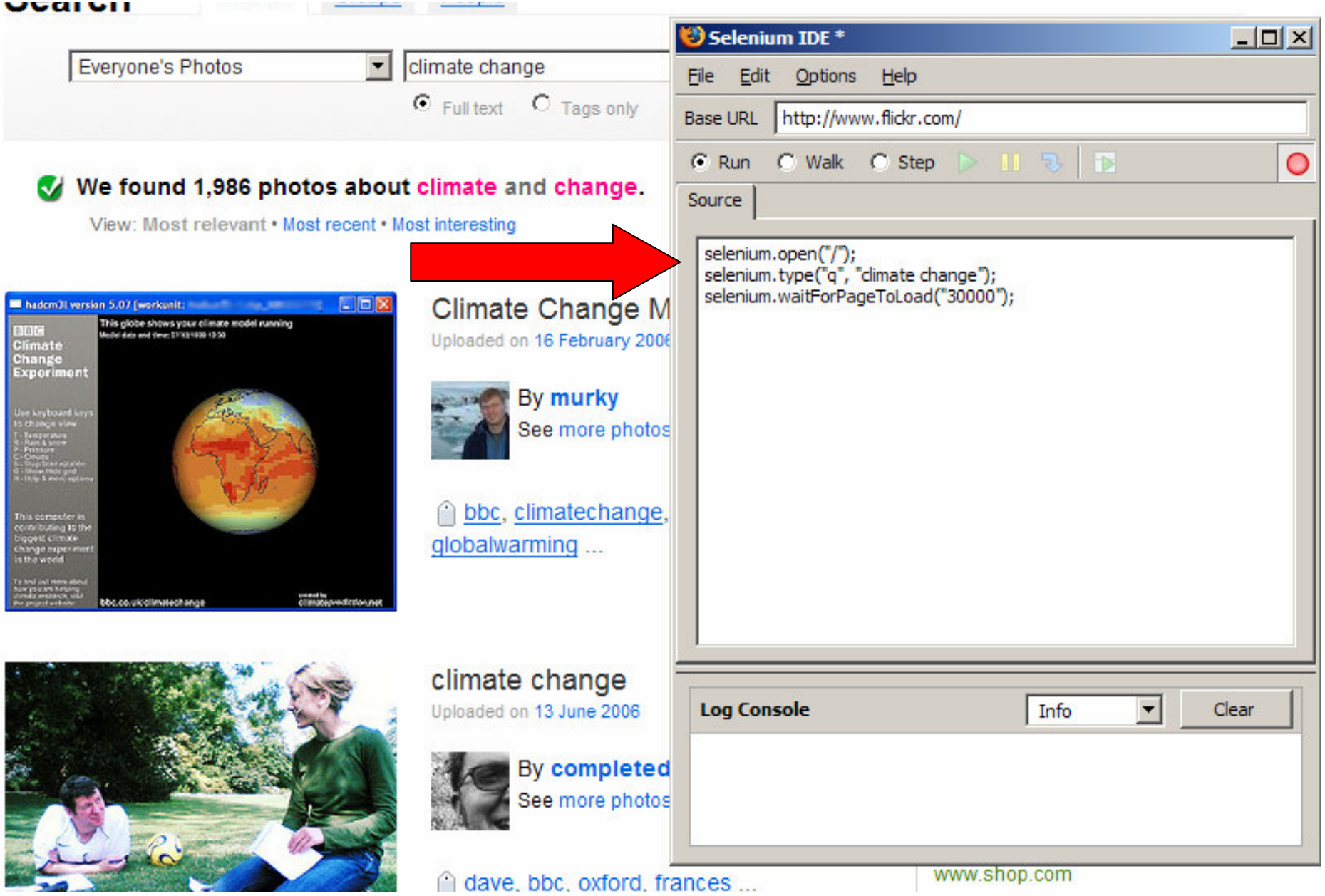


Functional Testing

nitobi
built for people

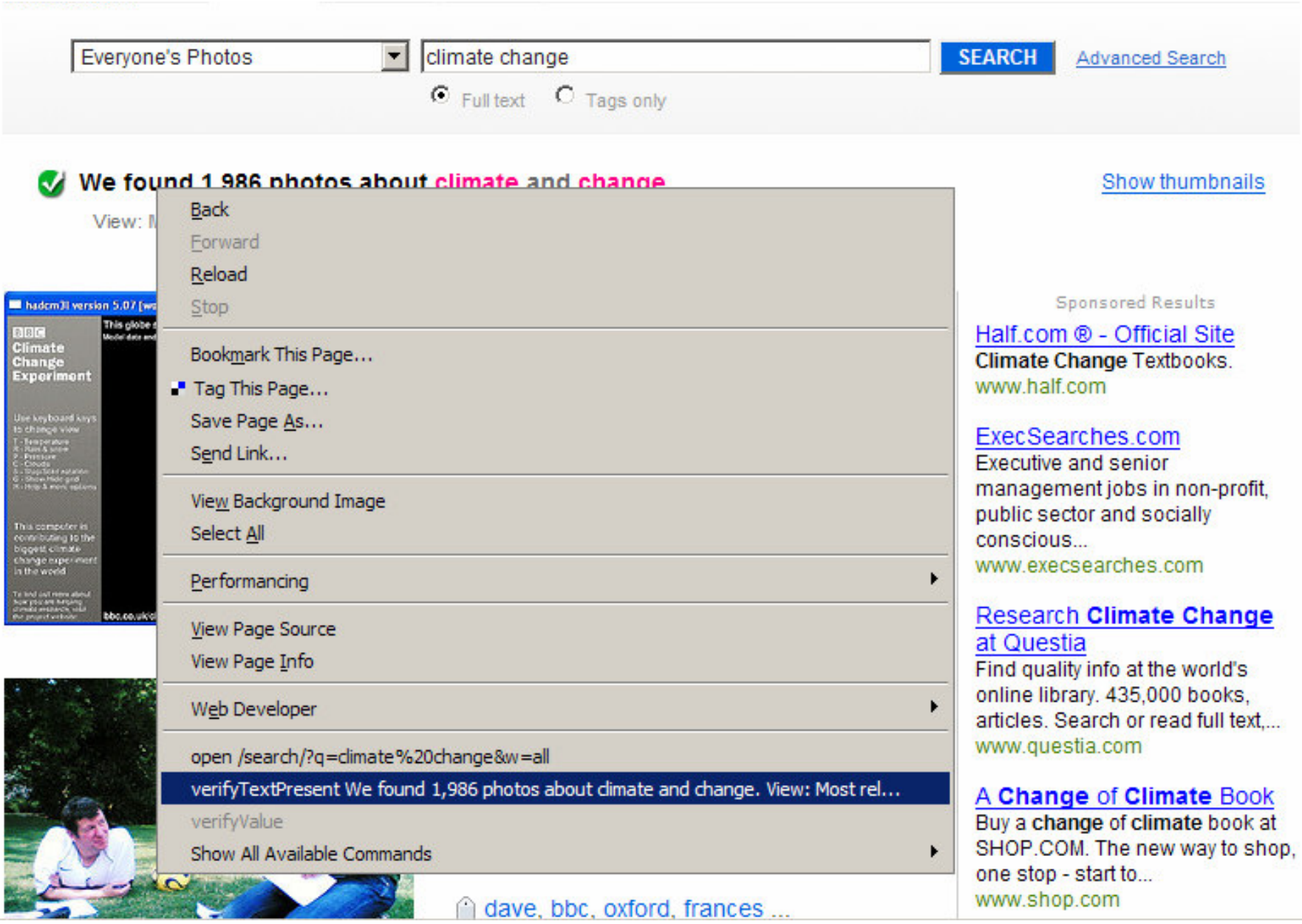
This is a Test?

```
<table>
<tr><td>setVariable</td><td>username</td><td>'user'+(new Date()).getTime()</td></tr>
<tr><td>open</td><td>./selenium_ft_tool/setup?user=${username}</td><td></td></tr>
<tr><td>setVariable</td><td>base_url</td><td>'http://www.example.com:8080/Plone'</td></tr>
<tr><td>open</td><td>${base_url}</td><td></td></tr>
<tr><td>type</td><td>__ac_name</td><td>${username}</td></tr>
<tr><td>type</td><td>__ac_password</td><td>${username}</td></tr>
<tr><td>click</td><td>submit</td><td></td></tr>
<tr><td>verifyTextPresent</td><td>Welcome! You are now logged in</td><td></td></tr>
<tr><td>setVariable</td><td>myfolder_url</td><td>'${base_url}/${username}/folder'</td></tr>
<tr><td>click</td><td>//a[@href='${myfolder_url}']</td><td></td></tr>
<tr><td>verifyTextPresent</td><td>${username}</td><td></td></tr>
<tr><td>setVariable</td><td>homepage_url</td><td>'${base_url}/${username}/index/view'</td></tr>
<tr><td>click</td><td>//a[@href='${homepage_url}']</td><td></td></tr>
<tr><td>open</td><td>${member_url}</td><td></td></tr>
<tr><td>verifyTextPresent</td><td>Home page for ${username}</td><td></td></tr>
<tr><td>verifyTextPresent</td><td>Hello World!</td><td></td></tr>
<tr><td>open</td><td>./selenium_ft_tool/tearDown?user=${username}</td><td></td></tr>
<table>
```



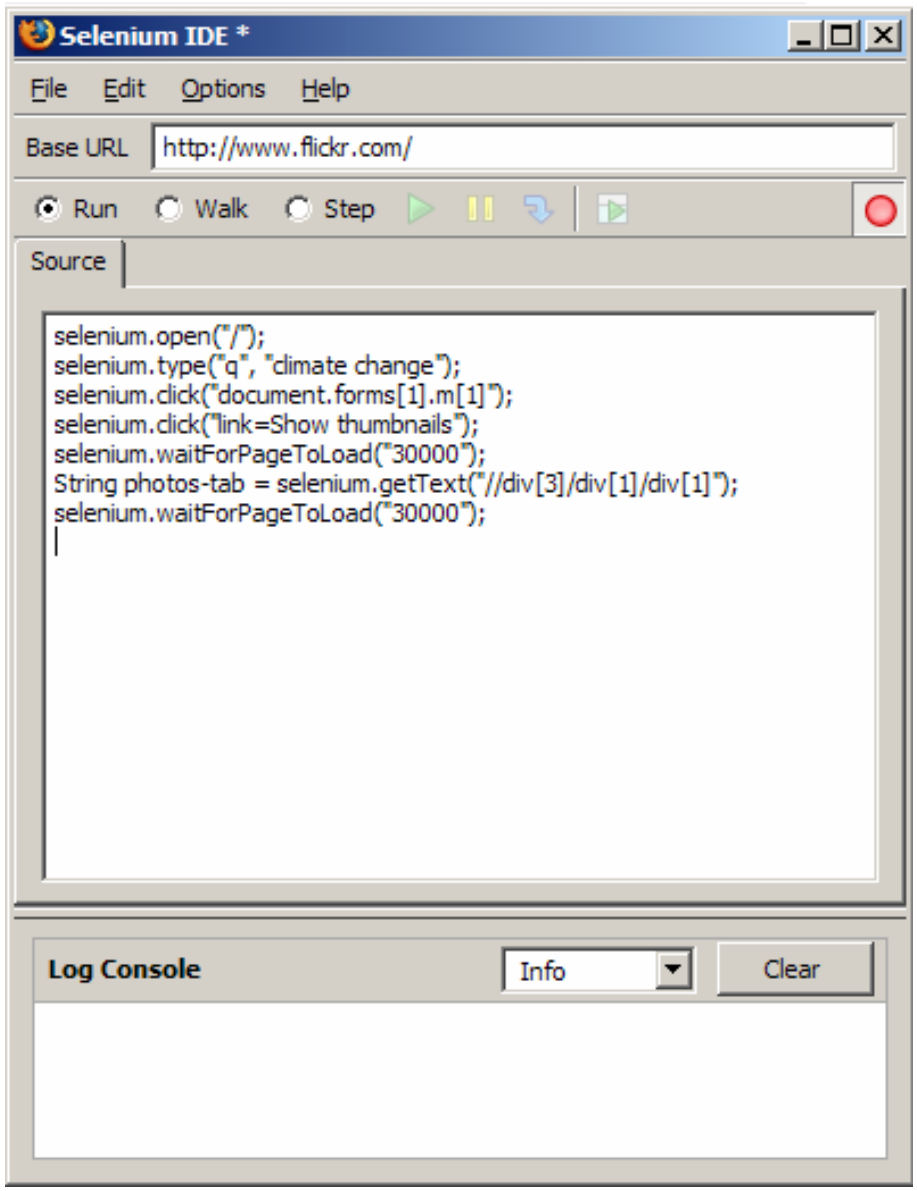
Functional Testing

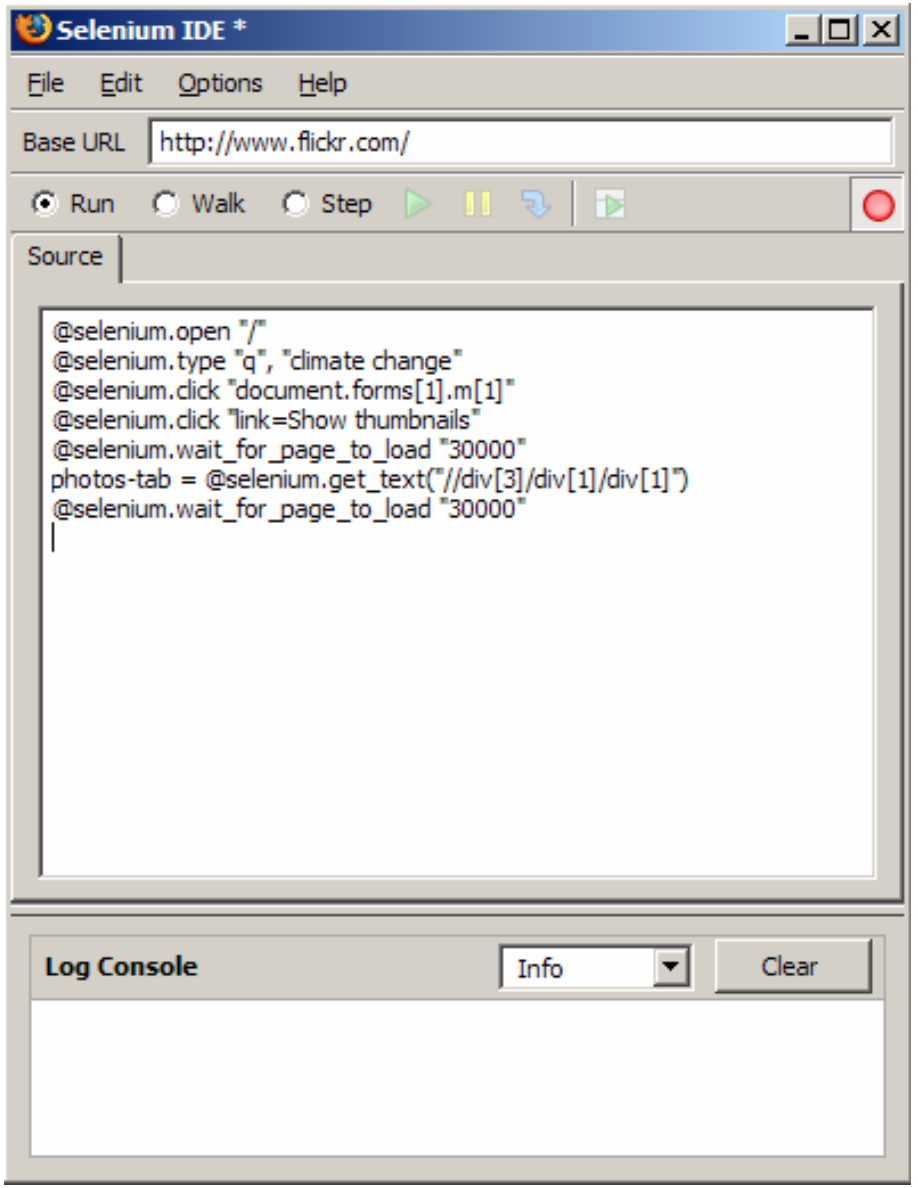
nitobi
built for people



Functional Testing

nitobi
built for people





Functional Testing

nitobi
built for people

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
    I. Unit Testing
    II. Functional Testing
    III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

Automated Testing

nitobi
built for people

Automated Testing

nitobi
built for people

Why Regression Testing?

- With many people working the same code regression bugs are inevitable
- This can be painful

Manual Testing

- Don’t underestimate its value
- Introduces randomness
- We are *good* at noticing oddities
- We are *bad* at checking precise results

Automated Testing

- This can be expensive
- The sooner you automate the greater value
- **Automated test value is in untargeted bugs it finds - completely unrelated to the specific purpose for which it was written**

Automate Testing with Care

- Some automation is easy (read cheap)
- That's good!
- But cheap automation is unlikely to reveal bugs
- That's bad!
- Automated tests can find regression bugs
- That's good!
- Automate tests can be fragile
- That's bad!
- Automated tests can be run with The Build
- That's good!
- Automated tests can ignore large areas of the software
- That's bad!

The Value Proposition

1. The more bugs you have the higher cost of automation
2. The more automation you have the fewer bugs you find
3. The more testing infrastructure you already have the better

Golden Rule

$$TC = \frac{NB \times AT}{TI}$$

Questions

- Do they test the right things?
- How many bugs do you know about?
- How long will tests be relevant?
- What is the value of the tests?

First Steps

- Have few automated tests that **coarsely** cover the software – Smoke Test
- If the smoke test succeeds, the product is testable and a “good build”
- Pluses
  - Configuration changes are easily observed
  - Gross regressions are caught preemptively
  - Keeps emphasis on manual testing

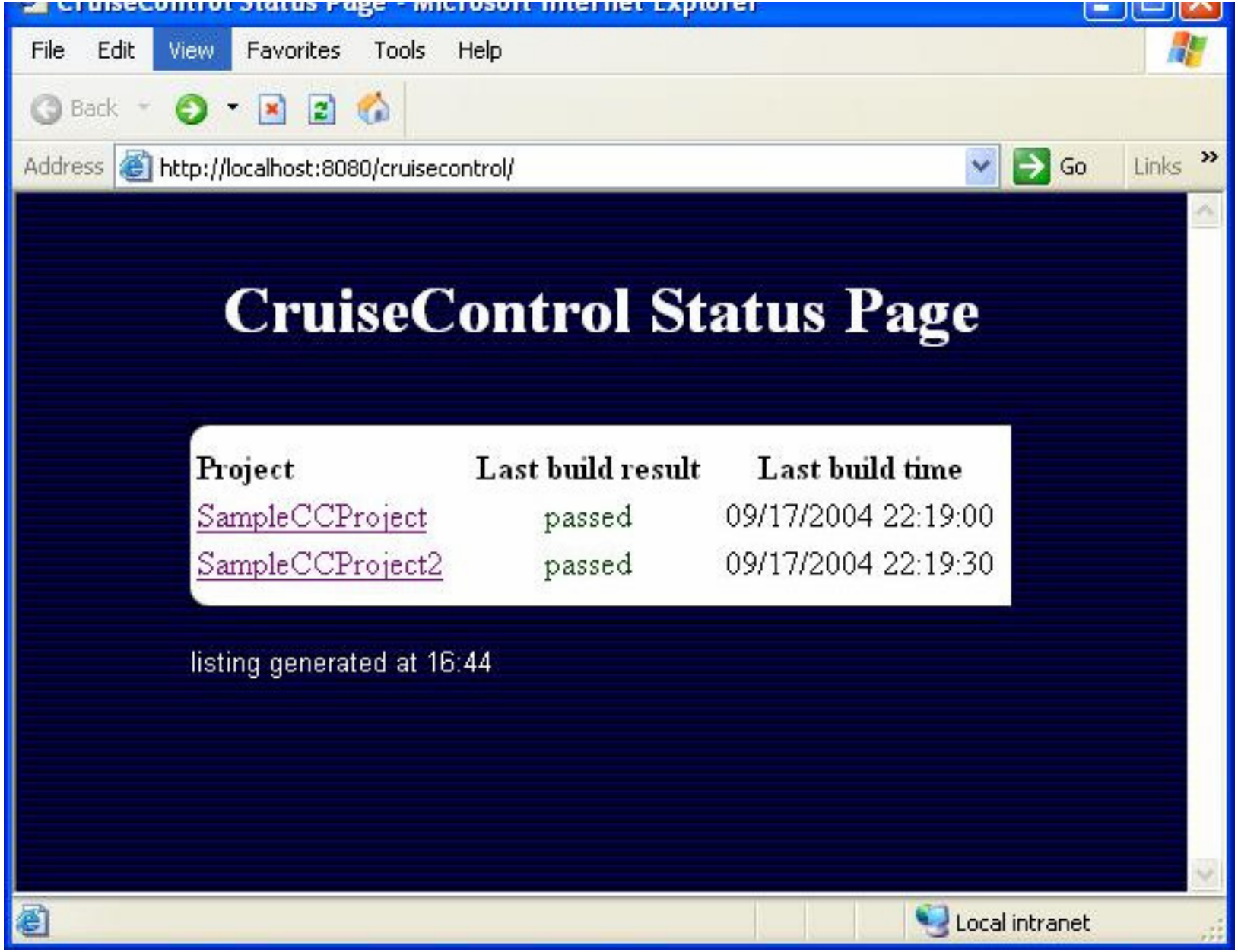
Automate Functional / Unit Tests

- Selenium server
- JSUnit server

Continuous Integration Tools

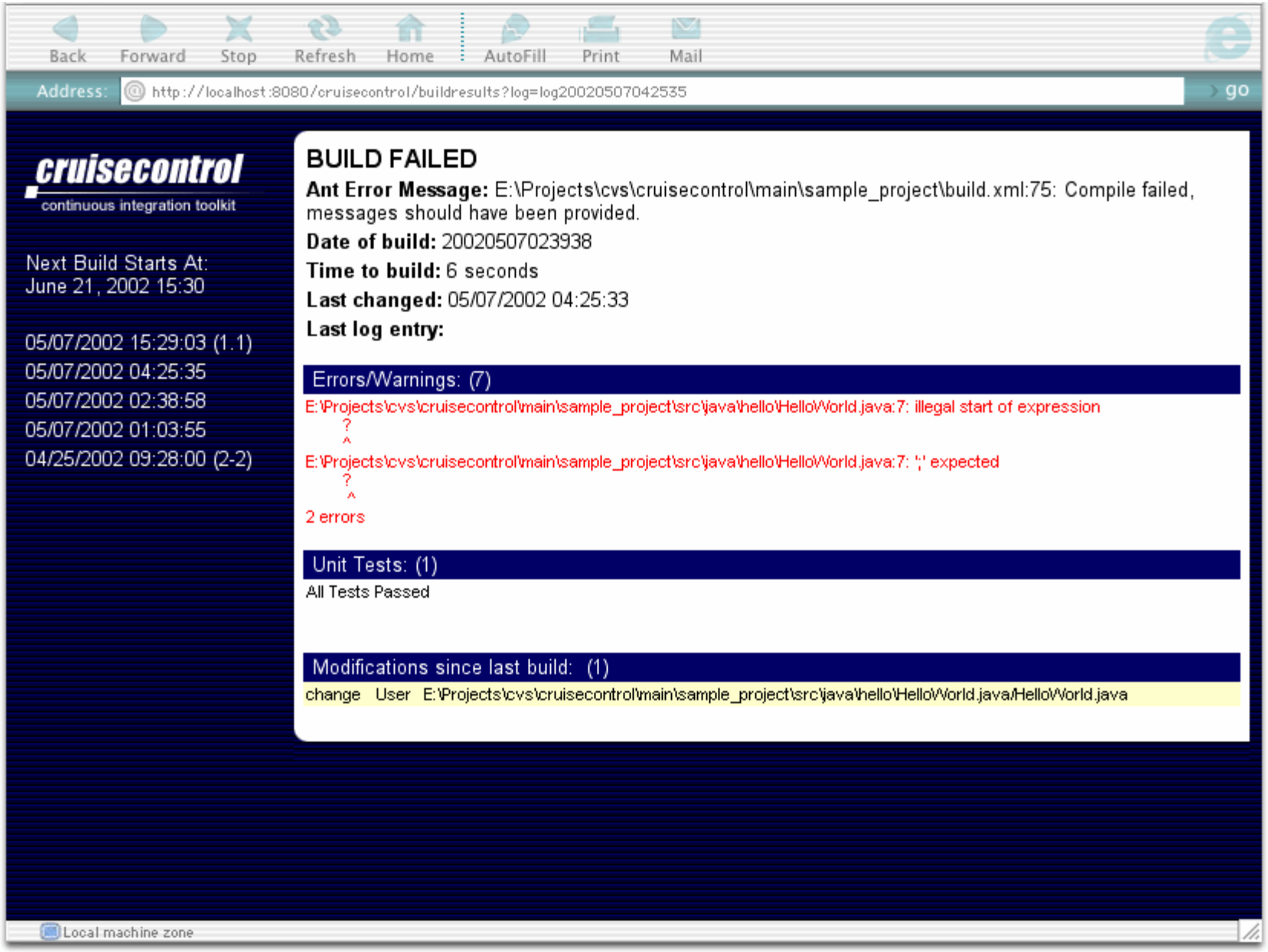
- CruiseControl
- AntHill
- Ant / NAnt

nitobi
built for people



Automated Testing

nitobi
built for people



Automated Testing

nitobi
built for people

Automated Testing

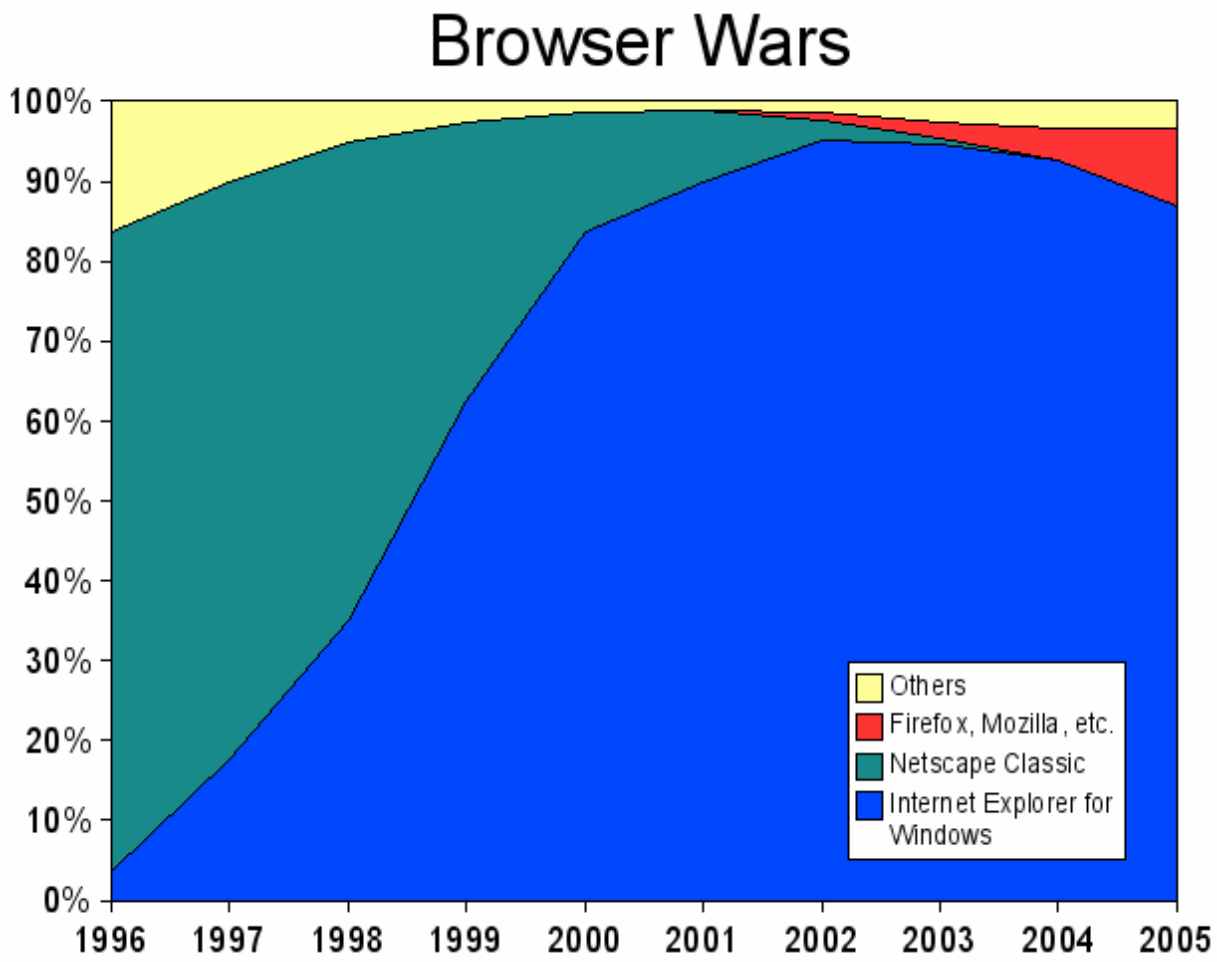
nitobi
built for people

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
   I. Unit Testing
   II. Functional Testing
   III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

Finally

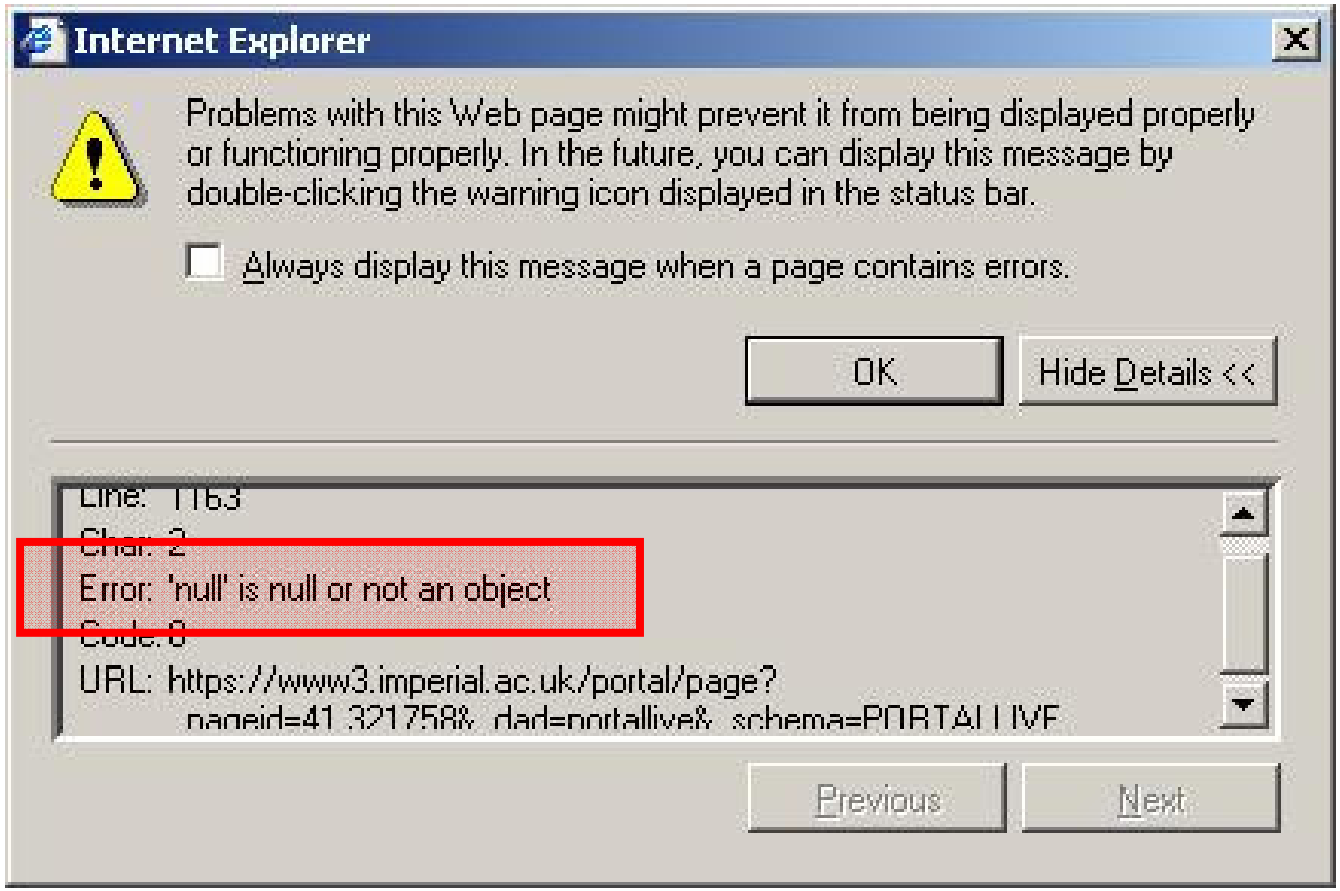
- Once the tests are written (and hopefully all fail) we can actually write some code.
- There's only one problem …
  creating bugs



nitobi
built for people

JavaScript Debugging

nitobi
built for people



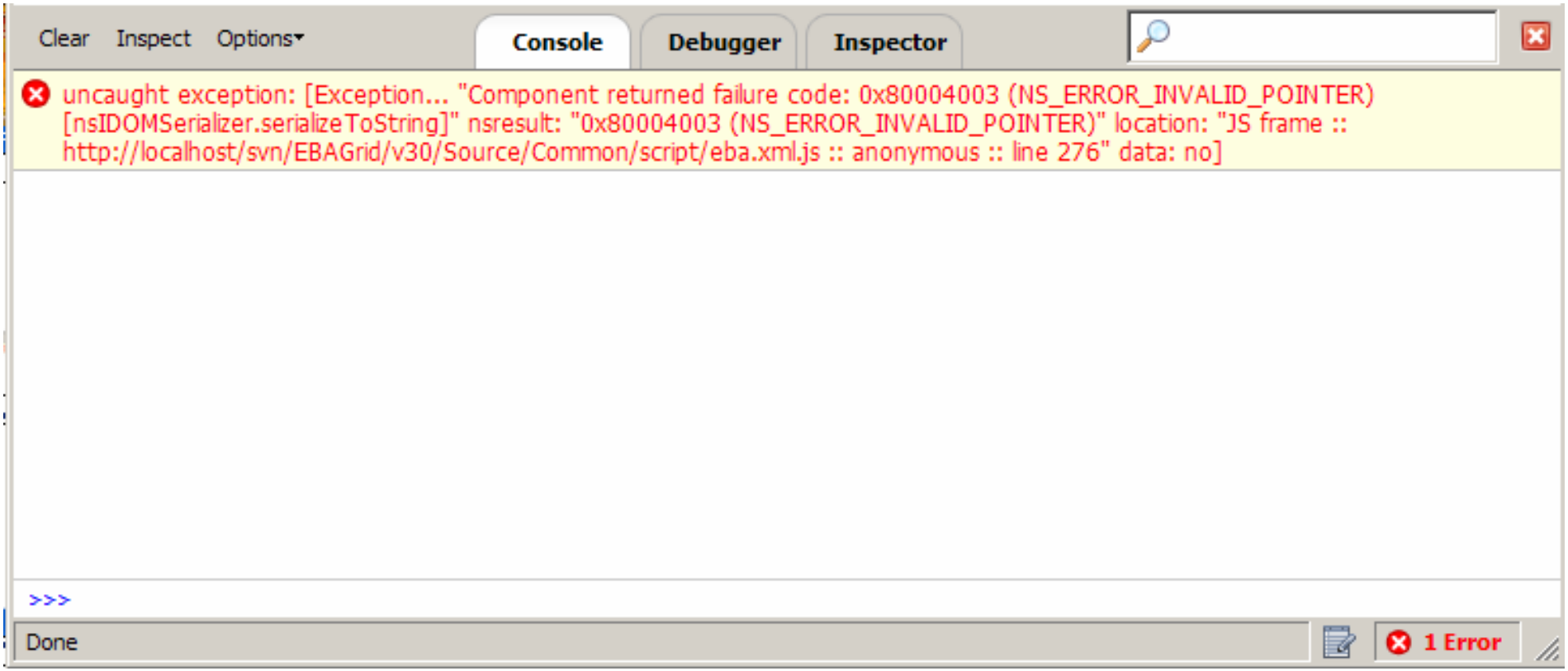
JavaScript Debugging

nitobi
built for people

http://www.flickr.com/photos/esther17/86558129/

JavaScript Debugging

nitobi
built for people



JavaScript Debugging

nitobi
built for people

Some Ground Rules

- Namespaces
  - nitobi.$
- Asynchronous programming
  - watch out for IE
- Inheritance, interfaces etc
- Readable names
  - “CustomerFirstNameWithMiddleInitial” or “a”
- Global variables
  - _this = this
- Closures
- Frameworks
- Documentation

Dealing with Errors

```
                team.pack();
                AppHelper.openModalDialog(t
                AddressHelper.getAddressBoo
                setList();
            }
        } catch (Exception e) {
        }
    }
}
```

http://www.flickr.com/photos/davesag/8519770/

Logging is your Friend

- FVLogger
- Aspect Oriented Programming

Tools

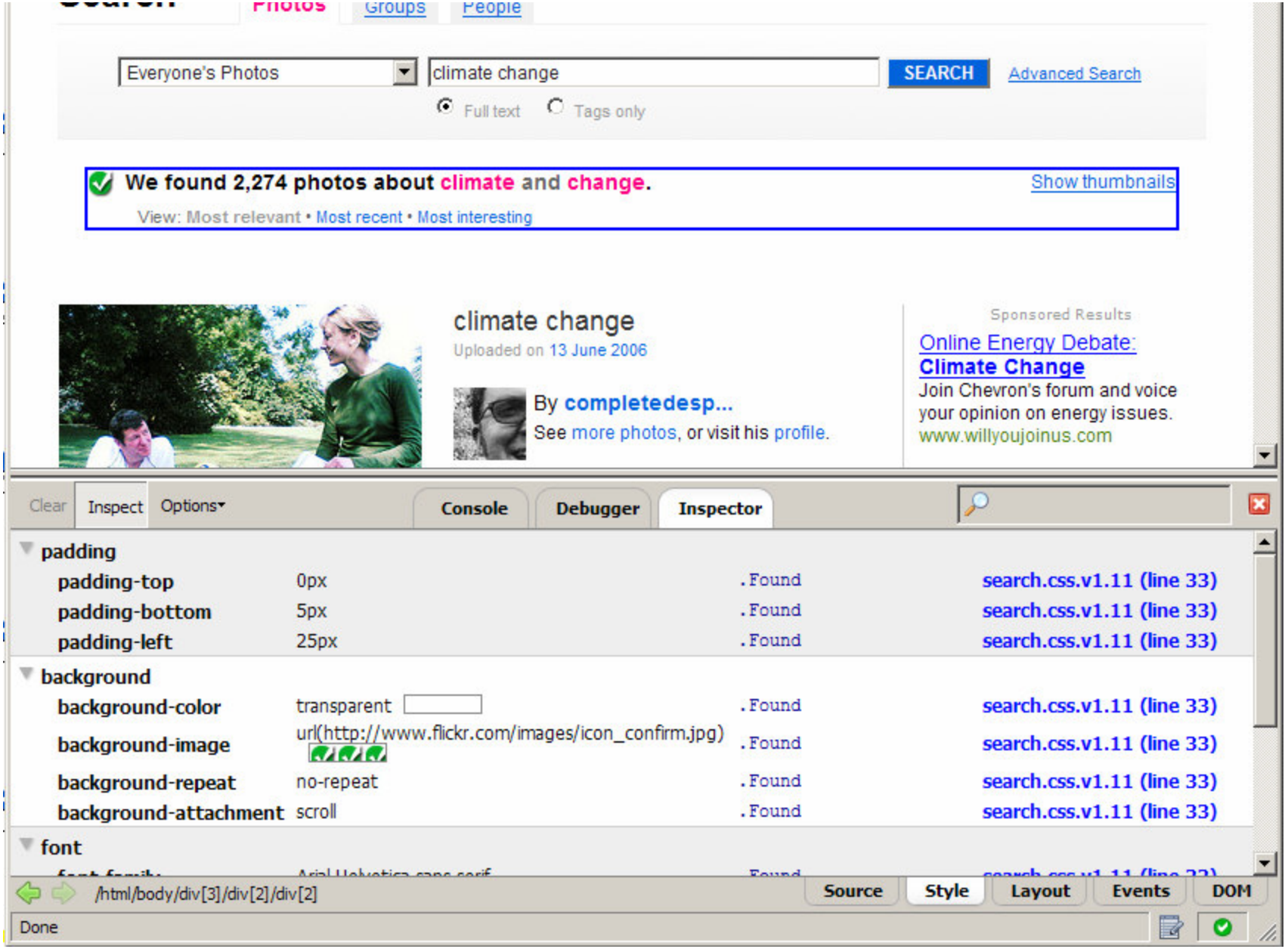
- Firebug (Mozilla)
- Developer Toolbar (IE & Mozilla)
- WebKit (Safari)

Firebug

- console.log('width: %d height: %d', w, h);
- Write arrays, elements, whatever
- Debug, info, warn, error
- console.trace
- console.time / timeEnd

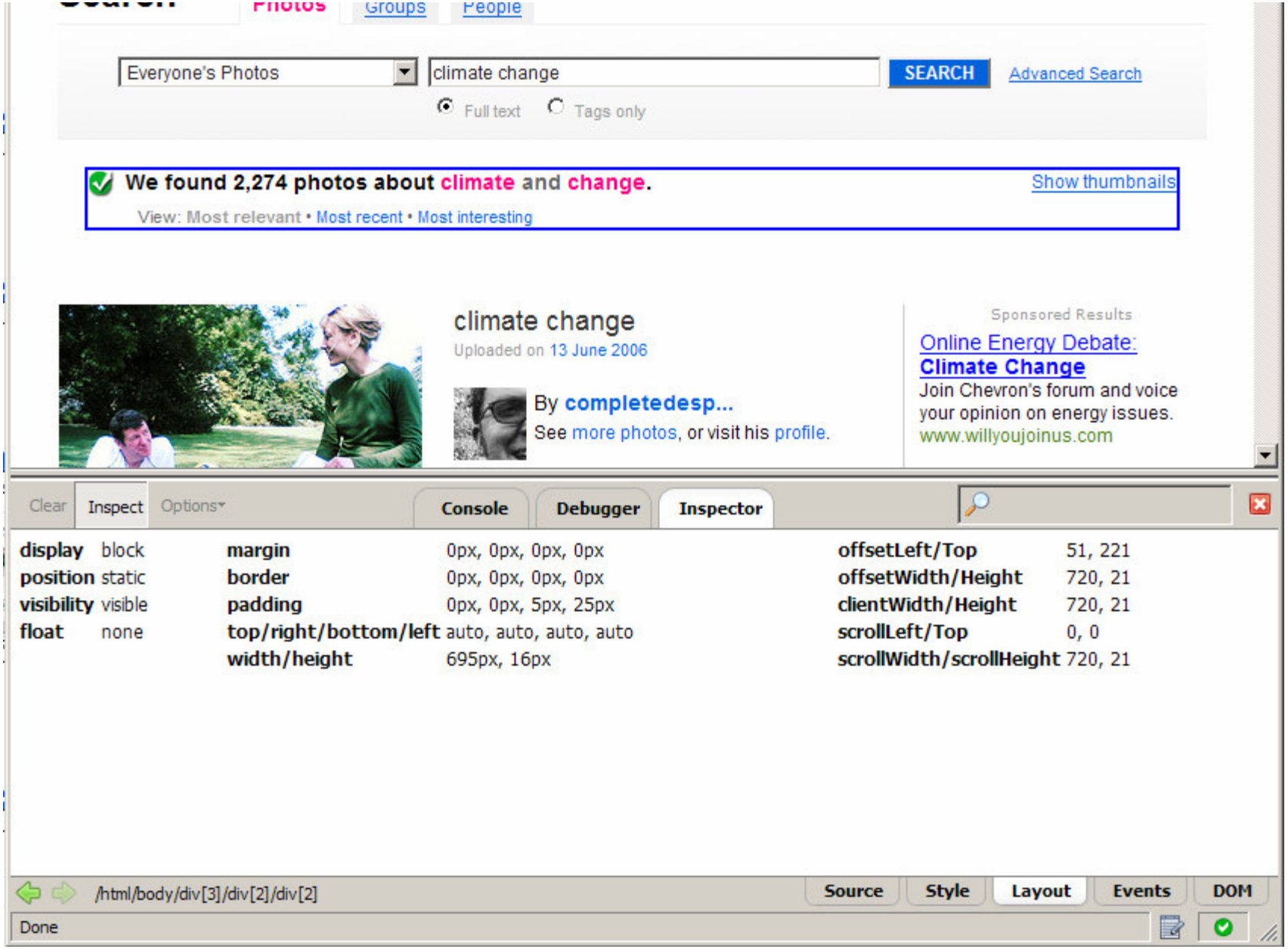
JavaScript Debugging

nitobi
built for people



JavaScript Debugging

nitobi
built for people



JavaScript Debugging

nitobi
built for people

JavaScript Debugging

nitobi
built for people

Debugger

- Most important JavaScript keyword?

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
   I. Unit Testing
   II. Functional Testing
   III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

http://www.flickr.com/photos/pulpolux/192587812/

Performance Optimization

nitobi
built for people

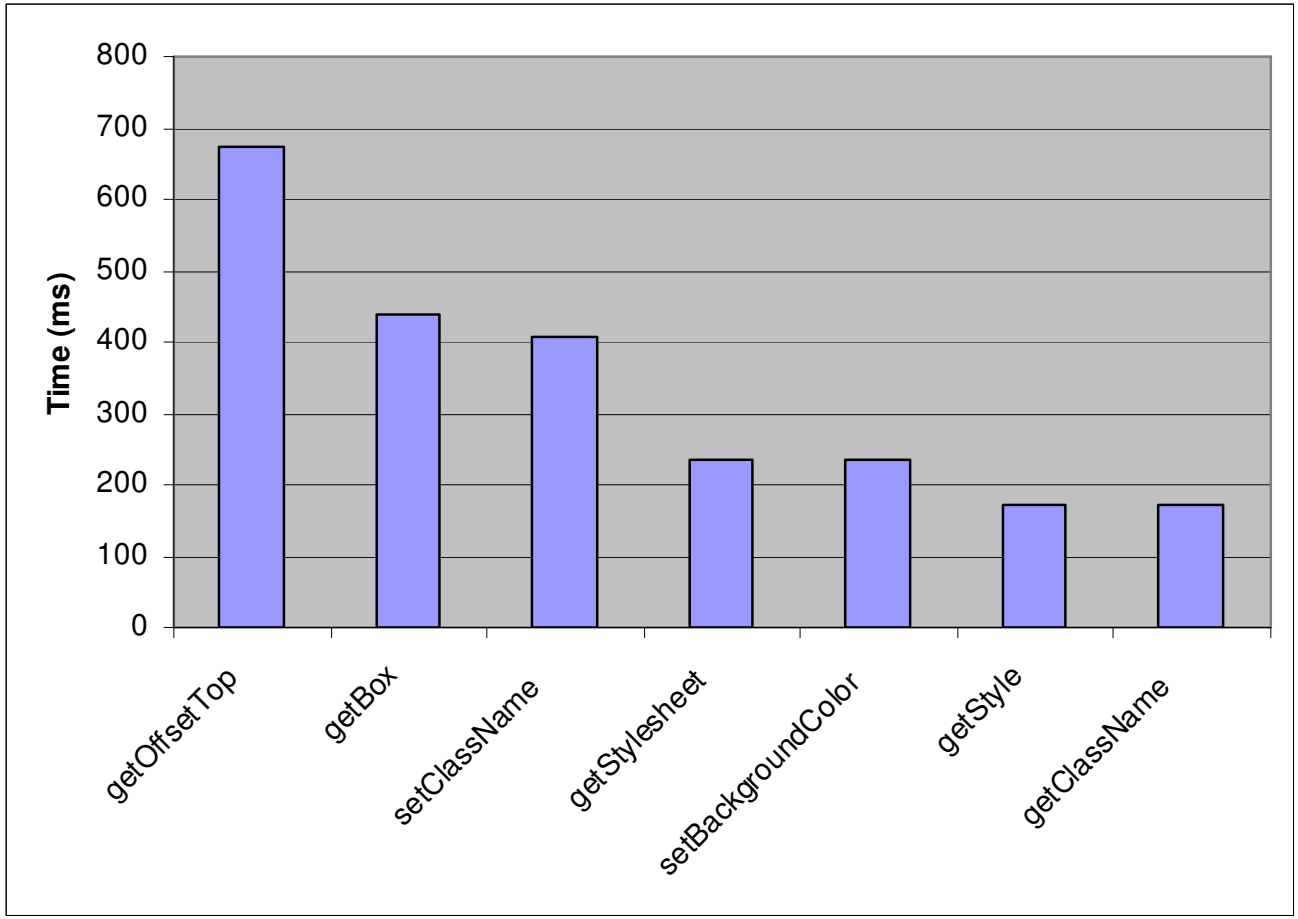
Golden Rule (#2)

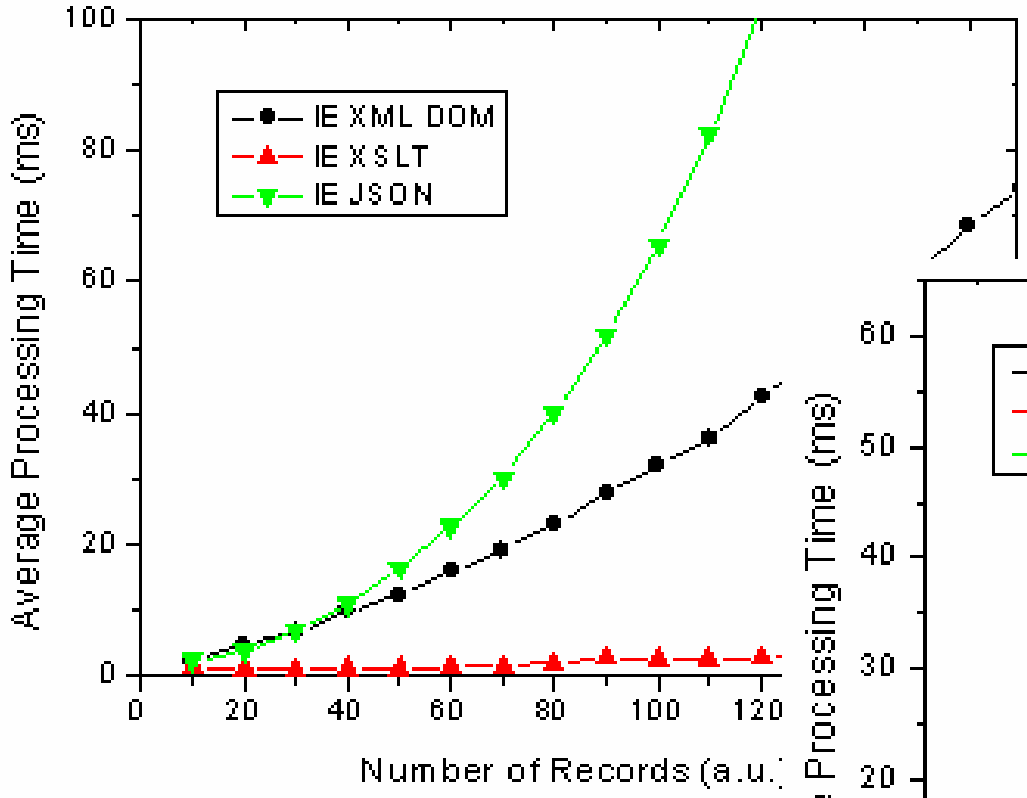
- Ask questions first optimize later

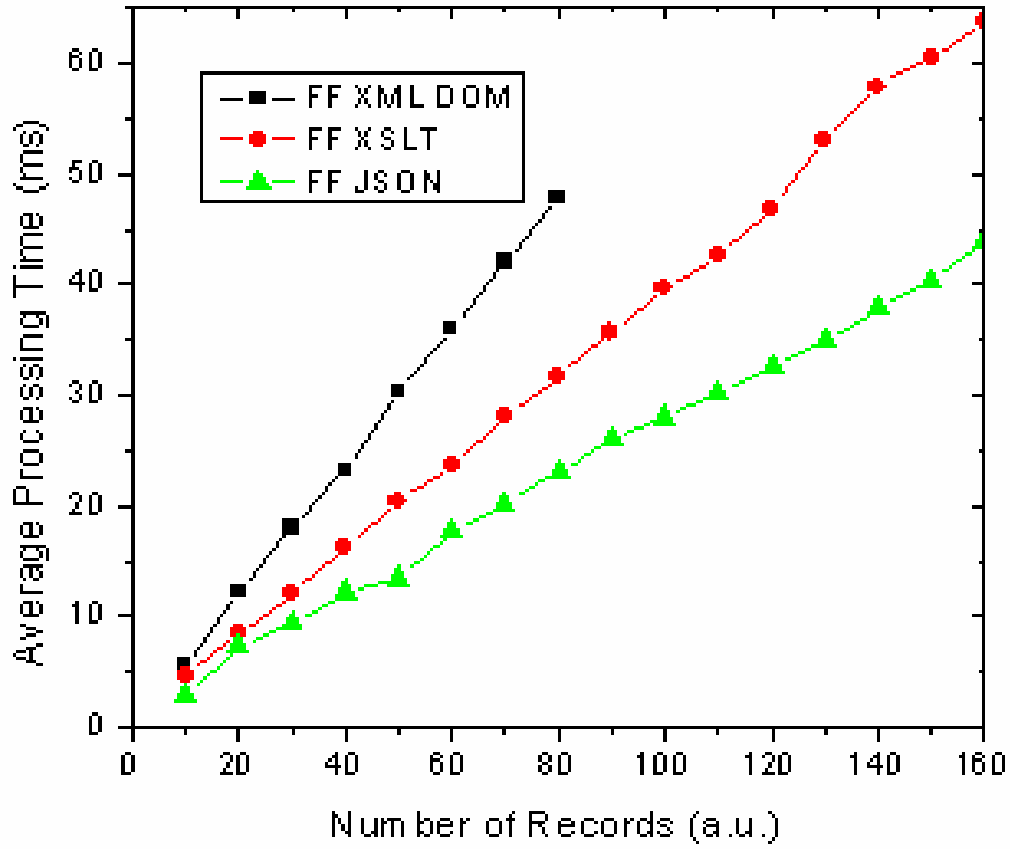
Optimization

- Situation dependent
- How much data?
- What type of data?
- How many server hits?
- What are the common workflows?

Big Picture


Time (ms)
800
700
600
500
400
300
200
100
0
getOffsetTop
getBox
setClassName
getStylesheet
setBackgroundColor
getStyle
getClassName





Performance Optimization

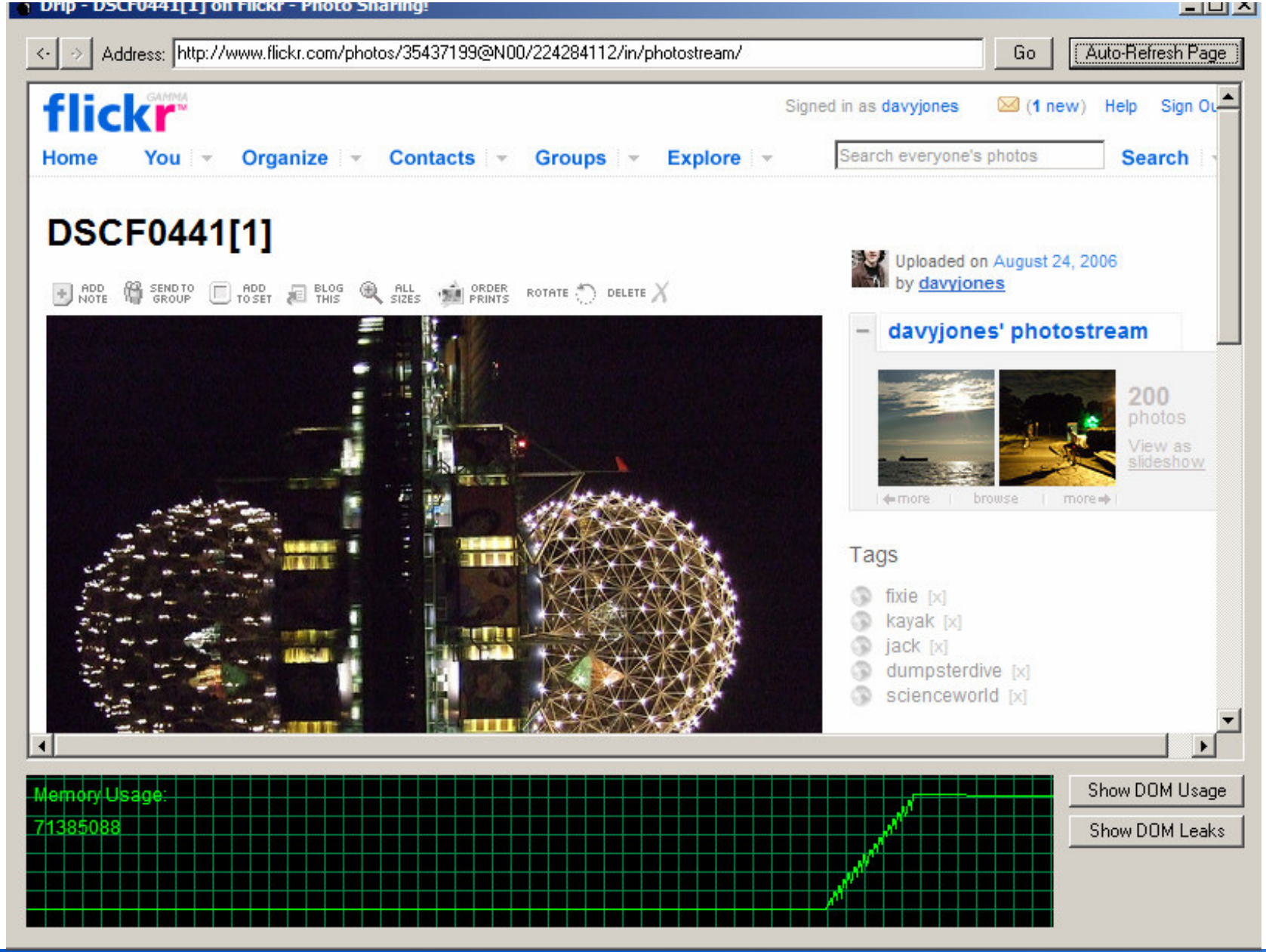
nitobi
built for people

Most Wanted

- innerHTML
- offsetTop / Left
- getBoundingClientRect / getBoxObjectFor
- Stylesheets

Tools

- Drip
- Logging / Debugging



Performance Optimization

nitobi
built for people

Overview

1. Who is this Guy? Who is nitobi?
2. Testing in Brief
   I. Unit Testing
   II. Functional Testing
   III. Regression Testing
3. JavaScript Debugging
4. Performance Optimization
5. Conclusions

Conclusions

- Assuring quality is an expensive task
- There ***is*** value in testing
- Don't test ***too*** much
- Use all the tools at your disposal
- Optimize with care

nitobi
built for people

Propaganda

- Talk to me for
  - Enterprise Ajax book flyers
  - Free Ajax component licenses