

Nachhaltiges Testen von Web-Applikationen mit dem Open Source Tool WebTest

Marc Guillemot
Dierk König



The Most Effective Way to Test
Your Web App!

Your speakers



- **Marc Guillemot**
 - Independent consultant
 - HtmlUnit & WebTest lead developer
 - Committer to Groovy & NekoHTML



- **Dierk König**
 - Software developer and architect at Canoo
 - WebTest founder
 - Committer to Groovy & Grails
 - Lead author of “Groovy in Action”



Agenda

- What is WebTest?
- A software engineering activity
- The four testing paradigms
- Conclusion



What is WebTest?

- automated Web functional testing tool
- open source (Apache license)
- founded in 2001
- currently 4 committers (Switzerland, Germany, Australia)
- used in banks, assurance, institutions, solution providers, ...
- adapted to a wide range of projects: from small to huge



Testing JAX website

ONLINE

Error

Message

JavaScript error loading page <http://it-republik.de/jaxenter/jax/>: ReferenceError: "v" is not defined. (http://it-republik.de/jaxenter/jax/scripts/js/functions.js#26)

Error Console

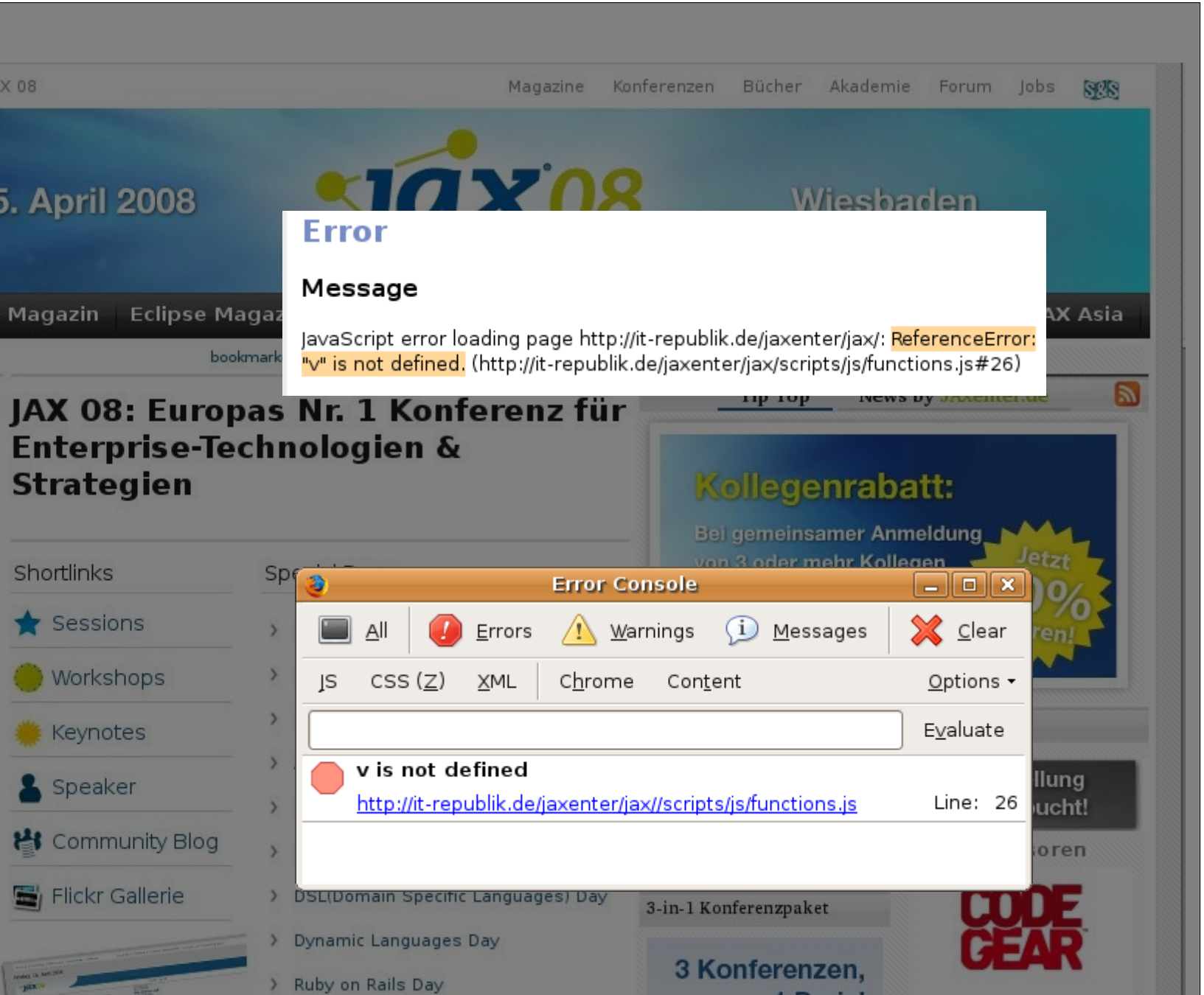
All Errors Warnings Messages Clear

JS CSS (Z) XML Chrome Content Options

Evaluate

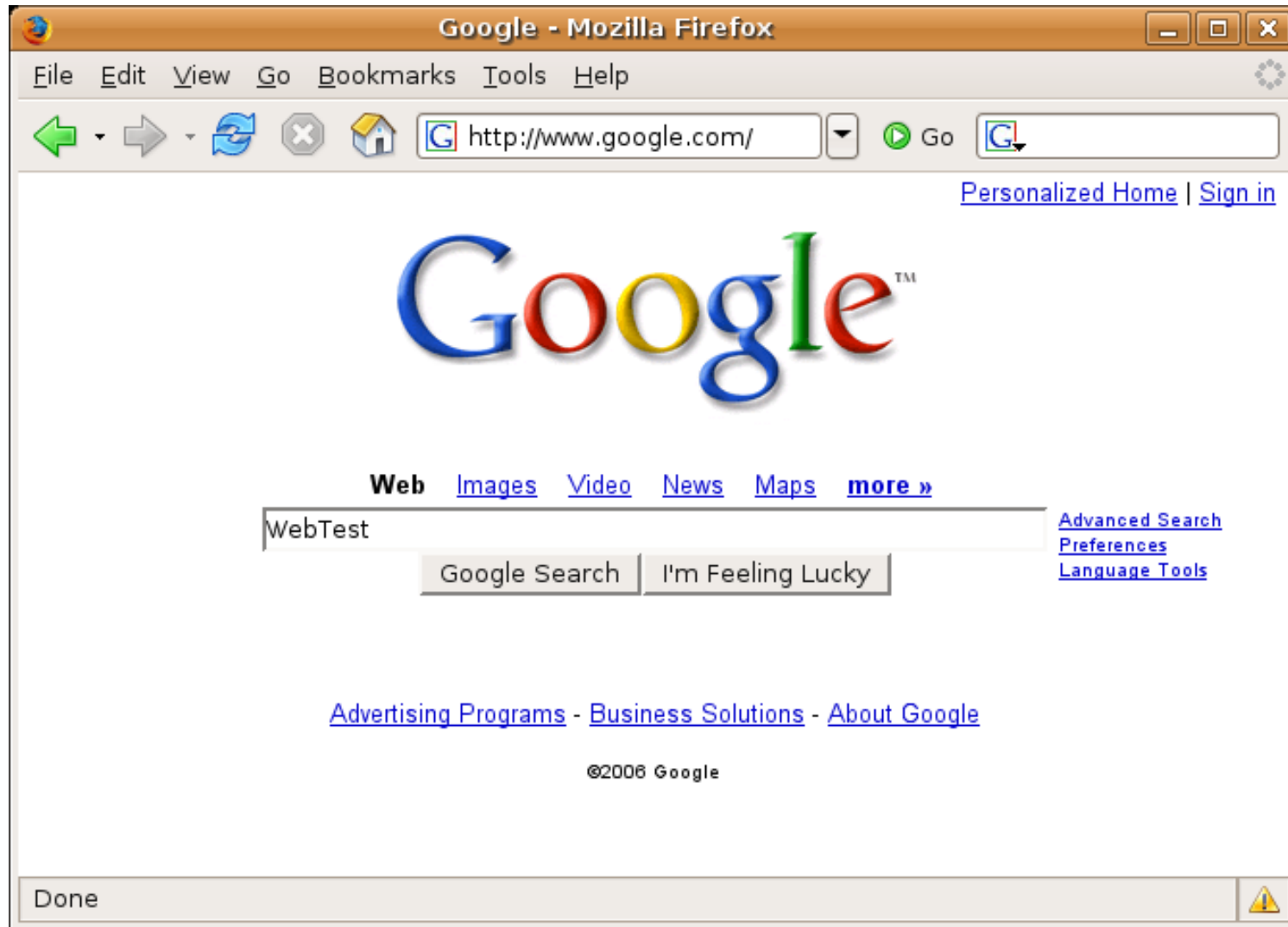
v is not defined

<http://it-republik.de/jaxenter/jax/scripts/js/functions.js> Line: 26



Testing Google

(DEMO)



WebTest Starter

google.xml

DEMO

```
<project default="test">
  <target name="test">
    <webtest name="check that WebTest is Google's top 'WebTest' result">
      <invoke url="http://www.google.com"/>
      <verifyTitle text="Google"/>
      <setInputField name="q" value="WebTest"/>
      <clickButton label="I'm Feeling Lucky"/>
      <verifyTitle text="Canoo WebTest"/>
    </webtest>
  </target>
</project>
```

```
# runWebtest google.xml
```





AllTests Tests started at Mon Oct 23 18:49:48 CEST 2006

Result Summary

Result	#	%	Graph
✓	5	100	
✗	0	0	
○	0	0	
Sum	5	100	

Server Roundtrip Timing Profile

Secs	#	%	Histogram
0 - 1	1	50	
1 - 5	1	50	
5 - 10	0	0	
10 - 30	0	0	
> 30	0	0	
Sum	2	100	
Avg		1	

Test Scenario Overview (1332)

No	Result	Name	# Steps	Duration	%	Timing profile	Graph	Failing step
1	✓	check that WebTest is Google's top 'WebTest' result	5 / 5	1332	100			

✓ check that WebTest is Google's top 'WebTest' result
 Test started at Mon Oct 23 18:49:48 CEST 2006, lasting 1332 ms.
 Source: /home/marc/fmr/OOPSLA2006/build.xml:4:
 Base URL (used by invoke steps with a relative URL): http://localhost/

No	Result	Name	Parameter	Duration
1	✓	invoke	method GET url http://www.google.com	1138
2	✓	verifyTitle	text Google	8
3	✓	setinputfield	name q value WebTest	3
4	✓	clickButton	label I'm Feeling Lucky	183
5	✓	verifyTitle	text Canoo WebTest Homepage	0

[Back to Test Report Overview](#)

(DEMO)

WebTest results (zoomed)

✓ check that WebTest is Google's top 'WebTest' result

Test started at Mon Oct 23 18:49:48 CEST 2006, lasting 1332 ms.

Source: /home/marc/fmr/OOPSLA2006/build.xml:4:

Base URL (used by **invoke** steps with a relative URL): <http://localhost/>

No	Result	Name	
1	✓	invoke	method GET
			url http://www.google.com
		Resulting page	
2	✓	verifyTitle	text Google
3	✓	setInputField	name q
			value WebTest
4	✓	clickButton	label I'm Feeling Lucky
		Resulting page	
5	✓	verifyTitle	text Canoo WebTest Homepage



Over 100 WebTest Steps

- General:
 - <invoke.../>
 - <clickLink.../>
 - ...
- Forms
 - <setInputField.../>
 - <setRadioButton.../>
 - <setCheckBox.../>
 - <clickButton.../>
 - ...
- Verification
 - <verifyTitle.../>
 - <verifyXPath.../>
 - <verifyInputField.../>
 - ...
- PDF
 - <pdfDecryptDocument.../>
 - <pdfVerifyField.../>
 - <pdfVerifyText.../>
 - ...
- Excel Documents
 - <excelFindRow.../>
 - <excelVerifyCellValue.../>
 - ...
- ...

Comprehensive online documentation:
<http://webtest.canoo.com/>



A software engineering activity



Marc Guillemot & Dierk König
JAX 08, 24.04.2008

webtest

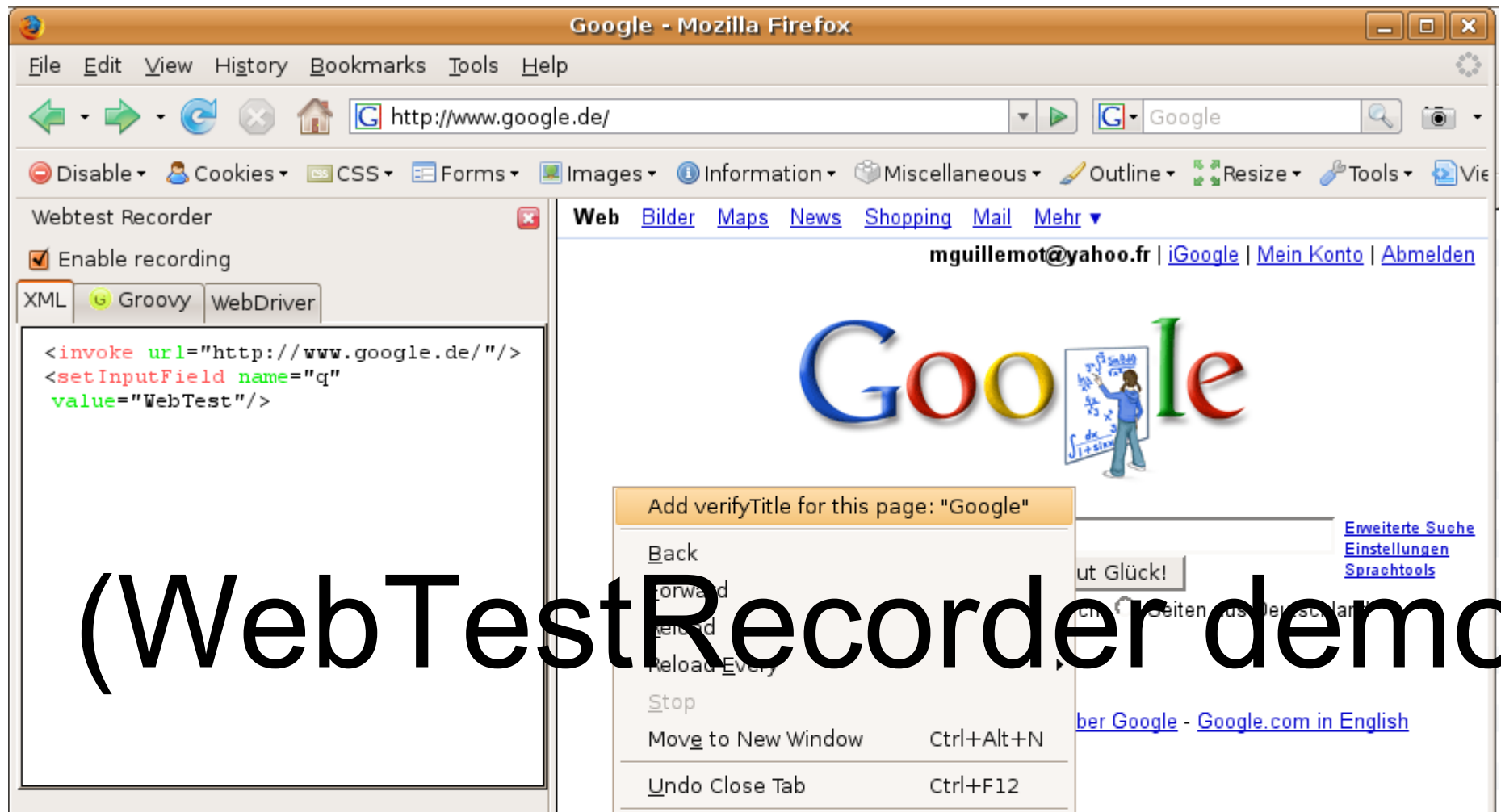
The 4 testing paradigms

- **Capture / replay**
 - "the least cost-effective way of test automation"
 - Use capturing wisely
- **Model based testing**
 - Specification: modelling expected behaviour
 - Fault model: modelling unexpected behaviour
 - Be tolerant to accidental behaviour
- **Data driven testing**
 - Narrow scope for data variations on same workflow
- **Scripted automation**
 - Maximum flexibility and power
 - Maximum responsibility



Capture / replay

- "the least cost-effective way of test automation"
- Use capturing wisely



The screenshot shows a Mozilla Firefox browser window with the URL `http://www.google.de/`. The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The address bar shows the URL and search engines. The main content area displays the Google homepage with the logo and navigation links. A WebTest Recorder extension is active, showing a recording status of "Enable recording" and a code editor with the following XML snippet:

```
<invoke url="http://www.google.de/" />
<setInputField name="q"
value="WebTest" />
```

A context menu is open over the Google logo, listing actions such as Back, Forward, Reload, Stop, Move to New Window, and Undo Close Tab. A large text overlay at the bottom of the image reads "(WebTestRecorder demo)".

Model based testing

- Specification: modelling expected behaviour
 - good XPath: `//*[@id='total']`
 - bad XPath: `/html/body/div[2]/table[3]/tbody/tr[6]/td[4]`
- Be tolerant to accidental behaviour
- Fault model: modelling unexpected behaviour

```
<not>  
  <verifyText text="NullPointerException"/>  
</not>
```



Determine what you want to test

How to test addition into basket:

Top Angebot!



2x Ajax Allzweckreiniger
Frischeduft - pH neutral

2x 1 l

~~8.60~~
6.60

Fr. 3.30/ l



Einkaufskorb

- `<clickLink label="Ajax"/> ?`
- `<clickLink xpath="//*[text() = 'Top Angebot']//a"/> ?`
- `<clickLink xpath="//*[text() = '6.60']/following-sibling::a"/> ?`

=> it depends: these 4 examples don't test the same thing!



Data driven testing

- Narrow scope for data variations on same workflow



Microsoft Excel - GoogleCalcTestData.xls

Type a question for help

File Edit View Insert Format Tools Data Window Help

100%

10 B / abc

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Expression	Result											
2	3 + 5	8											
3	4 * 13	52											
4	48 / 3	16											
5	12 !	479 001 600											
6	49.7 - 19.9	29.8											
7	sqrt(144)												
8	12^3												
9	i * i												
10	(3 * i) - (4 * i)												
11	sqrt(-4)												
12	9 modulo 2												
13	18 choose 5												
14	5th root of 3												
15	cos(pi / 3)												
16	ln(567)												
17	ln(e)												
18	sin(30 degra												
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													

```

<project default="test">
  <target name="test">
    <dataDriven tableContainer="GoogleCalcTestData.xls">
      <webtest name="verify ${Expression} = ${Result} with Google">
        <invoke url="http://www.google.com/ncr"/>
        <verifyTitle text="Google"/>
        <setInputField name="q" value="${Expression}"/>
        <clickButton label="Google Search"/>
        <verifyXPath xpath="//tr[.//img[@src='/images/calc_img.gif']]/td[3]//b"
          text="${Expression} = ${Result}"/>
      </webtest>
    </dataDriven>
  </target>
</project>
  
```

=> see screencast "Data Driven WebTest" for details

Scripted automation

- Maximum flexibility and power
- Maximum responsibility



WebTest Starter as Groovy code

```
class GoogleWebTest extends grails.util.WebTest {  
  
    def testSearchForWebTest() {  
        webtest("check that WebTest is Google's top 'WebTest' result") {  
            invoke "http://www.google.com"  
            verifyTitle "Google"  
            setInputField name: "q", value: "WebTest"  
            clickButton "I'm Feeling Lucky"  
            verifyTitle "Canoo WebTest"  
        }  
    }  
  
    def suite() {  
        testSearchForWebTest()  
    }  
}
```



Use HtmlUnit API

First name	Last name	Location	Since
Dierk	König	Switzerland	2001
Denis	Antonioli	Switzerland	2002
Marc	Guillemot	Germany	2003
Paul	King	Australia	2004

```
...
<webtest>
.....
  <groovy description="test table sorted by last name">
import com.canoo.webtest.engine.StepFailedException as SFE

def table = step.context.currentResponse.getHtmlElementById('theTable')
def tds = table.getByXPath('tbody/tr/td')
def texts = tds*.asText()
def sorted = new ArrayList(texts).sort()
if (sorted != texts)
  throw new SFE("Not correctly sorted",
  </groovy>
.....
</webtest>
...
```

Error

Message

Not correctly sorted

Location

(line: 0)

Details

expected value ["Antonioli", "Guillemot", "King", "König"]

actual value ["König", "Antonioli", "Guillemot", "King"]



Simply extend WebTest

```
<groovyScript>
class MyExtension extends com.canoo.webtest.steps.Steps
{
    String myProp
    void doExecute()
    {
        // do something
    }
}
project.taskDefinitions['myNewStep'] = MyExtension
</groovyScript>
....
<webtest>
...
<myNewStep myProp="whatever"/>
...
</webtest>
```



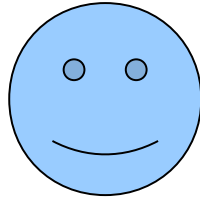
Lessons learned

- use recorder only to give you a jump start
- specify expectations
- distinguish guaranteed from accidental behaviour
- apply rules of software engineering
 - remove duplications, extract modules
 - use source control
- consider modifications and extensions
 - Ant, Java, Groovy
- design for testability
 - valid html code, test early

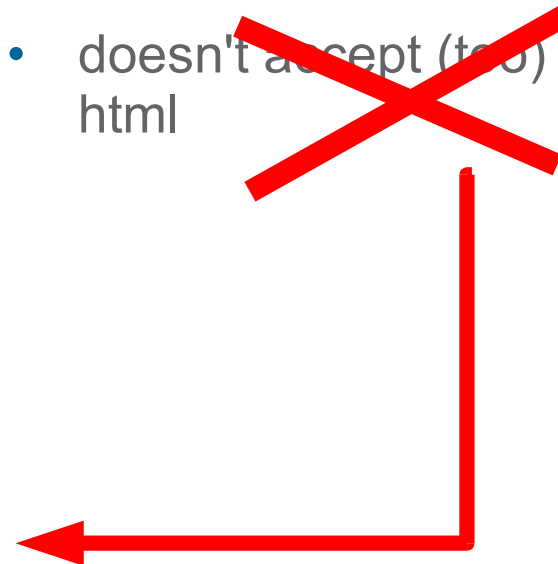


Key properties

- simple
- fast
- excellent reporting
- very low TCO
- runs everywhere
- no display needed
- easy to extend
- straightforward integration
- doesn't accept (too) badly formed html



- js support not as good as in "normal" browser
- doesn't accept (too) badly formed html



WebTest (possible) future(s)

- WebTestClipse
- “Functional load testing”
- Drive widespread browsers (demo)
- ...





The Most Effective Way to Test
Your Web App!

Happy Testing!

<http://webtest.canoo.com/>



Marc Guillemot & Dierk König
JAX 08, 24.04.2008

webtest

Bonus



Marc Guillemot & Dierk König
JAX 08, 24.04.2008

webtest

User testimonials

(from WebTest mailing list)

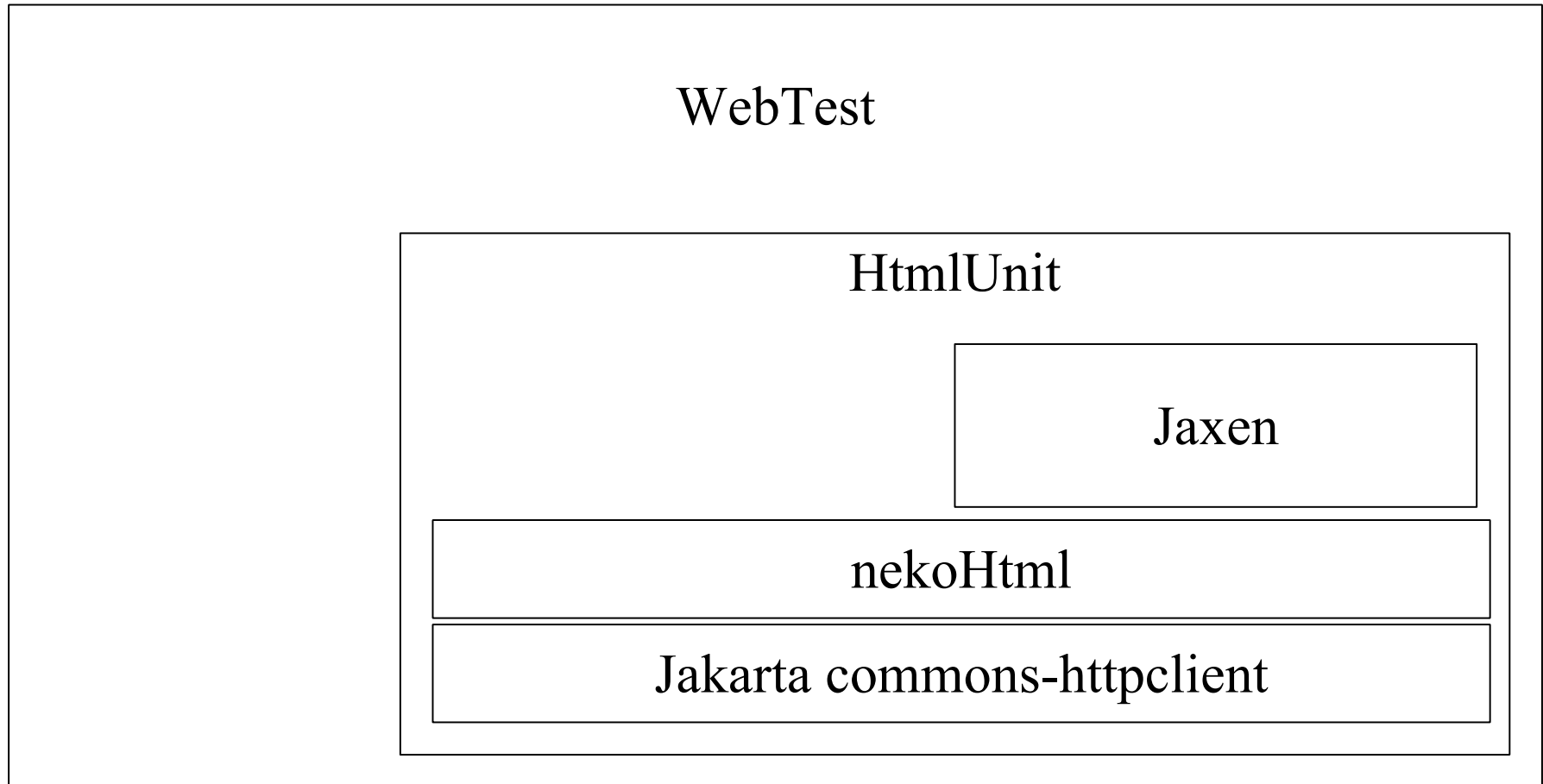
- The ROI on WebTest is many orders of magnitude higher than any tool I've used
- Support = good and \$0
- It still Just Works. It has been remarkably robust, mature, and bug-free
- It has scaled well
- Tests are much quicker to write and cheap to maintain
- Excellent reporting. Forget all the other reasons!



Marc Guillemot & Dierk König
JAX 08, 24.04.2008

webtest

Building blocks



Sustainable Tests

- testable HTML code
- structured tests
 - property files
 - XML entities
 - Ant macros



Use properties wisely

```
<project default="test">
  <target name="test">

    <webtest name="check that WebTest is
    Google's top 'WebTest' result">
      <invoke url="{startUrl}"/>
      <verifyTitle text="Google"/>
      <setInputField name="q" value="WebTest"/>
      <clickButton label="{luckyButton}"/>
      <verifyTitle text="Canoo WebTest
      Homepage"/>
    </webtest>

  </target>
</project>
```

de.properties

```
startUrl=http://www.google.de/de
luckyButton=Auf gut Glück!
```

fr.properties

```
startUrl=http://www.google.fr/fr
luckyButton=J'ai de la chance
```

en.properties

```
startUrl=http://www.google.com/ncr
luckyButton=I'm Feeling Lucky
```

```
...
<ant antfile="google.xml">
  <property file="de.properties">
</ant>
....
```



Use entities for common blocks

myTest.xml

```
<!DOCTYPE project SYSTEM
  "myDtd.dtd">
<project>
  ....
  <webtest name="my test">
    &goToLoginPage;
    ...
  </webtest>
  ...

```

myDtd.dtd

```
<!ENTITY goToLoginPage SYSTEM
  "includes/goToLoginPage.xml">

```

goToLoginPage.xml

```
<group description="go to login">
  <invoke url="http://mysite.com"/>
  <verifyTitle text="My great Web Site"/>
  <clickLink label="login"/>
  <verifyText text="Restricted area"/>
</group>

```



... or macros

```
...  
<macrodef name="doLogin"/>  
  <attribute name="login"/>  
  <attribute name="password"/>  
  <sequential>  
    &goToLoginPage;  
    <setInputField forLabel="Login" value="@{login}"/>  
    <setInputField forLabel="Password" value="@{password}"/>  
    <verifyText text="Hello .*\. Welcome to the restricted area" regex="true"/>  
  </sequential>  
</macrodef>  
...
```

myTest.xml

```
<project>  
  ....  
  <webtest name="simpe test">  
    <doLogin login="john" password="john"/>  
    ...  
  </webtest>  
  ...
```



AJAX

```
...  
<webtest name="Test DWR 2.0 RC3 demo App">  
  <config easyajax="true"/>  
  <invoke url="http://localhost:8080/dwr/simpletext"/>  
  <verifyXPath xpath="//*[@id= 'demoReply']" text=""/>  
  <clickButton label="Send"/>  
  <verifyXPath xpath="//*[@id= 'demoReply']"  
    text="Hello, Joe"/>  
</webtest>  
...
```

#	Result	Name	Parameter
1	✓	invoke Resulting page	url http://localhost:8080/dwr/simpletext
2	✓	verifyXPath text	xpath //*[@id= 'demoReply']
3	✓	clickButton label	Send
4	✓	sleep Wait for completion of async call	seconds 2
5	✓	verifyXPath text	Hello, Joe xpath //*[@id= 'demoReply']

easyajax setting experimental

- need to add `<sleep.../>`



Fine control

```
...  
<webtest>  
  <groovy description="configure HTTP 1.0 as default protocol version">  
    import org.apache.commons.httpclient.*  
    import org.apache.commons.httpclient.params.*  
  
    DefaultHttpParams.defaultParams.version = HttpVersion.HTTP_1_0  
  </groovy>  
  <invoke url="http://myHost"/>  
    .....  
</webtest>  
...
```



Customize WebTest

Home > Functional testing > WebTest

```
...  
<verifyXPath xpath="//*[@id='navPath1']" text="Home"/>  
<verifyXPath xpath="//*[@id='navPath2']" text="Functional testing"/>  
<verifyXPath xpath="//*[@id='navPath3']" text="WebTest"/>  
...
```

Works but:

- too low level
- not reusable



Customize WebTest

Home > Functional testing > WebTest

```
...  
<macrodef name="verifyNavPath"/>  
  <attribute name="level1"/>  
  <attribute name="level2"/>  
  <attribute name="level3"/>  
  <sequential>  
    <verifyXPath xpath="//*[@id='navPath1']" text="@{level1}"/>  
    <verifyXPath xpath="//*[@id='navPath2']" text="@{level2}"/>  
    <verifyXPath xpath="//*[@id='navPath3']" text="@{level3}"/>  
  </sequential>  
</macrodef>  
...  
<verifyNavPath level1="Home" level2="Functional testing"  
  level3="WebTest"/>  
...
```



Customize WebTest

```
<groovyScript name="verifyNavPath"><![CDATA[
class VerifyNavigationPath extends com.canoo.webtest.steps.Step {
    String level1, level2, level3, level4
    void doExecute() {
        def ant = new AntBuilder(project)
        def levels = [0, level1, level2, level3, level4]
        for (i in 1..<levels.size()) {
            if (levels[i])
                ant.verifyXPath(xpath: "//a[@id='navPath${i}']/text()", text: levels[i],
                    description: "Verify level ${i}")
        } } }
    project.addTaskDefinition('verifyNavPath', VerifyNavigationPath)
]}></groovyScript>
...
<verifyNavPath level1="Home"/>
<verifyNavPath level1="Home" level2="Functional testing" level3="WebTest"/>
```

