

AUTOMATED SOFTWARE TESTING

Automated
Testing Institute
testing • techniques • tools

MAGAZINE

JANUARY 2014 \$10.95

BDD, LABS AND THE RISING

Testing The Rest
RESTFUL INTERFACE
TESTING AND ITS
CHALLENGES

BDD Before the
Tools

THINGS TO FOCUS ON
PRIOR TO HEADING DOWN
THE TOOL PATH

Automated Test
Lab in the Cloud
AN OPEN SOURCE AND
AUTOMATED TEST LAB
SOLUTION

→ MONEY



AUTOMATING THE REST: Automating a RESTful Web Service in 5 Steps

TESTKIT CONFERENCE 2014

Testing & Test Automation Conference

TestKIT

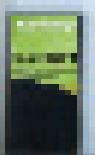
Test Automation
Conference

The D



TestKIT

TESTING & AUTOMATION CONFERENCE



Save the Date!

MARCH 3 - 5, 2014
CROWNE PLAZA HOTEL
ARLINGTON, VA

The KIT is Komming... Back

www.testkitconference.com

AUTOMATED SOFTWARE TESTING

January 2014, Volume 6, Issue 1



CONTENTS

BDD, LABS AND THE REST

“Give me six hours to chop down a tree and I will spend the first four sharpening the axe.” Lincoln obviously understood the importance of preparation when he made this quote. When you are appropriately prepared for a job, that job will go so much more smoothly than if you weren’t prepared. This issue of the magazine focuses on this concept by looking at preparation for Behavior Driven Development (BDD), automation lab creation, and REpresentational State Transfer (REST) testing.

FEATURES

TESTING THE REST 12

This article addresses RESTful interface testing and its challenges. *By Tom Wimsatt*

AN AUTOMATED TEST LAB IN THE CLOUD 18

This article discusses the importance of a well set up cloud-based automated test lab, and offers a potential solution that meets this need. *By Mukesh Sharma*

BEHAVIOR DRIVEN DEVELOPMENT BEFORE THE TOOLS 24

This article discusses the importance of performing an adequate level of preparation prior to the implementation of tools when implementing Behavior Driven Development (BDD). *By Liz Keogh*

COLUMNS & DEPARTMENTS

EDITORIAL 4

Armed and Ready for Battle

Preparing to battle bugs

AUTHORS AND EVENTS 6

Learn about AST authors and upcoming events.

TESTKIT TIP 8

Template For Automating the REST

Automating a RESTful Web Service with JUnit and Spring's Rest Template in 5 Steps

OPEN SOURCERY 19

Best Automation Fruit

Best ‘New’ Tools

LOCAL CHAPTER NEWS 28

Need the Latest Local Chapter Happenings?

I 'B'LOG TO U 36

[Read featured blog posts from the web](#)

The Automated Software Testing (AST) Magazine is an
Automated Testing Institute (ATI) publication.
For more information regarding the magazine visit
<http://www.astmagazine.automatedtestinginstitute.com>

Armed and Ready for Battle

by Dion Johnson

Don't you hate it when sports figures use war analogies to describe the games that they play? If so, then you may want to skip this editorial. Before you do, though, take into consideration that the analogy that sports figures use describes them running back and forth with a ball, puck, or some other similar object. I'm about to make an analogy that speaks to software testing in war-like terms, and although sports are typically more physical than software testing, I assert that my analogy is a little more appropriate. The reason? Software is often used to aid armed forces and war efforts, along with battle terrorist and other bad guys. So with that defense in mind, allow me to move on.

When testing software, we are in our own special kind of battle: a battle against crafty bugs that seek to destroy our software systems. Just as in real war, we can't simply jump into battle without preparation.



In war, soldiers don't just suit up and jump on the battle field. They prepare themselves and the field of battle first, so that they are in the best position to obtain success. When testing software, we are in our own special kind of battle: a battle against crafty bugs that seek to destroy our software systems. Just as in real war, we can't simply jump into battle without preparation. We need to make sure we have a certain level of battle readiness. We need to arm ourselves with the appropriate knowledge, resources, environment and tools.

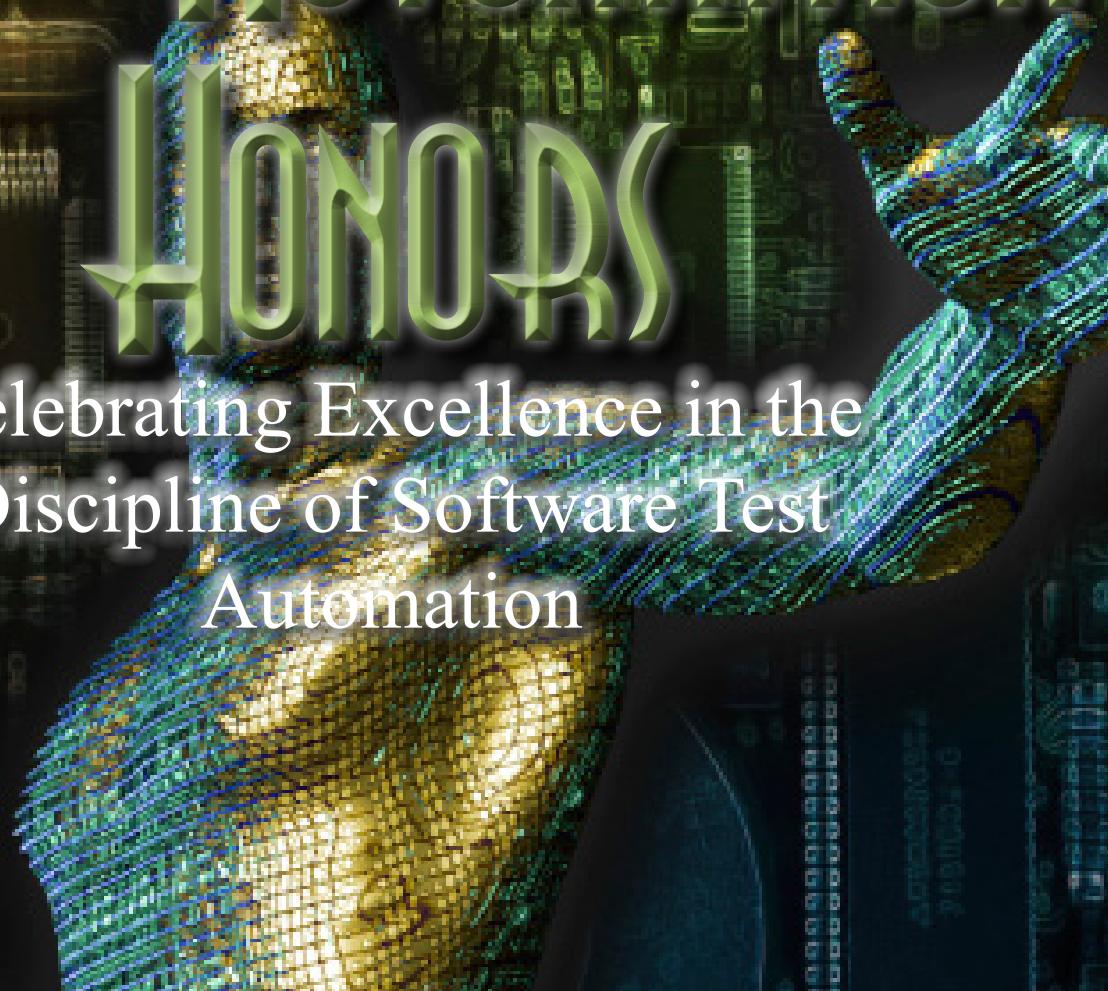
This issue of the magazine focuses on this type of preparation. The first feature entitled "Testing the REST" by Tom Wimsatt discusses a web services specification known as REpresentational State Transfer (REST), that is becoming fairly popular in software systems. He addresses this specification from

a tester's point of view offering information on how it works along with testing consideration that will help to equip you for, well... *testing the REST*.

Next, we take a look at important contemporary test environment considerations. In a feature article entitled "An Automated Test Lab In the Cloud," Mukesh Sharma discusses the importance of a well set up cloud-based automated test lab, and offers a potential solution that meets this need. Finally, we discuss the importance of performing an adequate level of preparation prior to the implementation of tools. In an article entitled "Behavior Driven Development Before the Tools," Liz Keogh describes the items that need to be addressed in behavior driven development (BDD) before heading down the path of implementing tools such as Cucumber. ■

5th Annual

ATI AUTOMATION HONORS



Celebrating Excellence in the
Discipline of Software Test
Automation

Winners Announced at TestKIT!

www.atihonors.automatedtestinginstitute.com

Authors and Events

Who's In This Issue?



Tom Wimsatt is an ISTQB/ASTQB Fully Certified Advanced Tester with Sotera Defense Solutions, Inc. where he is also a fellow with the Sotera Technical Community. He has been in the field of testing for more than 25 years beginning in the Parts Analysis labs with NASA at Goddard in Maryland.

He specializes in automated software testing developing tools and tests. He is a board member of the Automated Testing Institute (<http://AutomatedTestingInstitute.com/>). He can be reached at: Robert.Wimsatt@SoteraDefense.com

Mukesh Sharma is founder and CEO of QA InfoTech with a vision to provide unbiased Quality Assurance solutions for business partners worldwide. Responsible for the company's global operations, marketing, sales and development efforts, QA InfoTech, under Mukesh's leadership, has grown to five Centers of Excellence with over 700 employees. Mukesh is an active test evangelist, spreading the word on software testing through articles, conferences and most recently his book on a career in testing.



Liz Keogh is an experienced Lean and Agile consultant and well-known international speaker. Coming from a strong technical background, her work covers a wide variety of topics, from software development and architecture to psychology and complexity thinking. She is best known for her involvement in the BDD community, and was awarded the Gordon Pask award in 2010 for deepening existing ideas in the space and "coming up with some pretty crazy ones of her own".

AUTOMATED SOFTWARE TESTING

Managing Editor

Dion Johnson

Contributing Editors

Donna Vance
Edward Torrie

Director of Marketing and Events

Christine Johnson

A PUBLICATION OF THE AUTOMATED TESTING INSTITUTE



CONTACT US

AST Magazine

astmagazine@automatedtestinginstitute.com

ATI Online Reference

contact@automatedtestinginstitute.com

ATI and Partner Events

January 31, 2014

Baltimore Local Chapter Contact Deadline

dfern@verizon.net

March 4, 2014

ATI Honors Winners Honored

<http://www.atihonors.automatedtestinginstitute.com>

March 3-5

TestKIT 2014 Conference

<http://www.testkitconference.com>

The Automated Software Testing (AST) Magazine is an Automated Testing Institute (ATI) publication.

For more information regarding the magazine visit
<http://www.astmagazine.automatedtestinginstitute.com>

TESTKIT CONFERENCE 2014

Testing & Test Automation Conference

TestKIT

Testing & Test Automation
Conference

The Date



TestKIT

TESTING & AUTOMATION CONFERENCE



Save the Date!

MARCH 3 - 5, 2014
CROWNE PLAZA HOTEL
ARLINGTON, VA

The KIT is Komming... Back

www.testkitconference.com

Template For Automating the REST

Automating a RESTful Web Service with JUnit and Spring's Rest Template in 5 Steps



TestKIT is the name used by ATI for describing one's testing toolkit. A TestKIT is filled with knowledge, information and tools that go with us wherever we go, allowing our projects and organizations to quickly reap the benefits of the practical elements that we've amassed. This section provides tips to add to your TestKIT.

1. Get the RestTemplate Class
2. Create the HTTP Header
3. Set the HTTPEntity
4. Execute the HTTP Request
5. Interrogate the response in the ResponseEntity

The outcome of this process is illustrated in the JUnit test code shown in Figure 1.

STEP 1. GET THE RESTTEMPLATE CLASS

The Spring Framework can be obtained from GitHub (<https://github.com/spring-projects/spring-framework>). The import statement included in your JUnit class file will be as follows:

```
import org.  
springframework.web.  
client.RestTemplate
```

STEP 2. CREATE THE HTTP HEADER

The Header is where you specify whether you will be dealing with XML or JavaScript Object Notation (JSON) data. The sample in this article uses XML, which means the request body submitted to the web service and the response that you receive from that service will be in plain text XML, and will need to be handled as such.

“Web Services have been around so long now it's getting difficult to remember what things were like before them. The technology has continued to evolve and one has seemed to have risen above the crowd, at least in our circles – REpresentational State Transfer or REST. **”**

The above quote begins the article entitled “Testing the REST” by Tom Wimsatt that you will find featured in this issue of the magazine. In that article, Tom eloquently summarizes the importance of understanding Web Services and how to test them. Automation is an important part of this testing strategy. One automation approach is to have developers create a GUI test interface for accessing the web services and using

a GUI automation tool to automate tests via that interface. A second approach is to use an open source or commercial tool for connecting to and automating the web service tests. A third solution you may want consider is the creation of automated functional tests using JUnit and Spring's RestTemplate. This approach is often readily available and simple to implement in, given the fact that the RestTemplate is a freely available Java class, JUnit is a freely available framework, and a free IDE, such as Eclipse, can be used for test development. In addition, many projects use Eclipse and Java (JUnit) for development purposes, so there is already built in support for testers to implement this approach.

A functional test for a RESTful Web Service can be automated with JUnit and Spring's RestTemplate using the following 5 steps:

```

1  @Test
2  public void WebServiceTest1(){
3      //Step 2. Create The HTTP Header
4      HttpHeaders header = new NhttpHeaders();
5      header.setContentType(MediaType.APPLICATION_XML);
6      header.setAccept(Arrays.asList(MediaType.APPLICATION_XML));
7
8      //Step 3. Set the HTTP Entity
9      String xmlString = getXML();
10     HttpEntity<String> requestEntity = new HttpEntity<String>(xml,header);
11
12     //Step 4. Execute the HTTP Request
13     RestTemplate restTemplate = new RestTemplate();
14
15     ResponseEntity<String> response = restTemplate.postForEntity("http://local
16     host:12345/system/search", requestEntity, String.class)
17
18     //Step 5. Interrogate the Response in the Response Entity
19 }
```

Figure 1: Exercising a Web Service with RestTemplate

STEP 3. SET THE HTTP ENTITY

The HTTP Entity revealed in Line 10 in Figure 1 contains the Request Entity that will be submitted to the web service under test. In the previous step we determined that we would use XML in our request and response, therefore, we set the first parameter equal to the plain text XML that will go on the message body of the web service call. Line 9 is a placeholder line that calls a `getXML` method that you would either need to create to get your desired XML or replace with a hardcoded XML string. The second parameter in the HTTP Entity is set to the HTTP Header object constructed in Step 2 (Lines 4 through 6 of Figure 1).

STEP 4. EXECUTE THE HTTP REQUEST

It is now time to exercise our web service by submitting a request to it. There are two types of requests that will be considered in this article: GET and POST. The HTTP Entity from Step 3 is typically only used when the HTTP Request uses POST. Whether or not the Request uses GET or POST depends on how the web service is defined.

If the web service is defined to use POST, then the `postForEntity` RestTemplate method is used as shown in Lines 13 through 15.

Note that the method has three parameters:

- **Parameter 1:** Web Service URL
- **Parameter 2:** The HTTP Entity (Request Entity) defined in step 2.
- **Parameter 3:** The Response Type. Since we're using plain text XML, the type is `String.class`

If the web service is defined to use GET, then Line 13 would be modified as follows:

```
ResponseEntity<String>
response = restTemplate.
getForEntity("http://
localhost:12345/system/
search", String.class)
```

Note that the `postForEntity` method is replaced with the `getForEntity` method and that the `requestEntity` parameter is not being used.

In the event that the web service has request parameters instead of, or in addition to a message body (which is placed in the `HttpEntity`), then URI Variables will need to be submitted as part of the HTTP Request. This will require two changes. The first change is that the web service URL will need to include the necessary parameter(s).

For example, if the web service requires a parameter called 'userLogin' and a parameter called 'userPassword', the URL submitted will need to be altered as follows:

```
"http://localhost:12345/
system/search?userLogin=
{userLogin}&userPassword
={userPassword}"
```

Note that the word `userLogin` is included twice, once to the left of the equal sign and once to the right of the equal sign inside of brackets. The same holds true for the `userPassword` parameter. The actual values of the `userLogin` and `userPassword` parameters will not be added to the URL.

That brings us to the second change. The URI variable values will be added to the `getForEntity` or `postForEntity` method as follows:

```
Object [] uriVariables
= new Object[]{new
String("johnuser"),new
String("johnpassword")};

ResponseEntity<String>
response = restTemplate.
postForEntity("http://
localhost:12345/system/
search?userLogin={userLo
gin}&userPassword={userP
assword}", String.class,
uriVariables);
```

Upon executing either the `postForEntity` or `getForEntity` method against the web service, the plain text XML response will be returned in the `ResponseEntity` `response` variable.

STEP 5. INTERROGATE THE RESPONSE IN THE RESPONSEENTITY

This step largely depends on the test being performed. The XML response will be returned to the `response` entity, so you are now free to traverse the web service response elements to ensure it has returned the expected results. ■

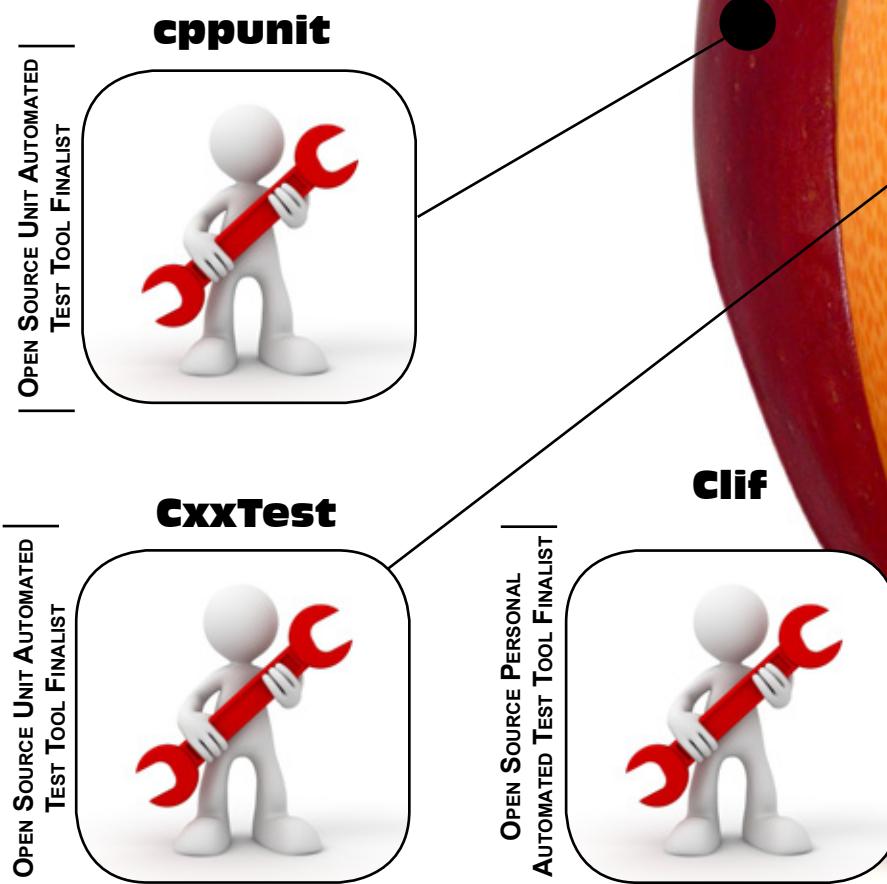
Wait! There's More!

Read more information on REST in the AST article entitled, "Testing the REST"

Fresh Automation Fruit

Best 'New' Tools

New open source tools seem to crop up everyday. In the ATI Automation Honors, however, many of the same tools seem to continuously rise to the top. Each year, however, some fresh new tools break through the 'wall of regulars' nominated by the community to find themselves among the list of finalists. In honor of those new tools that made the finals in the 5th Annual ATI Honors, a new set of Best New Entry categories were created. The Best New Entry nominees are highlighted here.



SpockOPEN SOURCE UNIT AUTOMATED
TEST TOOL FINALIST**Capybara**OPEN SOURCE FUNCTIONAL
AUTOMATED TEST TOOL FINALIST**Mechanize**OPEN SOURCE FUNCTIONAL
AUTOMATED TEST TOOL FINALIST**Wetator**OPEN SOURCE FUNCTIONAL
AUTOMATED TEST TOOL FINALIST

People love new tools, because they are refreshing like... fresh fruit. The new tools in this year's Automation Honors awards are separate, but collectively make up one single ATI entity known as our Best New Tool fruit.

Wait! There's More!

Additional Best New Entry Tools not listed include: Siege and Tsung. Learn more at:

[http://atihonors.
automatedtestinginstitute.com](http://atihonors.automatedtestinginstitute.com)

TESTING THE R&D

ONSE

EST

E

S

WEB SERVICES HAVE BEEN AROUND SO LONG NOW IT'S GETTING DIFFICULT TO REMEMBER WHAT THINGS WERE LIKE BEFORE THEM. THE TECHNOLOGY HAS CONTINUED TO EVOLVE AND ONE HAS SEEMED TO HAVE RISEN ABOVE THE CROWD, AT LEAST IN MY CIRCLES – **REPRESENTATIONAL STATE TRANSFER OR REST.** TESTING SERVICES THAT EMPLOY THIS ARCHITECTURE HAVE SPECIAL CHALLENGES BUT ITS SIMPLICITY WILL HAVE YOU WONDERING, “WHY DIDN’T THEY THINK OF THIS LONG AGO?”



RESTFUL INTERFACE TESTING AND ITS CHALLENGES

by Tom Wimsatt

WEBSERVICESTESTINGREQUEST



The author of the REST specification, Thomas Fielding, has a number of resources available to help you get more details about it, but it is instructive to know that he was one of the authors of HTTP and the communication is based on that protocol. The following concepts apply:

- Stateless – each message can stand alone. There are no set up or tear down commands as is common with audio communications.
- Transfer XML or JavaScript Object Notation (JSON) – Key/Value pairs of data elements. There cannot be a value without a key. The value may be empty but it cannot be null.
- Text based – the stream of data can be read quite easily without a format document or character map.

For the purposes of this article, we will concentrate on JSON. The same concepts can be easily applied to XML messages.

Also, there are very few commands defined for HTTP:

- GET
- POST
- PUT
- PATCH
- DELETE

The responses are all HTTP, as well (e.g. 404 – Not Found). The basics are all well understood and easily implemented. Java, C# and many other modern languages have libraries making all of this straight forward.

Testing this interface entails tackling areas usually reserved for website testing:

- Incomplete requests.
- Correct responses (not just codes but formats).
- Large POSTs causing a condition like a denial of service.
- Multiple repeated requests which is, basically, a denial of service.

INCOMPLETE REQUESTS

HTTP is very forgiving as far as content goes. There is no real penalty for sending a malformed HTML document as the browser just interprets what it can and drops the remaining as unintelligible. However, if you are posting a stream of data to a database system, partial messages are unacceptable. In this case, we must take care that the server deals appropriately with the missing content by rejecting the post and returning a code indicating the problem such as 400.

There are several ways to force an incomplete request such as:

- A key with a null value. There is nothing that translates NULL in text.
- A key with no matching value – two keys back to back in the message. For example: {key1 : } {key2 : value}
- Drop a closing character such as the curly brace: {key1:value1 {key2:value2}.
- Drop required fields.

The last one is a great test to run as a sequence of drops of each required field and then attempt to drop more than one field at a time. It is typical to expect and handle a single missing field but two or more has resulted in some surprising reactions on the servers I have tested.

One final test that is related but does not fall entirely in this category is sometimes called a misdirected post. This is the case where the URI is missing one or more of the parts of the hierarchy resulting in the code listening at that location to attempt to process the request. In a similar vein, attempt to send to a location that exists on the server but is not expecting this request.

CORRECT RESPONSES

It is one thing to get a simple HTTP 500 error when the server rejects the request, but is it appropriate or even

helpful? Also, what is the format of the response? If we are sending JSON, we should expect the responses to be formatted in JSON. Modern web servers typically do this automatically but the web service may sit between our server application and the clients. In this case, the back end server may need to format it's response.

HTTP STATUS CODES

A great place to start is the World Wide Web Consortium (W3C.org) for the definitions of the codes. To some extent, there may be some interpretation on just what some of the codes mean for your system but it is safe to say that “500” is not always appropriate. Some examples are:

- 200 OK – The basic response that the request worked.
- 400 Bad Request – Expected in the case of malformed requests such as incomplete requests discussed earlier. 404 Not Found is a related error in this case, as well.
- 401 Not Authorized – An important security related error.
- 502 Temporarily Overloaded – This is subject to interpretation and a decision must be made as to who then is responsible for the data. Does the client resend on some predetermined schedule or is the server going to handle the request from cache?

HELPFUL RESPONSES

In addition to giving the correct HTTP response, the server should return some information to help determine the cause. In the cases of incomplete requests, if the server can tell the client which field is missing, that can go a long way toward troubleshooting for all concerned.

Some thought should have been given also to indicate who is responding. A malformed URL can be caught by the web server itself and return some ID to indicate this, while missing data in the stream can be caught by the back end server and passed up to the web server.

LARGE POSTS

A set of tests for large posts must be done in two parts: individual field limits and an extremely large message.

The first case, individual field limits, should already be part of the testing as the back end database or server must have some limit checking and the determination

made as to what is done when a field is too large:

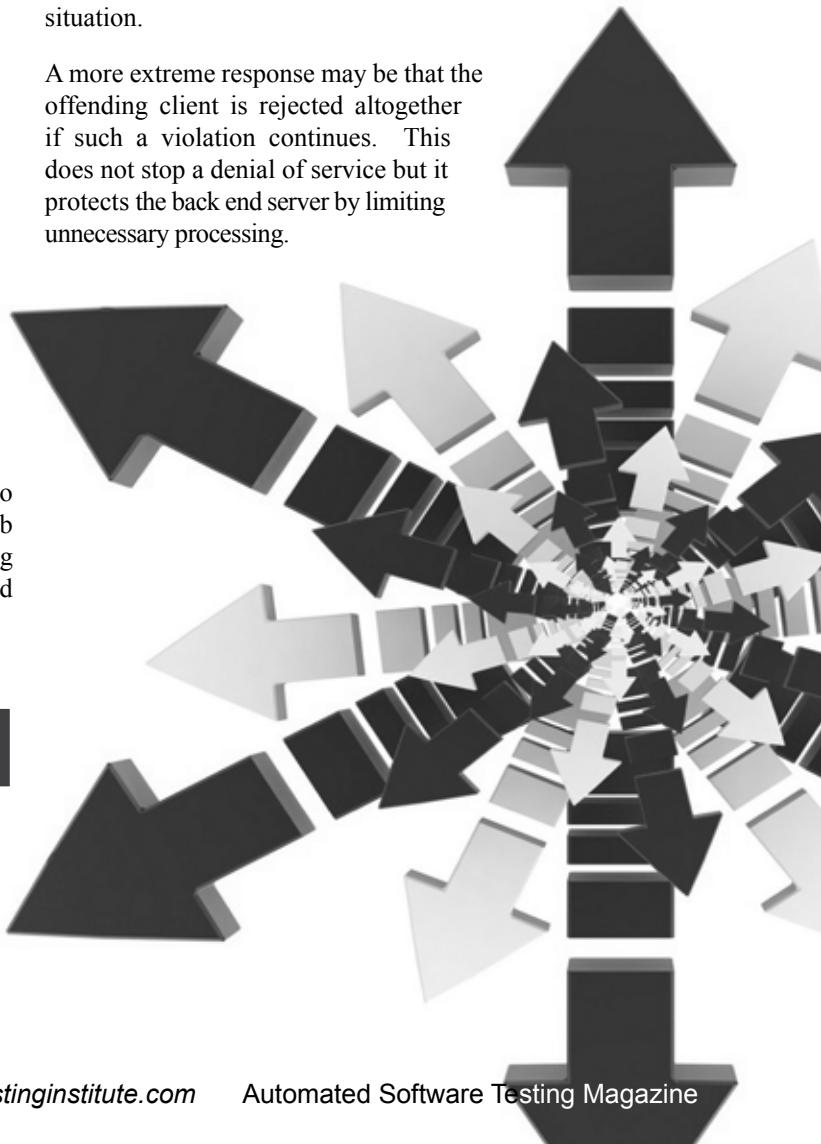
- Truncate,
- Split into another field – in the case of critical data such as bank transactions, or
- Drop the request and respond with an error.

There are two basic ways to test an extremely large request:

- Insert data up the maximum field sizes for every field, or
- Exceed the size of one or more fields regardless of the field limit.

In each case, the URI can be tied up with processing the request leading to other failure conditions that must be addressed. In the first case, the server handling of limits may mitigate this issue and all that you need to do is make sure the fields are correctly handled. In the second case, one or more fields are beyond the limits to such an extent that the web sever and/or the back end server chokes on the request. In this case, the appropriate status code should be returned telling the client something helpful to correct the situation.

A more extreme response may be that the offending client is rejected altogether if such a violation continues. This does not stop a denial of service but it protects the back end server by limiting unnecessary processing.



There is no way to do this type of testing without automation. There are tools to consider such as SoupUI, Jmeter, Junit. It is likely that a home-grown solution will need to be developed.

This type of testing is best done by ramping up the sizes until a problem is encountered in order to see where the possible breakpoints may be hiding.

MULTIPLE REPEATED REQUESTS

This is a basic denial of service type of test. There are a couple of points to consider when working in this area:

- Timing between the requests – Decrease the time between the requests until something breaks.
- Increasing number of repeated requests – Submit an increasing number of requests without any delay between them until something breaks.

Again, when something does give in either case, what is the appropriate response? It is not, “We will never see that in production”! Every time I have heard that, it seems, two weeks go by and the same thing happens in production! That’s not just for this type of test, either.

IN GENERAL

When working with web services fronting databases as I do, here are some general tips and guidelines to consider:

- Response times can get quite lengthy. Reports from databases can be of special concern when testing all of the different filter combinations.
- The returning datasets can get quite large and require another tool or technique to verify the data. Consider the case of multiple repeated requests – were all of the responses correct or was there a problem beginning to show up in the data before the system crashed or responded that there was a problem?
- There is no way to do this type of testing without

automation. There are tools to consider such as SoupUI, Jmeter, Junit. It is likely that a home-grown solution will need to be developed.

- Consider running these tests overnight or on a separate system to avoid interrupting other testing or development efforts. This type of testing can get quite destructive as the system may block the URI or cause any number of crash worthy conditions.
- Consider using multi-threaded techniques when developing your tools to take advantage of sending requests and processing responses at the same time. This will also help in doing some of the more extreme denial of service tests as described.

IN CONCLUSION

REST is a simpler web service than the competition leading to an increase in its adoption rate; so, you will see it eventually. Fortunately, its simplicity makes for a short learning curve and let's us concentrate on the tests themselves. Also, troubleshooting can be easy as the protocol is text and in key/value pairs as defined by existing standards: JSON and XML. Being built on the HTTP standard also provides a common point to start from. However, for all of its simplicity there is the additional complication of adding in testing like a website even though the URI may not be exposed to browsers on the web. The challenges are similar if not the same and must be tested, as well. ■■■

Wait! There's More!

Learn more about how to Test the REST by attending Wimsatt's presentation at TestKIT 2014.

Also, get more automation tips in the article entitled “Template for Automating the REST”

TestKIT^{OD} ONDEMAND

If you can't be live, be virtual



ONDEMAND Sessions

Anywhere, anytime access to testing and test automation sessions

-
- >>> Available anywhere you have an internet connection <<<
 - >>> Learn about automation tools, frameworks & techniques <<<
 - >>> Explore mobile, cloud, virtualization, agile, security & management topics <<<
 - >>> Connect & communicate with testing experts <<<

ONDEMAND.TESTKITCONFERENCE.COM

>> OPENING IDEA

Collaboration and looking for ways to improve efficiencies in operations have become important survival factors in all disciplines, and the field of software development and the testing discipline are no exception.



An Autom... Lab In t...

By Mukesh

WOULDN'T IT BE NICE TO
HAVE AN AUTOMATED CLOUD
BASED LAB SOLUTION THAT
IS VERSATILE?



h Sharma

Collaboration and looking for ways to improve efficiencies in operations have become important survival factors in all disciplines, and the field of software development and the testing discipline are no exception.

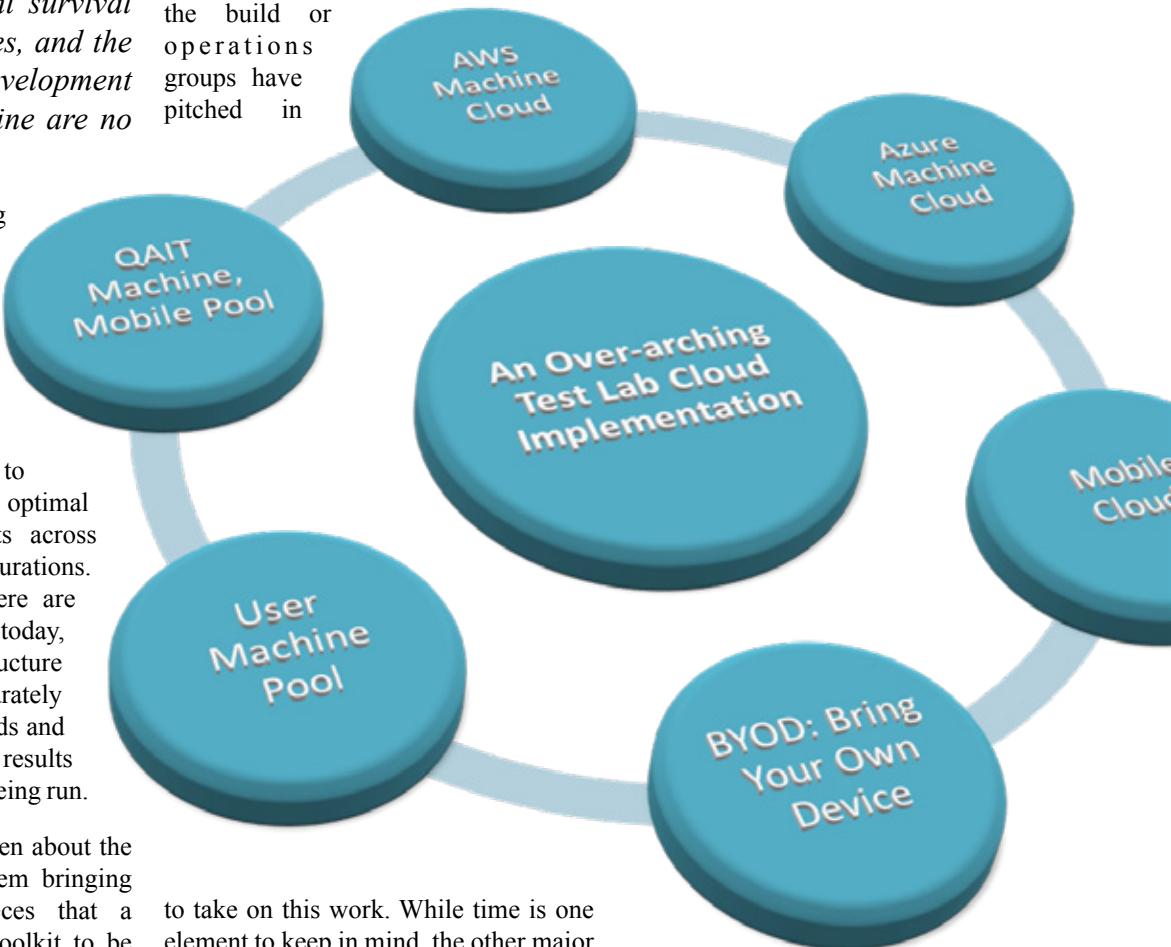
Test case design, planning for what tests to execute and ways to optimize the overall testing effort while striving to maximize coverage together form one piece of the pie. Another equally important area for the team to focus on, is determining the optimal strategy in executing tests across the required test configurations. This is an area where there are still a lot of inefficiencies today, in terms of unused infrastructure bandwidth, machines disparately spread between public clouds and local labs and insufficient results monitoring while tests are being run.

Earlier this year I had written about the need for a testing ecosystem bringing together the various pieces that a tester would need in his toolkit to be empowered to focus on his core testing task in building a product of superlative quality, not having to worry about the overhead of dealing with isolated and disparate solutions that he uses on a daily basis. While the testing discipline as an industry is still far away from implementing such a comprehensive ecosystem, there are some players in the market today, who offer selective pieces as a product rather than as an ecosystem, especially in the area of test labs.



Machine setup and maintenance have always been time guzzling activities on a tester's plate. In teams that are collaborative in their efforts, the build or operations groups have pitched in

systems and browsers. Mobile clouds are becoming inevitable in a tester's toolkit, to give him mobile connections on the go. Similarly a tester needs a range of



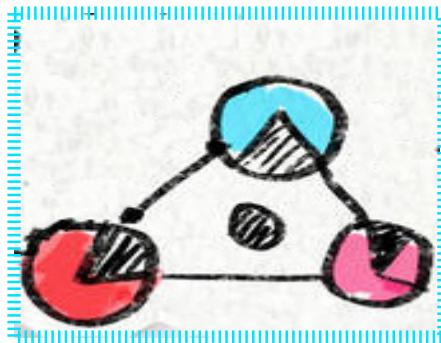
to take on this work. While time is one element to keep in mind, the other major pieces to watch for include, the cost of machine setup, whether the machines are being used optimally to their fullest capacities and whether they are available on demand as and when the teams need them.

The advent of public clouds such as Amazon's EC2, Microsoft's Azure have largely alleviated these problems, making machines available on demand. Also given their scale of operations, availability of machines has typically not been an issue with such big players. However given how diverse the needs of a tester are, we need to analyse whether these commercial solutions are scalable enough – not just from a size standpoint, but more importantly from scope and features standpoints. For instance, testers' needs are no longer confined to just machines clouds – a cloud of machines providing desktop operating

machines even in a machine cloud – machines that cover a range of operating systems including ones where support may have officially been withdrawn by the software maker. For instance, as part of backward compatibility testing, a tester may need an XP machine on his AWS account to execute tests for end users who may still be using Windows XP. Similarly, there aren't public clouds for MAC machines as of today. Microsoft Azure only provides a cloud of server machines. While these may not be limitations running into the future, as of today, a tester who wants to use the cloud is crippled by these shortcomings.

On a more welcoming note, there are a few players who offer better diversity, providing both a cloud of machines and a cloud of mobile systems to operate

from which has been very useful for the testing community. However, these are still individual services which don't take into account a very big factor of efficiency in operations. When a tester moves into using the cloud, be it specific machine clouds or a combination of both machine and mobile clouds, what happens to the existing lab of machines they already have? Let us say, an organization has about 1000 machines in



which is open source and automated, helping us build a test lab on the cloud not just for our needs but also for our clients who run several more automated tests at their ends daily. This is a solution that we envisioned in our test automation research and development group and as a first step towards building such an ecosystem, we developed a solution that currently has or is planned to have the following features:

A SOLUTION WHICH IS OPEN SOURCE AND AUTOMATED, HELPING US BUILD A TEST LAB ON THE CLOUD NOT JUST FOR OUR NEEDS BUT ALSO FOR OUR CLIENTS WHO RUN SEVERAL MORE AUTOMATED TESTS AT THEIR ENDS DAILY

its lab as of today, and decides to take the big leap to using the cloud. Overnight, these 1000 machines all of a sudden become useless to the organization – what would it do with these machines? Do they dispose of them or keep them for a potential later use? If they decide to dispose of them isn't the cost of doing so, quite high, especially given that most machines are relatively new ones that the team has been using? If they decide to keep them for some more time, for a potential future use, isn't this a huge storage and maintenance cost for them to bear with no immediate returns? These are equally important considerations to bear in mind, when a decision to move to the cloud is being made.

Having set the premise, both in terms of benefits as well as core points to analyse in moving to a cloud, wouldn't it be nice to have an automated cloud based lab solution that is versatile in addressing the points discussed above? – A solution that incorporates:

- Public clouds across providers covering both machines and mobile devices along with their associated operating systems and browser options
- The existing pool of devices that the tester brings in from his organization
- Both queued and on-demand test execution
- Detailed test monitoring and data collection from unattended test runs, to provide the testers enough ammunition to debug and troubleshoot an issue at a later point in time.

At QA InfoTech, an independent testing services company, we execute hundreds of thousands of tests for our clients on a daily basis. We continue to face the issues mentioned above in our test automation execution, and realized the need for a solution as outlined above. A solution

1. A user friendly interface to play around with the various execution options
2. Option to choose between on-demand and queued test execution
3. An implementation that lets you integrate more than 1 public cloud and bring your own machines, letting you maximize usage of your available resources and minimize any wait times
4. Detailed logs including video recordings of test execution, helping testers review results especially when they were run in an un-attended mode

The solution that we implement is a first step towards building a larger testing ecosystem that will become a comprehensive tool kit for testers in the coming years. ■

Wait! There's More!

More about QA InfoTech's solution at

<http://www.qainfotech.com/>

Automated Testing

HOME

FRAMEWORKS

TECHNIQUES

ABOK

TOOLS

FAQS

RESOURCES

NEWS

Weekly Automation Tube Featured Videos

Automated
Testing Institute

testing techniques tools

HOME

FRAMEWORKS

TECHNIQUES

ABOK

TOOLS

FAQS

RESOURCES

NEWS

Weekly Automation Tube Featured Videos



Google:
Interaction Based Testing



Google:
Frakenstein and Swing



GoogleDevelopers:
SeleniumUserMeeting



pycon08:
py.test



ATIAutomationTube:
This Week In Automation



ATIAutomationTube:
Automation ROI Teaser



ATIAutomationTube:
Automated Testing Institute

Check out this week's featured Automated Testing Videos from the web. For more information about ATI click on the header links, or click [here](#).

Weekly Featured Articles

An article is featured in each of the following categories:
Lifecycle/Methodology, Frameworks, ROI,
Techniques/Toolsmithing, Performance/Load, Security
and Unit.

Open Source Test Automation



Summary: This paper introduces the reader to Open-Source Test Automation (OSTA) and outlines the procedure for

ATI's Latest And Greatest

- Espionage and Software: U.S. Vulnerable to Cyberspies!?
- Microsoft Powerpoint Defect Posing Security Risk
- 'Software Problem' Results in Shocking Letter To Parents of Dead Teen
- Benerator Releases 0.5.8
- Software Testing: The Fastest Growing IT Service
- New Year, New Level: What's Next in Automation
- Catching a Code in the Summer
- HP Release QTP 10.0

Automation (OSTA) and

HP Release QTP 10.0

search...

ADVERTISE SEARCH ABOUT

Login Form

Welcome to ATI! If you're registered feel free to log in. If not, register to receive the full benefits of the site.

Username



Password


Remember Me **Login**

- [Forgot your password?](#)
- [Forgot your username?](#)
- [Create an account](#)

Profile Management Menu

- [About ATI](#)
- [Why Register?](#)
- [Getting Started With Automation](#)

Who's Online

We have 203 guests online

Additional Information

[Expert Software QA & Test](#)

Achievo cuts product QA and testing time by 40% while slashing costs.

www.achievo.com

[Top 5 QA Testing Tools](#)

2009 Top 5 QA Testing Tools Vendor Rankings. Get Free Report
Business-Software.com/QATesting

Login Form

Welcome to ATI! If you're registered feel free to log in. If not, register to receive the full benefits of the site.

The Automated Testing Institute (ATI) Online

Username



Institute (ATI) Online

Password



Reference is the web's

Remember Me

foremost comprehen-

- [Forgot your password?](#)

- [sive online resource for](#)

- [Create an account](#)

obtaining, exchanging

Profile Management Menu

and certifying industry

- [Why Register?](#)

standard test automation

knowledge, information,

Who's Online
 We have 8 guests online

and techniques.

Additional Information

[Expert Software QA & Test](#)

Achievo cuts product QA and testing time by 40% while slashing costs.

www.achievo.com

VISIT US TODAY!

[Top 5 QA Testing Tools](#)

2009 Top 5 QA Testing Tools Vendor

Rankings. Get Free Report

Business-Software.com/QATesting



BDD isn't about the tools. It's about the conversations you have, exploring examples (or scenarios) of an application's behaviour, to see if everyone has the right understanding.

(based on her popular blogs)

BEHAVIOR DRIVEN DEVELOPMENT BEFORE THE TOOLS

By Liz Keogh

Things to Focus on Prior to Heading Down the Tools Path

I was approached by a client a couple months ago about Behavior Driven Development (BDD) and using tools like Cucumber to automate scenarios. While tools can be useful, there are a few things I'd love to see teams focus on before heading down the tools path, including:

- An eye on the big picture,
- The ability to question and explore,
- The ability to spot and embrace uncertainty,
- Developing great relationships between people.

Upon doing these things, you may increase the chances of tools proving useful, or conversely, you may decide that you've unleashed enough benefits to render tools unnecessary.

AN EYE ON THE BIG PICTURE

Who are the stakeholders? Who championed the project and who else has needs that must be met to make that vision live? Answering this also helps teams come up with much smaller Minimum Viable Product (MVPs). It should help to define which scenarios are in and out of scope, and also tends to lead to better language and better appreciation of who to talk to for getting a hold of the scenarios in the first place. Chris Matts introduced me to a great technique for accomplishing this known as Feature Injection. Feature Injection is defined as:

“A business analysis process framework that allows teams to exploit the value of the traditional business analysis techniques in projects involving frequent iterative deliveries, such as the ones implemented using agile and lean processes. It focuses test-driven-development and behaviour-driven-development on delivering business value and ensures that a team only builds features and projects that deliver value, avoiding the wastes of scope creep and just-in-case code.”

The following points further expound on how Feature Injection works.

Every project worth pursuing has a *vision* - either something new, or something that you've seen someone else implement that you'd like to implement as well. The vision is designed to increase revenue, decrease costs or protect revenue – for instance, keeping customers from going to a competitor.

There may be other stakeholders whose goals need to be met in order to allow the vision to go live. Frequently these stakeholders have traditional gatekeeping roles – UAT, Architecture, Legal, Security. By recognizing these stakeholders early, we get a chance to turn them into educators instead.

To meet the stakeholder's goals, we create *capabilities*. For instance, employees might need the capability of booking a holiday. Notice that they could quite easily do that with paper! To make our differentiating vision compelling, we may need a certain amount of commoditised functionality – things that people will expect in a software application that books holidays. An example of a differentiator is the first ever camera in a phone. We'll also need to be able to make calls, receive calls, look up numbers, etc.

To deliver the capabilities, we design *features*. We may decide that some features are too complex, and create manual workarounds instead. The features represent the way in which users will use the capabilities we're giving them – through a browser, a windows client, a mobile phone, etc. Now we're starting to move into the areas of UI design, preferably with a UI designer, and developers and testers can really get involved if they weren't already! Even if the development team wasn't involved in the conversations before this point, I recommend they be aware of the outcomes so they can map their features back to the capabilities,

stakeholder goals and initial vision.

To really explore the features, we talk through scenarios; examples of the outcomes a system should produce when a user uses it in different contexts. We can use the scenarios to decide what we want to get feedback on, and thus divide the features into stories. The main purpose of dividing things into stories is to be able to get feedback quickly.

THE ABILITY TO QUESTION AND EXPLORE

I generally use two conversational patterns when talking through scenarios:

- Context Questioning
- Outcome Questioning

Both of these patterns are based on the typical BDD scenario template:

Given a context
When an event happens
Then an outcome should occur.

CONTEXT QUESTIONING

We can ask, "Is there any other context which, when this event happens, will produce a different outcome?"

It might be an additional context, or a context which occurs instead of the one(s) we're considering. We can ask this question, or we can think of contexts ourselves and ask about those. For instance:

Given a client manager, Fred, has two clients, George and Henry

When Fred uses the application

Then Fred should only be able to see George and Henry's accounts.

I ask, "Given I have no client managers, what happens then?" (I'm thinking of handling empty lists, because I'm a developer.)

The head of Client Management says, "What? Who cares? Why are you even using the application? Get on the

" THE ABILITY TO SPOT AND EMBRACE UNCERTAINTY IS OFTEN THE FIRST THING THAT GOES OUT OF THE WINDOW WHEN PEOPLE HEAD DOWN THE TOOLS PATH "

#	BDD Focus
1	Eye On the Big Picture
2	The Ability To Question and Explore
3	The Ability To Spot and Embrace Uncertainty
4	Developing Great Relationships Between People

Figure 1: Areas to Focus On Prior To Tool Implementation

“

IN SOFTWARE DEVELOPMENT, TOOLS SHOULD SUPPORT YOUR INTERACTIONS, NOT DEFINE THEM.

phone and get some clients already!” (He’s thinking of his failing business.) You can sometimes learn a lot about a business domain this way, even if the behaviour of the application isn’t going to change immediately!

OUTCOME QUESTIONING

This is similar to Context Questioning, but in this pattern, we ask, “Given this context, when this event happens, is there another outcome that’s important? Something we missed, perhaps?” For instance:

Given the legacy system has a problem

When a trader stores a trade

Then the trader should be told about the problem.

I ask, “Is that all that should happen? I’d be pretty upset if I were a trader and this happened a lot.” My business analyst says, “Well, we’ll still have the details in the system, so you’ll be able to do it again later. Oh, yes! We need to store the trade in our system still, too, but make sure it doesn’t have a legacy system ID so that we can tell it didn’t go through.”

Testers are extremely good at coming up with scenarios that may have been missed, identifying stakeholders whose outcomes need to be considered, etc.

THE ABILITY TO SPOT AND EMBRACE UNCERTAINTY

The ability to spot and embrace uncertainty is often the first thing that goes out of the window when people head down the tools path. If something has never been done before, it’s highly unlikely that the business will know exactly what they want. They might not be able to define all the scenarios. If the team hears that there’s uncertainty

or conflict in requirements, rather than attempting to nail it down, collaborate instead to try something out instead – spiking, prototyping, A/B testing, etc. The scenarios will emerge. This lets teams focus on risk.

DEVELOPING GREAT RELATIONSHIPS BETWEEN PEOPLE

If people want to use BDD to understand how to help the business better, to help out the testers, or to help the developers code the right thing, then I want to ensure that desire to help genuinely exists. There are lots of small ways to help each other that don’t involve BDD. A focus on “stop starting, start finishing”, limiting WIP and helping each other finish off work and get feedback on it, is a good sign. A heavy focus on estimates, commitment and “sprint failures” is not.

IF YOU HAVE THESE THINGS, GO AHEAD AND USE THE TOOLS IF YOU WANT TO

It normally takes a couple of weeks to a couple of months for teams to get the hang of the conversational side of BDD. During this period a lot of successful teams use the half-way house of just writing the scenarios down, perhaps on a wiki, where they’re easy to access and change. Some teams focus particularly on the scenarios which are unexpected or from which they learned a lot, and that’s a good thing. I like to see devs writing the scenarios down if anyone does, as they can then play them back to the business afterwards, and it forces the conversations to happen. When teams embark on the tool route prior to addressing the items I’ve discussed in the article the following often occurs:

- The team ends up producing the wrong thing (which is now hard to change, because automation also cements it in place).
- The team often stops having conversations altogether, as the business or business analysts (BAs) start writing scenarios down ahead of time and passing them to the devs, losing the ability to spot any misunderstandings.
- Innovation is stifled and the scenarios start to become used as “contracts” for what the devs produce, creating an “us and them” culture, undoing all the good work of Agile, and pushing risk towards the end of the project.
- Any schisms which were there before are hugely magnified, and the team is now starting to head down a Waterfall path, only with lots and lots of scenarios instead of lots and lots of paper
- Sprint planning meetings take 4 hours.

The good news is that if you do go down the tool route and find this happening, it can be reversed fairly quickly by focusing on the conversations again. In software development, tools should support your interactions, not define them. ■

Latest From the Local Chapters

Local Chapter Announcement



NEW LOCAL CHAPTER!

Hey Baltimore! We are forming an Automated Testing Institute (ATI) Local Chapter in Baltimore, Maryland (USA) and want to invite you to join. ATI is the premiere organization worldwide for software test automation. If you've attended ATI's TestKIT conference you would agree that it is a "World Class" organization sharing software test automation knowledge. How would you like to meet on a regular basis with fellow automation Engineers in the **Baltimore, Maryland** area to:

- "Talk shop",
- Share ideas with other automation professionals,
- See new tools, and
- Learn the Test Automation Body of Knowledge (TABOK)?

Here's your chance.! If you are interested in participating in the Chapter leadership or just want to attend the meetings, please send me your e-mail so I can invite you to the first chapter planning conference call.

Please send me your contact information by 1/31/2013 so I can include you in on the first call.

Thanks,

David Fern

"Automatewisely my friends."
dfern@verizon.net

ATI LOCAL CHAPTER PROGRAM

**HELP PROVIDE
COMPREHENSIVE, YET
READILY AVAILABLE
RESOURCES** that will aid
*people in becoming more
knowledgeable and equipped to
handle tasks related to testing
and test automation*

**ENHANCE THE
AWARENESS OF TEST
AUTOMATION** as a discipline
*that, like other disciplines,
requires continuous education
and the attainment of a standard
set of skills*

**OFFER TRAINING AND
EVENTS** for participation by
*people in specific areas around
the world*

ATI's Local Chapter Program is established to help better facilitate the grassroots, global discussion around test automation. In addition, the chapter program seeks to provide a local base from which the needs of automation practitioners may be met.

Start a Local Chapter Today

Email contact(at)automatedtestinginstitute.com to learn more



TRAINING THAT'S PROCESS FOCUSED, YET HANDS ON
Software Test Automation Training

WWW.TRAINING.AUTOMATEDTESTINGINSTITUTE.COM

PUBLIC COURSES

- Software Test Automation Foundations
- Automated Test Development & Scripting
- Designing an Automated Test Framework
- Advanced Automated Test Framework Development
- Mobile Application Testing & Tools

VIRTUAL COURSES

- Automated Test Development & Scripting
- Designing an Automated Test Framework
- Advanced Automated Test Framework Development
- Mobile Application Testing & Tools

Come participate in a set of test automation courses that address both fundamental and advanced concepts from a theoretical and hands on perspective. These courses focus on topics such as test scripting concepts, automated framework creation, ROI calculations and more. In addition, these courses may be used to prepare for the TABOK Certification exam.

Public and Virtual Training Available

Are you Contributing Content Yet?

The **Automated Testing Institute** relies heavily on the automated testing community in order to deliver up-to-date and relevant content. That's why we've made it even easier for you to contribute content directly to the ATI Online Reference! Register and let your voice be heard today!



As a registered user you can submit content directly to the site, providing you with content control and the ability to network with like minded individuals.

>> COMMUNITY COMMENTS BOX - This comments box, available on the home page of the site, provides an opportunity for users to post micro comments in real time.



>> ANNOUNCEMENTS & BLOG POSTS - If you have interesting tool announcements, or you have a concept that you'd like to blog about, submit a post directly to the ATI Online Reference today. At ATI, you have a community of individuals who would love to hear what you have to say. Your site profile will include a list of your submitted articles.



>> AUTOMATION EVENTS - Do you know about a cool automated testing meetup, webinar or conference? Let the rest of us know about it by posting it on the ATI site. Add the date, time and venue so people will know where to go and when to be there.

Learn more today at <http://www.about.automatedtestinginstitute.com>

Automated Testin

HOME

FRAMEWORKS TECHNIQUES ABOK

Automated
Testing Institute
testing techniques tools

HOME

FRAMEWORKS

TECHNIQUES

ABOK

TOOLS

FAQS

RESOURCES

NEWS

Weekly Automation Tube Featured Videos



Google:
Interaction Based Testing

Google:
Frakenstein and Swing

GoogleDevelopers:
SeleniumUserMeeting

pycon08:
py.test

ATIAutomationTube:
This Week In Automation

ATIAutomationTube:
Automation ROI Teaser

ATIAutomationTube:
Automated Testing Institute

Check out this week's featured Automated Testing Videos from the web. For more information about ATI click on the header links, or click [here](#).

Weekly Featured Articles

An article is featured in each of the following categories:

Lifecycle/Methodology, Frameworks, ROI,

Techniques/Toolsmithing, Performance/Load, Security and Unit.

Open Source Test Automation



Summary: This paper introduces the reader to Open-Source Test Automation (OSTA) and outlines the procedure for implementation.

ATI's Latest And Greatest

- Espionage and Software: U.S. Vulnerable to Cyberspies!?
- Microsoft Powerpoint Defect Posing Security Risk
- 'Software Problem' Results in Shocking Letter To Parents of Dead Teen
- Benerator Releases 0.5.8
- Software Testing: The Fastest Growing IT Service
- New Year, New Level: What's Next in Automation
- Catching a Code in the Summer
- HP Release QTP 10.0

Automation (OSTA) and outlines the procedure for

■ HP Release QTP 10.0

ADVERTISE SEARCH ABOUT

Login Form

Welcome to ATI! If you're registered feel free to log in.
If not, register to receive the full benefits of the site.

Username

Password

Remember Me

Login

- [Forgot your password?](#)
- [Forgot your username?](#)
- [Create an account](#)

Profile Management Menu

- [About ATI](#)
- [Why Register?](#)
- [Getting Started With Automation](#)

Who's Online

We have 203 guests online

Additional Information

[Expert Software QA & Test](#)

Achievo cuts product QA and testing time by 40% while slashing costs.

www.achievo.com

[Top 5 QA Testing Tools](#)

2009 Top 5 QA Testing Tools Vendor Rankings. Get Free Report
Business-Software.com/QATesting

Login Form

Welcome to ATI! If you're registered feel free to log in.
If not, register to receive the full benefits of the site.

The Automated Testing

Username



Institute (ATI) Online

Password



Reference is the web's

Remember Me

foremost comprehen-

[Forgot your password?](#)

sive online resource for

[Create an account](#)

obtaining, exchanging

Profile Management Menu

and certifying industry

[Why Register?](#)

standard test automation

knowledge, information,

We have 8 guests online

and techniques.

Additional Information

[Expert Software QA & Test](#)

Achievo cuts product QA and testing time by 40% while slashing costs.

www.schmids.com VISIT US TODAY!

[Top 5 QA Testing Tools](#)

2009 Top 5 QA Testing Tools Vendor Rankings. Get Free Report
Business-Software.com/QATesting

Latest From the I

Automation blogs are one of the greatest sources of automation information, so the Automated Testing Institute decided to keep you up-to-date with some of the best posts from around the web. Read below for some great posts, and keep an eye out, because you never know when you will be spotlighted.

Blog Name: Bug Huntress

Post Date: Jan 9, 2014

Post Title: 6 Tools for Web Software Functionality QA

Author: Bug Huntress

Though the importance of manual testing of a product in real world conditions, also known as in-the-wild testing, is constantly stressed, there is also a significant need in automation testing tools. Whether to do it manually, or use tool for functional testing is up to you and your project's format.

Presented below is the list of 6 helpful test tools you are definitely going to like...

Read More at:

<http://blog.bughuntress.com/automated-testing/6-tools-for-web-software-functionality-qa>

Blog Name: AgileWay

Post Date: Nov 13, 2013

Post Title: Quality web application super fast

Author: AgileWay

I talked and wrote a lot about test automation and continuous integration, which are the key reasons behind the success of some big names such as Facebook and LinkedIn. (“FaceBook pushes two releases per day”; LinkedIn joined to this elite club by “luring” Kevin Scott, the senior vice president of engineering and longtime Google veteran, “completely overhauled how LinkedIn develops and ships new updates”...

Read More at:

http://itest2.com/articles/2013/11/03/TestAutomation_CI_Secrets/

Blogosphere

ces of up-to-date test
sting Institute has
the latest blog
ome interesting
know when your post



Blog Name: DZone
Post Date: Dec 30, 2013

Post Title: Speed up BDD with Parallel Cucumber
Author: Shashikant Jagtap

First of all, if you are using Cucumber as a BDD tool and having large feature set, then you are at the right place. Now we have an awesome Ruby Gem called ‘Parallel_tests’ which is going to make our life much easier by parallelising large set of feature to run much faster. You can read more ‘Parallel_tests’ on GitHub. In this short tutorial, we will see how to use ‘Parallel_tests’ in our project...

Read More at:

<http://shashikantjagtap.net/running-cucumber-features-in-parallel-and-aggregating-reports/>

Blog Name: cognifide
Post Date: N/A

Post Title: Master Chef's choice: from spaghetti to cucumber
Author: Łukasz Morawski

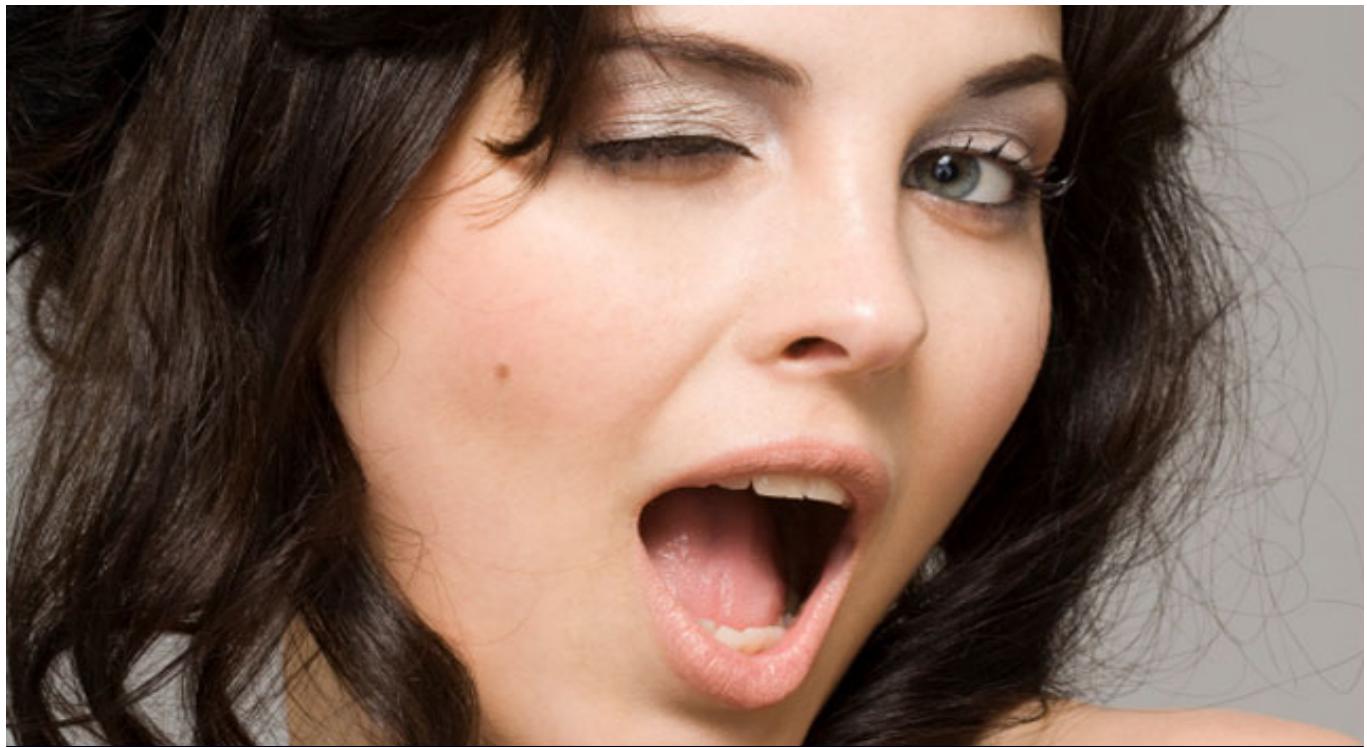
Implementing automated tests is something that everybody wants to do. If you ask any tester – test automation is her/his aim. It is the golden target that every tester aims for, but only a few of them take pains to assess the required knowledge, being under the illusion that a programming language or expensive tool will suffice to cope with all the problems that are likely to arise. This is not true. Writing good automated tests is much harder than that...

Read More at:

<http://www.cognifide.com/blogs/quality-assurance/evolution-of-test-automation-frameworks-master-chefs-choice-from-spaghetti-to-cucumber/>

Temptation.gov

Learning from the Healthcare.gov Launch



Talk about a crash course in software testing. A couple of months ago, most of America had probably never heard of software testing. Well, after the salacious scandal known as the Healthcare.gov website launch, they heard of it now. People got **caught trying to cheat** on their product delivery, and now everywhere you turn, you hear about testing. Check out some of the **Quotes** and media **Headlines** that have been swirling about.

Q

"The system didn't receive adequate end-to-end testing." (2)

- Andrew Slavitt of Optum

...as soon as a cute little project with a sexy budget and nice firm schedule comes along again, all of their new found testing morals will go right out of the window.

Q

"Due to a compressed time frame, the system just wasn't tested enough" (1)

- Julie Bataille, the Centers for Medicaid and Medicare Services (CMS)

Q

"The system didn't receive adequate end-to-end testing." (2)

- Andrew Slavitt of Optum

Q

A “compressed timeline” made adequate testing impossible. (4)

- Julie Bataille, the Centers for Medicaid and Medicare Services (CMS)

Q

“This website did not go through proper security testing before it went live on October 1”(5)

- “Technology Experts

H

“Obama official says HealthCare.gov not tested for October 1 volumes” (9)

- Yahoo News

H

“Team Obama Never Finished Testing HealthCare.Gov Before Launching It” (6)

- CBS News

H

“Full testing of HealthCare.gov began too late, contractors say” (7)

- WashingtonPost

H

“Emails show little testing on HealthCare.gov” (8)

- Politico

There's always enough testing until... there isn't. Had Healthcare.gov not run into so many massive production problems, the software testing discipline would still be under the national radar, and there would probably be a team of officials at this very moment trying to figure out how they could shave money and time out of the testing budget for the next go 'round.

All of the technology people involved in this disastrous launch that are now running around touting the need for more testing are equivalent to a husband running around saying sorry after he has been caught cheating on his wife (or vice versa). Sure they *say* they're sorry and see the error in their ways, but are they really sorry for what they've done or are they sorry that they got caught? The Healthcare.gov people will likely behave properly for a little while, but as soon as a cute little project with a sexy budget and nice firm schedule comes along again, all of their new found testing morals will go right out of the window.

If they want to show they've learned their lessons, let's hear about how they won't simply test within some arbitrary schedule, or test to find defects. Let's hear about how they will work to increase test coverage and focus their testing on the reduction of risks to reasonable levels. Let's hear about their investment in test planning and test automation.

Software testing is often overlooked to the chagrin of many that work in the discipline. And the more it is overlooked by the project team, the more of a spotlight testing will have shown on it for the end users in production to see. Many projects

don't give testing the focus it needs, and get away with it. The question that projects have to ask, however, is:

“Is it worth the risk?”

In hindsight, the Healthcare.gov team would probably say “No.”

REFERENCES

1. <http://www.cnn.com/2013/10/24/politics/congress-obamacare-website/>
2. <http://www.cnn.com/2013/10/24/politics/congress-obamacare-website/>
3. <http://swampland.time.com/2013/10/24/404-accountability-not-found-contractors-blame-feds-for-obamacare-snafu/>
4. <http://swampland.time.com/2013/10/24/404-accountability-not-found-contractors-blame-feds-for-obamacare-snafu/>
5. <http://www.cbsnews.com/news/obamacare-website-security-tests-never-finished-before-launch/>
6. <http://townhall.com/tipsheet/danieldoherty/2013/11/05/cbs-news-team-obama-never-finished-testing-healthcaregov-before-launch-date-n1737851>
7. http://www.washingtonpost.com/politics/house-panel-grills-contractors-on-troubled-health-insurance-web-site/2013/10/24/8f42c748-3ca7-11e3-b7ba-503fb5822c3e_story.html
8. <http://www.politico.com/story/2013/11/emails-show-little-testing-on-healthcaregov-100238.html>
9. <http://news.yahoo.com/obama-official-says-healthcare-gov-not-tested-october-143041564--sector.html>

TIRED OF SEARCHES THAT RETURN IRRELEVANT INFORMATION?



GOOGLE Automation - AUTOMATED TESTING SEARCH ENGINE

The Automated Testing Institute has partnered with Google to create a software test automation search engine. If you're looking for software test automation information, you will get the best results by using this search engine, because it only searches the sites that matter most to automators. You can also restrict searches to blogs and forum sites.

<http://www.googleautomation.com>

TRAINING THAT'S PROCESS FOCUSED, YET HANDS ON
Software Test Automation Training

WWW.TRAINING.AUTOMATEDTESTINGINSTITUTE.COM

PUBLIC COURSES

Software Test Automation Foundations
Automated Test Development & Scripting
Designing an Automated Test Framework
Advanced Automated Test Framework Development
Mobile Application Testing & Tools

VIRTUAL COURSES

Automated Test Development & Scripting
Designing an Automated Test Framework
Advanced Automated Test Framework Development
Mobile Application Testing & Tools

Come
participate in a set
of test automation courses
that address both fundamental
and advanced concepts from a
theoretical and hands on perspective.
These courses focus on topics such
as test scripting concepts, automated
framework creation, ROI calculations
and more. In addition, these courses
may be used to prepare for the
TABOK Certification
exam.

Public and Virtual Training Available

TESTKIT CONFERENCE 2014

Testing & Test Automation Conference

TestKIT

Test Automation
Conference

The D



TestKIT

TESTING & AUTOMATION CONFERENCE



Save the Date!

MARCH 3 - 5, 2014
CROWNE PLAZA HOTEL
ARLINGTON, VA

The KIT is Komining... Back

www.testkitconference.com