Avoiding regressions in an agile development environment

At Yottaa

Speaker Intro

- Studied Computer Science at Helsinki University
- Previously, Consulting Engineer at DEC
  - Developed middleware for OLTP systems
- Founder of Automature
  - built tools for making QA more transparent
- Currently a senior member of Quality Engineering team at Yottaa

Topics

- What does Yottaa do?
- The quality scene at Yottaa
- Tools used for automation at Yottaa
- The quality process in action...
- How tests are executed, metrics gathered, and reported
- A quick walk-through of the management portal UI test automation
- Questions

What Yottaa does...

eCommerce Acceleration platform

Enables online retailers significantly improve website performance

- Customers are well known eCommerce sites worldwide
- **No code modifications required** for customer sites!
- Sites are optimized by choosing a set of optimization policies/rules
- Traffic optimized **in-transit** through techniques including
  - Caching, Image optimizing, HTML rewriting, In-lining, Compression
- All traffic is routed through Yottaa's “data-centers”
- An average of 30% **reduction in page load times** is achieved
- Significant boosts in “conversion rates”

The software scene at Yottaa

- Production is a SaaS environment
- Distributed Linux based architecture, consisting of
  - Management Portal (Ruby)
  - Load balancers (Java, C)
  - Traffic optimizers (Java)
  - Change broadcasters (Java, GoLang)
  - Traffic analytics (Java)
- Build & deployment uses Ant, Groovy & Jenkins CI
- SW repositories on private GitHub
- Puppet used for deployment

QA Automation Tools

- All tests are automated...
- Regression Tests are executed as Jenkins jobs, kicked off after components have been deployed in CI environment
- Automation has been developed using
  - Java for unit tests
  - Python for web-services testing
  - Automature's Test Generator for UI (Selenium) tests
  - Ruby for Analytics testing
- Automated Tests run as Jenkins jobs, using Python, Ruby, Spark runtime environments
- Test Results are uploaded to Zermatt using Jenkins plugins

Some Terminology

- What is a Test Plan?
  - A Collection of Test Cases, bound by a common theme, serving a specified purpose, targeted at a specific Topology Set
- What is a Topology Set?
  - A collection of machines (physical or virtual) that mimic a deployment environment
- What is a Test Cycle?
  - An instance of execution of a Test Plan on target topology sets

Reporting using Jenkins Plugins

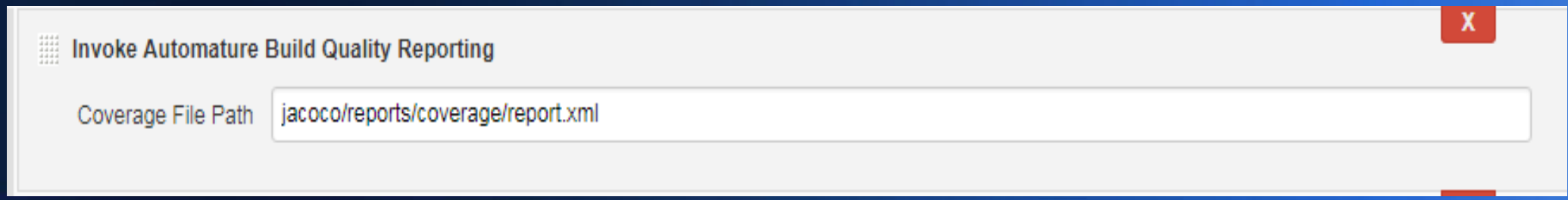
- Results of test cases/test suites executed, organized by
  - topology sets
  - date/time
  - release/sprint/build/plan
- Code coverage by test plan (or collection of plans)
- Build Quality Index Computation (scale 1-10, lowest is 1)
  - Function of tests passing & coverage achieved

Jenkins Plugin for Reporting Results

```
<testsuite errors="" failures="0" name="httpsrewrite" skips="" tests="4" time="645">
        <testcase classname="case-plt-511" name="httpsrewrite check https rewrite support - enabled, https -&gt; http" time="196" />
        <testcase classname="case-plt-513" name="httpsrewrite check https rewrite support - enabled, send over http, no rewrites expected" time="269" />
        <testcase classname="case-plt-510" name="httpsrewrite check https rewrite support - enabled, http -&gt; https" time="53" />
        <testcase classname="case-plt-512" name="httpsrewrite check https rewrite support - enabled, send over https, no rewrites expected" time="127" />
</testsuite>
```

Jenkins Coverage Reporting Plugin

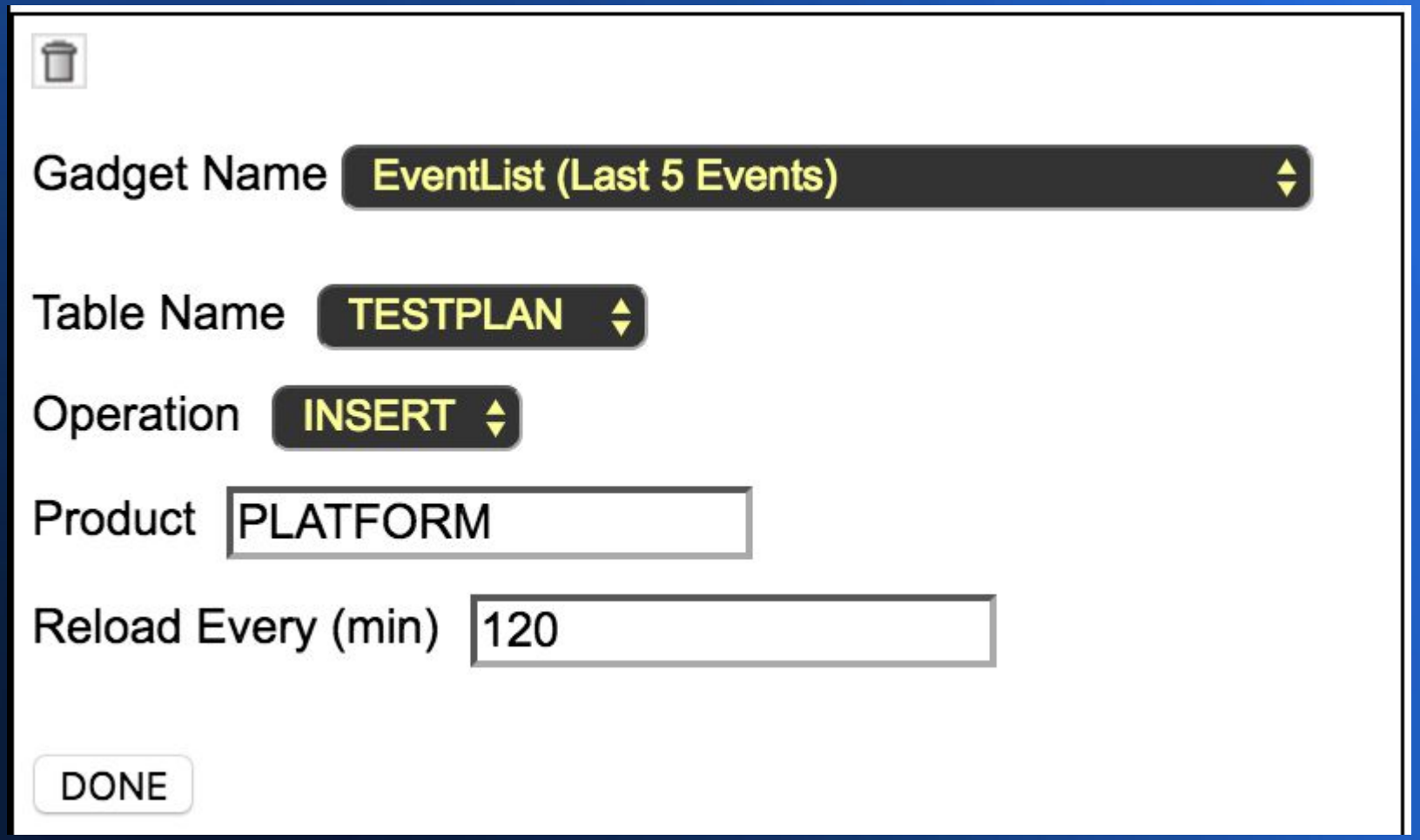
```
<package name="com/blah/common/ta/rc/dao">
 <class name="com/blah/common/ta/rc/dao/PageCategoryRow">
   <method name="getSiteId" desc="()V" line="20">
       <counter type="INSTRUCTION" missed="3" covered="0"/>
       <counter type="LINE" missed="1" covered="0"/>
       <counter type="COMPLEXITY" missed="1" covered="0"/>
       <counter type="METHOD" missed="1" covered="0"/>
   </method>
   <method name="toString" desc="()Ljava/lang/String;" line="27">
       <counter type="INSTRUCTION" missed="33" covered="0"/>
       <counter type="LINE" missed="6" covered="0"/>
       <counter type="COMPLEXITY" missed="1" covered="0"/>
       <counter type="METHOD" missed="1" covered="0"/>
   </method>
</package>
```



Displaying QA Results & Metrics

- Unit & Integration Tests are executed as Jenkins jobs
- Test results are written in XML (using Xunit schema)
- Automature's Jenkins Plugin uploads results to Zermatt, provides contextual information (which build, where & when executed)
- Zermatt provides many pre-built configurable custom charts, called gadgets, for displaying quality metrics
- Multiple Gadgets are combined into Wallboards
- Multiple wallboards are sequenced into a movie show

Creating a gadget



The Events List Gadget

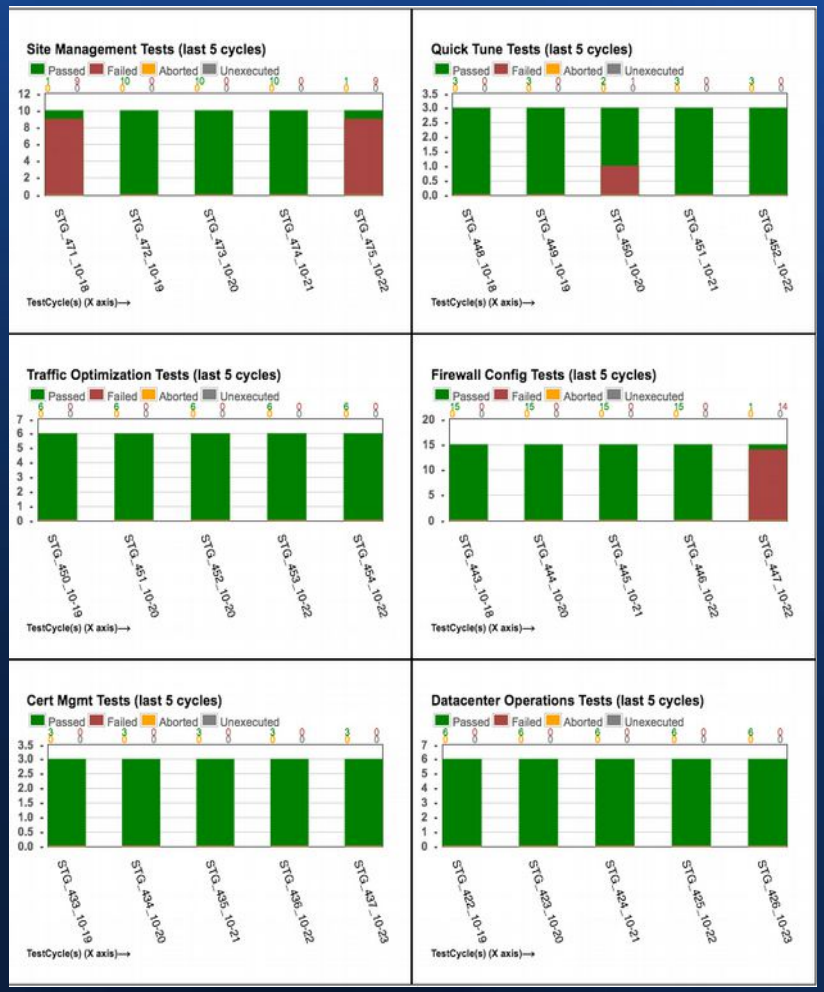
The Coverage Rankings Gadget

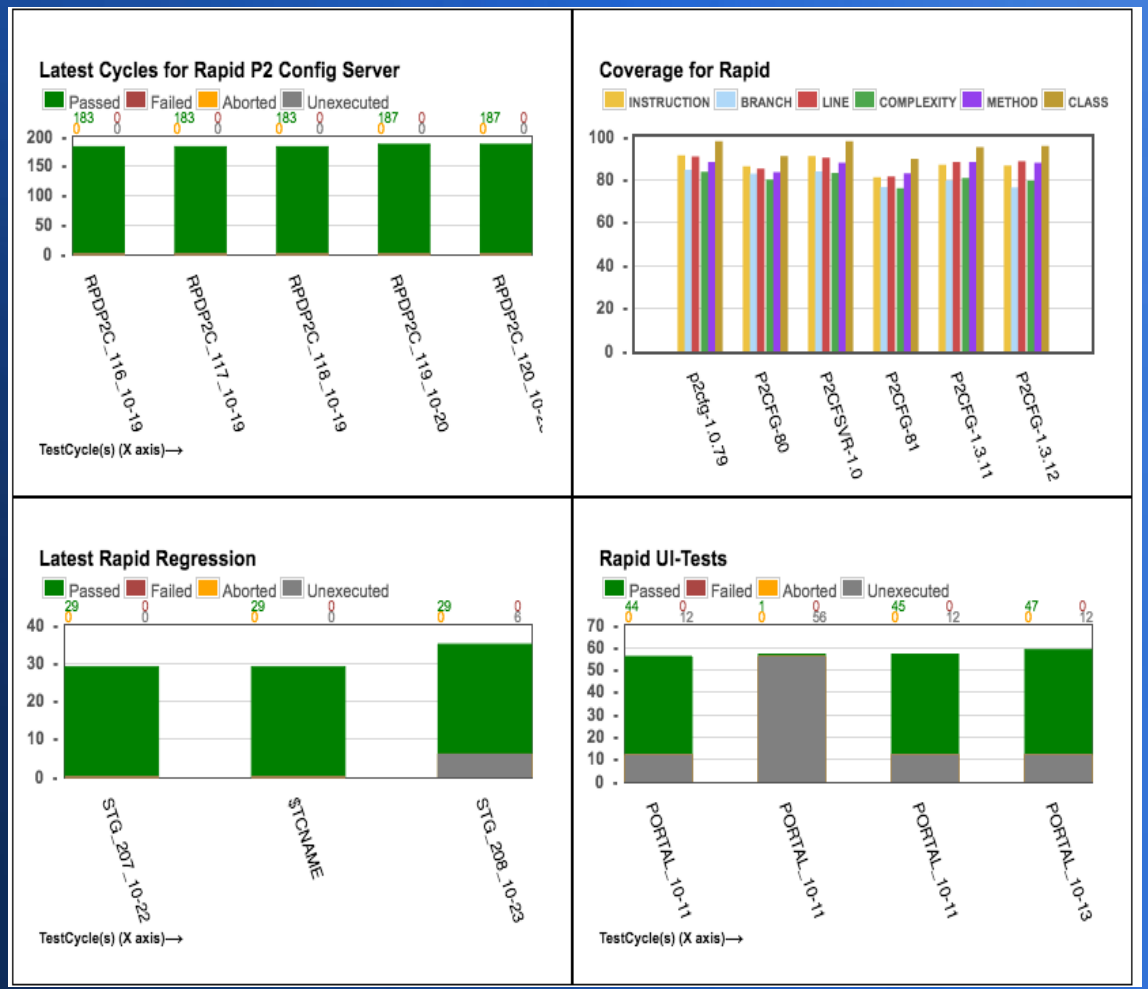
Alt Coverage Rankings for Platform TPU

PACKAGE	CLASS COVERED PCT ASC	METHOD MISSED DESC	LINE MISSED DESC
com/yottaa/platform/router/tpu/instanton	C : 0 , M : 13 , CP : 0.0%	C : 0 , M : 104 , CP : 0.0%	C : 0 , M : 440 , CP : 0.0%
com/yottaa/platform/router/store/zkcache	C : 0 , M : 7 , CP : 0.0%	C : 0 , M : 59 , CP : 0.0%	C : 0 , M : 308 , CP : 0.0%
com/yottaa/cloud/api/aws/network	C : 0 , M : 10 , CP : 0.0%	C : 0 , M : 47 , CP : 0.0%	C : 0 , M : 508 , CP : 0.0%
com/yottaa/cloud/api/aws/ec2/instances	C : 0 , M : 31 , CP : 0.0%	C : 0 , M : 41 , CP : 0.0%	C : 0 , M : 138 , CP : 0.0%
com/yottaa/cloud/api/core/object	C : 0 , M : 4 , CP : 0.0%	C : 0 , M : 38 , CP : 0.0%	C : 0 , M : 115 , CP : 0.0%
com/yottaa/platform/router/cache/local/diskbackendstore	C : 0 , M : 4 , CP : 0.0%	C : 0 , M : 32 , CP : 0.0%	C : 0 , M : 59 , CP : 0.0%
com/yottaa/cloud/api/aws/sqs	C : 0 , M : 27 , CP : 0.0%	C : 0 , M : 28 , CP : 0.0%	C : 0 , M : 68 , CP : 0.0%
com/yottaa/cloud/api/aws/s3/objects	C : 0 , M : 14 , CP : 0.0%	C : 0 , M : 26 , CP : 0.0%	C : 0 , M : 55 , CP : 0.0%
com/yottaa/platform/router/tpu/sprite/internal	C : 0 , M : 6 , CP : 0.0%	C : 0 , M : 23 , CP : 0.0%	C : 0 , M : 138 , CP : 0.0%
com/yottaa/platform/router/interfaces/timing/client	C : 0 , M : 5 , CP : 0.0%	C : 0 , M : 22 , CP : 0.0%	C : 0 , M : 140 , CP : 0.0%
com/yottaa/platform/router/interfaces/keepalive	C : 0 , M : 3 , CP : 0.0%	C : 0 , M : 21 , CP : 0.0%	C : 0 , M : 121 , CP : 0.0%
com/yottaa/platform/router/cache/local/disk	C : 0 , M : 1 , CP : 0.0%	C : 0 , M : 20 , CP : 0.0%	C : 0 , M : 21 , CP : 0.0%
com/yottaa/platform/router/backplane/services/logging	C : 0 , M : 5 , CP : 0.0%	C : 0 , M : 19 , CP : 0.0%	C : 0 , M : 100 , CP : 0.0%
com/yottaa/cloud/api/aws/elb	C : 0 , M : 15 , CP : 0.0%	C : 0 , M : 18 , CP : 0.0%	C : 0 , M : 54 , CP : 0.0%
com/yottaa/cloud/api/aws/s3	C : 0 , M : 5 , CP : 0.0%	C : 0 , M : 17 , CP : 0.0%	C : 0 , M : 124 , CP : 0.0%
com/yottaa/platform/mdp/exception	C : 0 , M : 10 , CP : 0.0%	C : 0 , M : 15 , CP : 0.0%	C : 0 , M : 40 , CP : 0.0%

Wallboard for Build Quality Index

Wallboards for Regression Results, Code Coverage (by Testplan)





A quick walk-through of the management portal UI test automation

- Web UI to manage the Yottaa Universe
- Provision & manage Yottaa's network SaaS infrastructure
  - 5 worldwide data centers (Asia, Europe & Americas)
  - multi-layer routing for load balancing & optimization
- Configure client sites
  - fine-tune content optimization strategies & rules
  - manage cache
- Monitor performance
- Developed using Ruby-on-Rails

How are test cases created?

- Automature's **Basel** Web Testing Framework
- **Interactive test generator for Automature's Spark**
  - Two phase approach
    - Phase 1: Generate Page Objects for web elements
    - Phase 2: Create Test Cases
      - sequence of steps to manipulate web elements
- What are Page Codes?
  - JSON objects, that uniquely identify web elements by name
  - May also describe a meta-element, e.g.
    - A grid structure, or
    - A sequence dependency,
      - e.g. grid remains hidden, until check-box is selected, making the frame interactable

What do test cases look like?

A Test Case

TestCase ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
Comment							
CDNAndCacheControl				&YottaaYourSiteOptimization.ClickOnCDNAndCacheControl	handle=%BrowserHandle%		
				Zbrowser.findLatestWindowHandle	BrowserHandle		
				&YottaaCDNandCacheControl.ClickOnGeneral	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.ClickOnAddException	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.PopulateAll	handle=%BrowserHandle%	csvfilename=$CSV_YottaaCDNandCacheControl	rowno=$RowInYottaaCDNandCacheControl_2
				&YottaaCDNandCacheControl.ClickOnAssetCachingDone	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.ClickOnMobile	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.ClickOnAddException	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.PopulateAll	handle=%BrowserHandle%	csvfilename=$CSV_YottaaCDNandCacheControl	rowno=$RowInYottaaCDNandCacheControl_3
				&YottaaCDNandCacheControl.ClickOnAssetCachingDone	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.ClickOnTablet	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.ClickOnAddException	handle=%BrowserHandle%		
				&YottaaCDNandCacheControl.PopulateAll	handle=%BrowserHandle%	csvfilename=$CSV_YottaaCDNandCacheControl	rowno=$RowInYottaaCDNandCacheControl_4
				&YottaaCDNandCacheControl.ClickOnAssetCachingDone	handle=%BrowserHandle%		

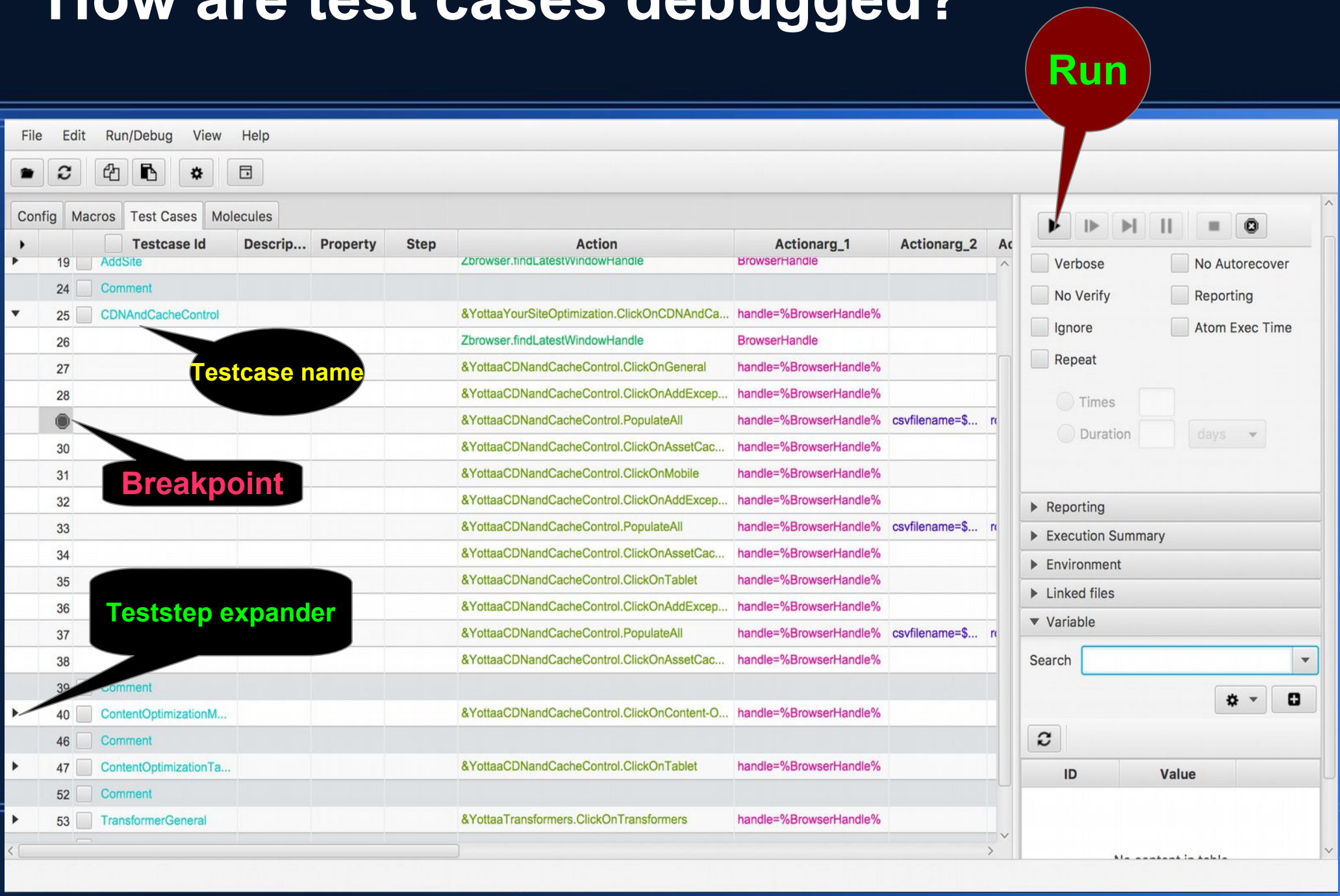
A Molecule

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4
Comment								
VerifyElementsAreReadOnly				#define_args	handle	elements		
				setvar	lookup			
				ZFileOps.ReadFileContent	$lookup|json	lookup		
				Zbrowser.VerifyElementsAreReadOnly	#handle	#elements	lookup	
Comment								
ClickOnForwardButton				#define_args	handle			
				Zbrowser.forward	#handle			
Comment								
ClickOnBackButton				#define_args	handle			
				Zbrowser.back	#handle			
Comment								
TakeSnapshot				#define_args	handle			
				Zbrowser.takeSnapshot	#handle			
Comment								
ReadGeneratedElements				#define_args	handle	jsonpath		
				Zbrowser.readGeneratedElement	#handle	#jsonpath		
Comment								
ClickOnUsage				#define_args	handle			
				SetVar	CV_ElementValue			
		silent		&GetPageCodeOfElement	Name	Usage	YottaaCDNandCacheControl.lookup	CV_ElementValue
				Zbrowser.clickPageCodeElement	#handle	Usage	CV_ElementValue	

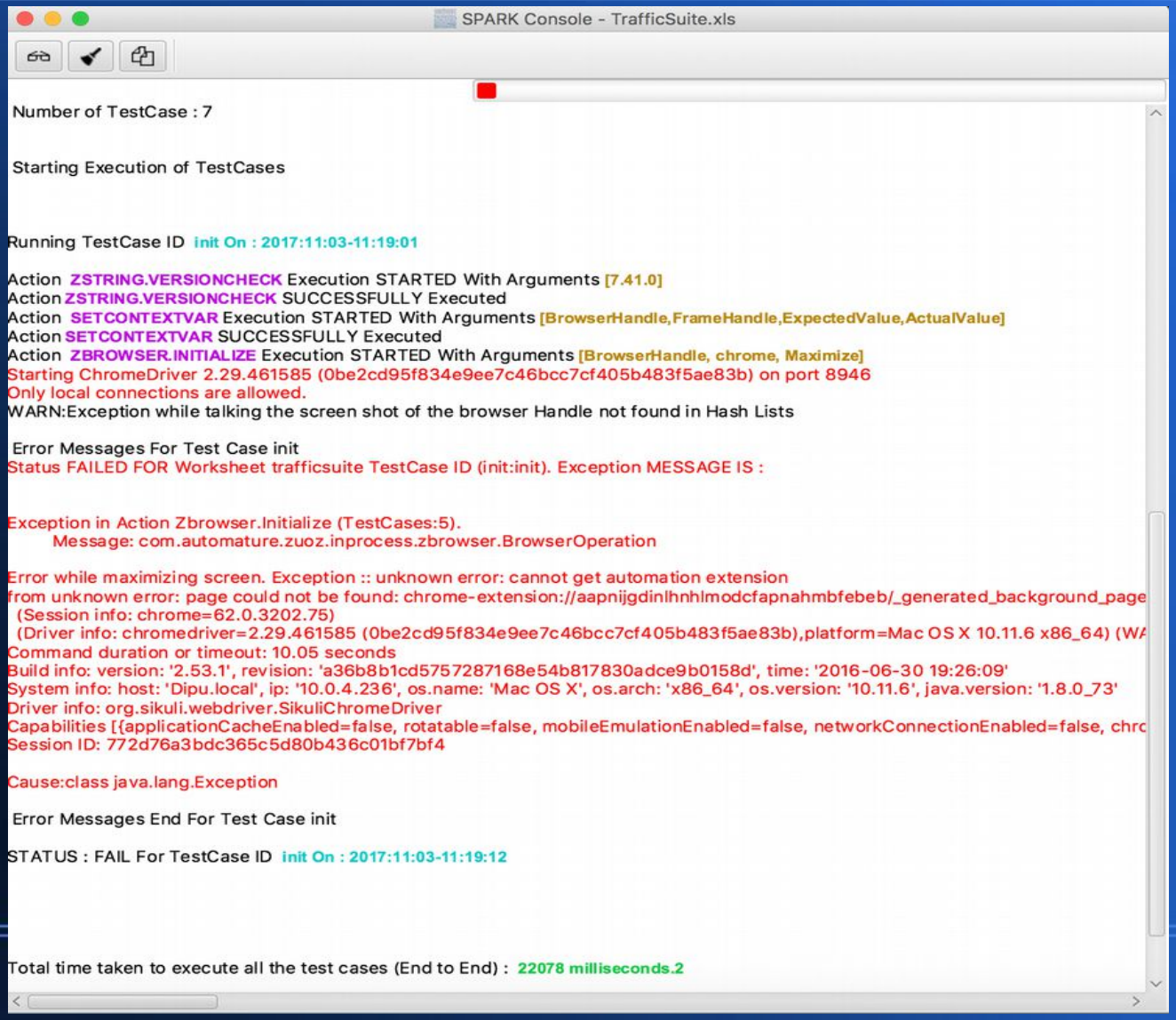
How are test cases executed?

- Through the command line
- Through Jenkins shell executor (Mac, Linux or Windows)
- Through Spark's Graphical Execution Environment/Debugger

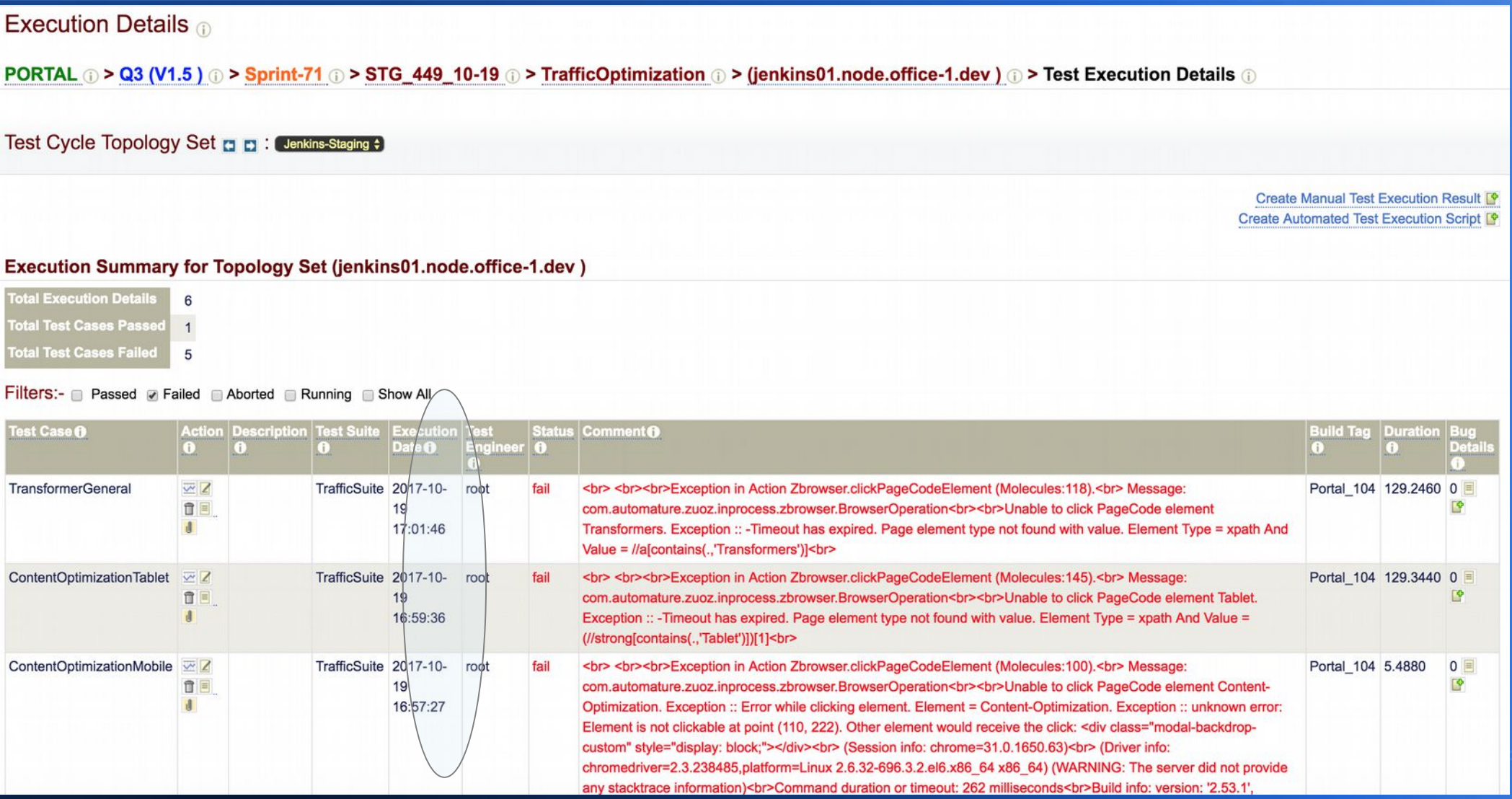
How are test cases debugged?



How do test case execution logs look like?



What a test execution report looks like?



Execution Details

PORTAL > Q3 (V1.5) > Sprint-71 > STG_449_10-19 > TrafficOptimization > (jenkins01.node.office-1.dev) > Test Execution Details

Test Cycle Topology Set : Jenkins-Staging

Create Manual Test Execution Result
Create Automated Test Execution Script

Execution Summary for Topology Set (jenkins01.node.office-1.dev)

Total Execution Details	6
Total Test Cases Passed	1
Total Test Cases Failed	5

Filters:- Passed Failed Aborted Running Show All

Test Case	Action	Description	Test Suite	Execution Date	Test Engineer	Status	Comment	Build Tag	Duration	Bug Details
TransformerGeneral			TrafficSuite	2017-10-19 17:01:46	root	fail	 Exception in Action Zbrowser.clickPageCodeElement (Molecules:118). Message: com.automature.zuoz.inprocess.zbrowser.BrowserOperation Unable to click PageCode element Transformers. Exception :: -Timeout has expired. Page element type not found with value. Element Type = xpath And Value = //a[contains(.,'Transformers')] 	Portal_104	129.2460	0
ContentOptimizationTablet			TrafficSuite	2017-10-19 16:59:36	root	fail	 Exception in Action Zbrowser.clickPageCodeElement (Molecules:145). Message: com.automature.zuoz.inprocess.zbrowser.BrowserOperation Unable to click PageCode element Tablet. Exception :: -Timeout has expired. Page element type not found with value. Element Type = xpath And Value = (//strong[contains(.,'Tablet')])[1] 	Portal_104	129.3440	0
ContentOptimizationMobile			TrafficSuite	2017-10-19 16:57:27	root	fail	 Exception in Action Zbrowser.clickPageCodeElement (Molecules:100). Message: com.automature.zuoz.inprocess.zbrowser.BrowserOperation Unable to click PageCode element Content-Optimization. Exception :: Error while clicking element. Element = Content-Optimization. Exception :: unknown error: Element is not clickable at point (110, 222). Other element would receive the click: (Session info: chrome=31.0.1650.63) (Driver info: chromedriver=2.3.238485,platform=Linux 2.6.32-696.3.2.el6.x86_64 x86_64) (WARNING: The server did not provide any stacktrace information) Command duration or timeout: 262 milliseconds Build info: version: '2.53.1',	Portal_104	5.4880	0

What a test plan looks like?

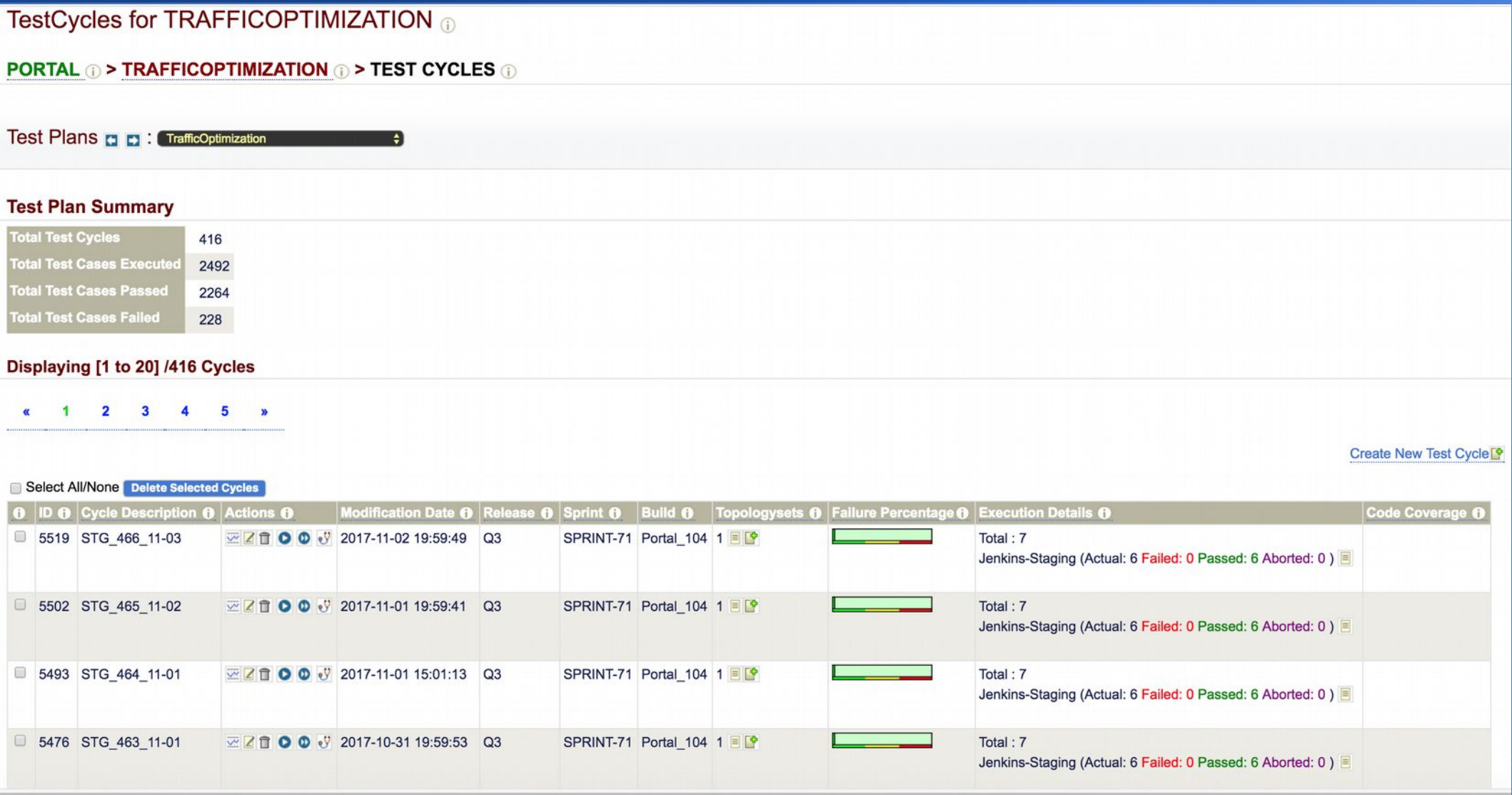
CONTAINER : Smoke

Create New Testplan

Select All/None | Move Selected Plans | Delete Selected Plans

Plan Name	Actions	Purpose	Execution Option	Status	Acceptance Criteria	Milestones	Last Execution Date	Schedule	Test Cycles	Tags	Test Suites	Topology Sets	Resources	Email List
DatacenterOperations		Smoke Testing on commonly used workflows in Datacenter Operations		Active	Sprint Default	Sprint Default	2017-11-02		324	0	1	3	0	
CertificateManagement		Smoke Testing on commonly used workflows in managing certificates		Active	Sprint Default	Sprint Default	2017-11-02		429	0	1	3	0	
FirewallConfiguration		Smoke Testing on commonly used workflows in configuring firewall and proxy		Active	Sprint Default	Sprint Default	2017-11-02		419	0	1	3	0	
TrafficOptimization		Smoke Testing on commonly used workflows in optimizing traffic		Active	Sprint Default	Sprint Default	2017-11-02		416	0	1	3	0	*NULL*
QuickTuneRules		Smoke Testing on commonly used workflows in Tuning Sites		Active	Sprint Default	Sprint Default	2017-11-02		414	0	1	3	0	
SiteManagement		Smoke Testing on commonly used workflows in managing SIte s		Active	Sprint Default	Sprint Default	2017-11-02		427	0	1	3	0	

What a test cycle report looks like?



TestCycles for TRAFFICOPTIMIZATION

PORTAL > TRAFFICOPTIMIZATION > TEST CYCLES

Test Plans : TrafficOptimization

Test Plan Summary

Total Test Cycles	416
Total Test Cases Executed	2492
Total Test Cases Passed	2264
Total Test Cases Failed	228

Displaying [1 to 20] /416 Cycles

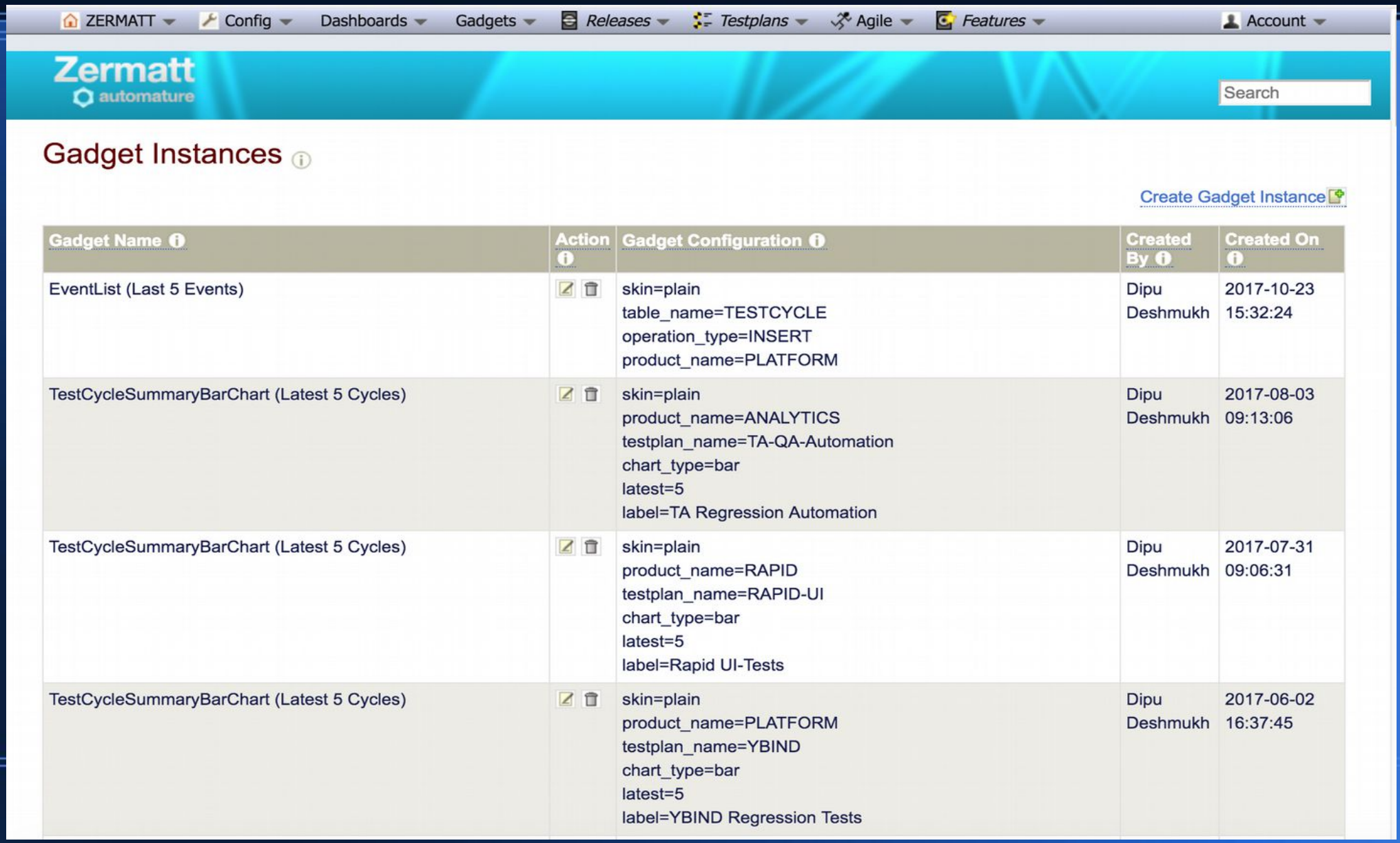
« 1 2 3 4 5 »

Create New Test Cycle

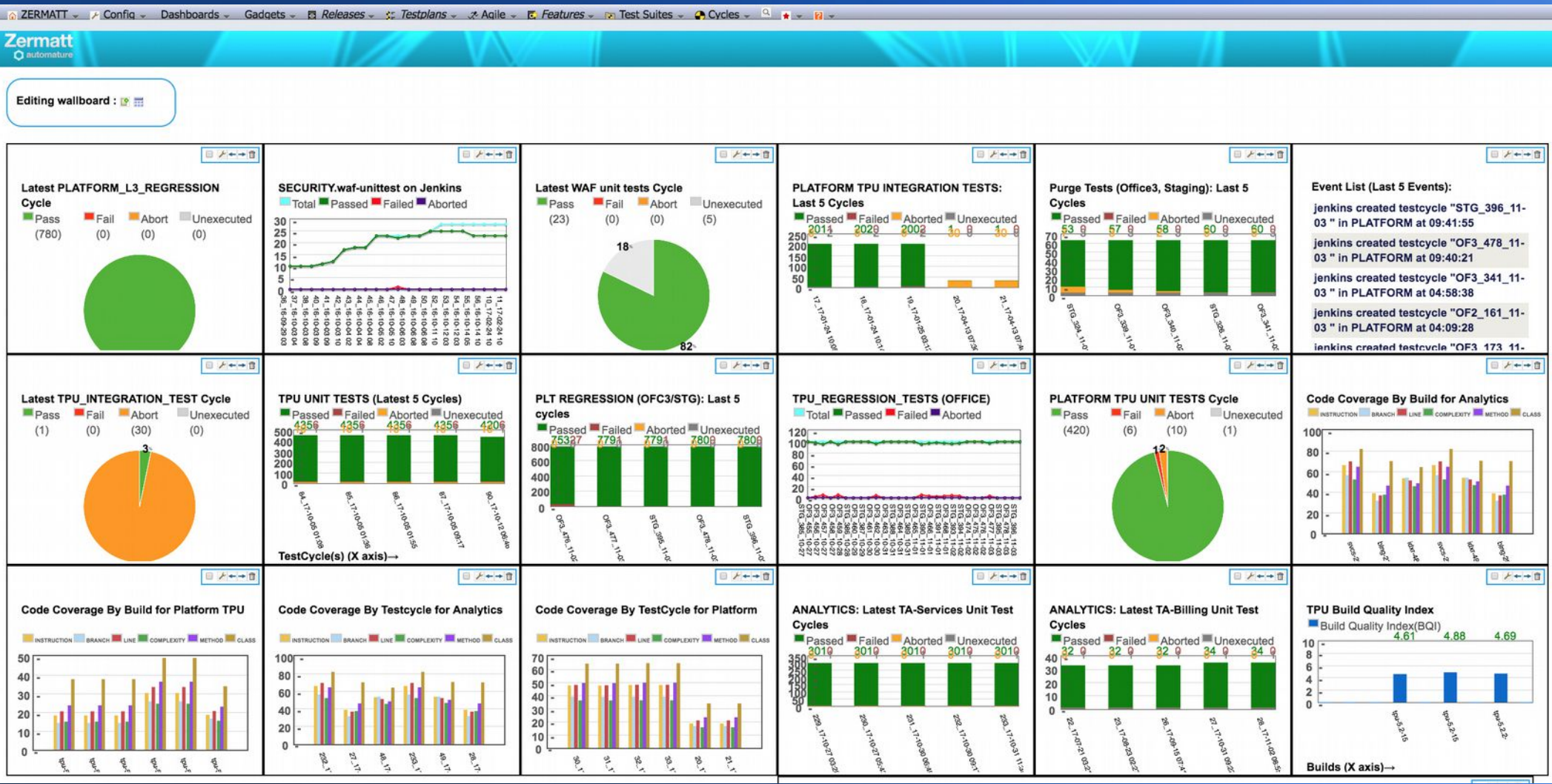
Select All/None Delete Selected Cycles

ID	Cycle Description	Actions	Modification Date	Release	Sprint	Build	Topologysets	Failure Percentage	Execution Details	Code Coverage
5519	STG_466_11-03		2017-11-02 19:59:49	Q3	SPRINT-71	Portal_104	1		Total : 7 Jenkins-Staging (Actual: 6 Failed: 0 Passed: 6 Aborted: 0)	
5502	STG_465_11-02		2017-11-01 19:59:41	Q3	SPRINT-71	Portal_104	1		Total : 7 Jenkins-Staging (Actual: 6 Failed: 0 Passed: 6 Aborted: 0)	
5493	STG_464_11-01		2017-11-01 15:01:13	Q3	SPRINT-71	Portal_104	1		Total : 7 Jenkins-Staging (Actual: 6 Failed: 0 Passed: 6 Aborted: 0)	
5476	STG_463_11-01		2017-10-31 19:59:53	Q3	SPRINT-71	Portal_104	1		Total : 7 Jenkins-Staging (Actual: 6 Failed: 0 Passed: 6 Aborted: 0)	

How gadget instances are created?



Wallboards are created by choosing gadgets



Thank You!

Questions?