Test Challenges and Approaches With SaaS and PaaS

Dr. Ganesh Neelakanta Iyer
Principal QA Engineer
Progress Software
About Me

- Completed B.Tech. in Computer Science and Engineering from Mahatma Gandhi University, India in 2004 with University First Rank
- Five years of Industry work experience in Bangalore (2004–2007) and HYD (2012–till date), India
- Finished Masters and Ph.D. from National University of Singapore in 2008 and 2012 respectively
- **Research interests**: Cloud computing, Game theory, Wireless Networks, Pricing, Software Quality Analysis
- **Personal Interests**: Kathakali, Teaching, Traveling, Photography, Cooking
- **Website**: http://ganeshniyer.com
- Currently Principal Engineer at Progress Software, India
Agenda

- Challenges Introduced by Cloud for Testing
- Cloud Test Dimensions
- Integration Testing & Test Automation
- Overview of an In-House Web UI Automation Framework
- Conclusions
Introduction

1. Web-driven applications
2. On-demand resource availability
3. Faster time to market
4. Reduced capital and operational expenses
Overview of Cloud Delivery Models…

<table>
<thead>
<tr>
<th>Cloud Service Models</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software as a Service (SaaS)</td>
<td></td>
</tr>
<tr>
<td>Platform as a Service (PaaS)</td>
<td></td>
</tr>
<tr>
<td>Infrastructure as a Service (IaaS)</td>
<td></td>
</tr>
</tbody>
</table>
Software Test Challenges With Cloud

**Paradigm Shift**
- Web-driven SaaS applications are getting more popular
- Everything happens in few mouse clicks

**Seamless Upgrades**
- With Cloud, upgrade should happen live with minimal or no down time
- Test the upgrade process of software products seamlessly

**Sharing of Resources: Multi-Tenancy**
- The resources used for software development as well as deployment might be in publically shared resources
- Need for having specific testing for such systems such as multi-tenant penetration testing
Cloud Test Dimensions
Cloud Test Dimensions in a Nut-shell

- Elasticity Testing
- Security Testing
- Per/HA Testing
- Compatibility Testing
- API Integration Testing
- Multi-Tenancy Testing
Elasticity Testing

<table>
<thead>
<tr>
<th>Elasticity Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource acquisition/release time</td>
</tr>
<tr>
<td>Provisioning on-the-go</td>
</tr>
<tr>
<td>Table text load testing for ELB</td>
</tr>
</tbody>
</table>

Cloud App Development Platform

- Limitations on max objects/applications at a time
- Number of applications that can be executed per platform instance
- 100’s of administrators accessing the management console

SaaS Application Development
PaaS Capabilities
Security Testing

Automation:
- Websecurify
- ZAP tool
- Scripts

All Areas of SaaS and PaaS
High Availability (HA) and Performance Testing

Automation:
- NeoLoad
- JMeter

HA & Performance
- Time to deploy
- Multi-tenancy
- Time to genesis
- Connectivity and reliability with 3rd parties
- Reliability and availability
- Latency

HA/Performance Testing
- Reliability
- Latency
- Availability
- Stress

All Areas of SaaS and PaaS
Compatibility Testing

**Compatibility Testing**

- Accessibility testing
- Globalization and localization testing
- Compatibility under different situations

**Automation:**
- Sahi
- Selenium

**All Areas of SaaS and PaaS**
API Integration Testing

**Automation:**
- SoapUI
- NeoLoad

**API Integration**
- Connectivity and invocation testing
- API load testing
- API security
- Multi-tenancy

SaaS Apps Which Uses APIs
Multi-Tenancy Testing

<table>
<thead>
<tr>
<th>Multi-Tenancy</th>
<th>Automation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-tenant penetration</td>
<td>• UI frameworks</td>
</tr>
<tr>
<td>Rigid failure containment</td>
<td>• Hybrid frameworks</td>
</tr>
<tr>
<td>Availability and business continuity</td>
<td></td>
</tr>
<tr>
<td>Risk of correlated behaviors</td>
<td></td>
</tr>
<tr>
<td>Service transition activity analysis</td>
<td></td>
</tr>
</tbody>
</table>

SaaS Apps for ISVs / PaaS
Cloud Test Dimensions in a Nut Shell

<table>
<thead>
<tr>
<th>Elasticity Testing</th>
<th>Security Testing</th>
<th>HA &amp; Performance</th>
<th>Compatibility</th>
<th>API Integration</th>
<th>Multi-Tenancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource acquisition/release time</td>
<td>Traversal vulnerability</td>
<td>Time to deploy</td>
<td>Accessibility testing</td>
<td>Connectivity and invocation testing</td>
<td>Multi-tenant penetration</td>
</tr>
<tr>
<td>Provisioning on-the-go</td>
<td>User access/roles</td>
<td>Multi-tenancy</td>
<td>Globalization and localization testing</td>
<td>API load testing</td>
<td>Rigid failure containment</td>
</tr>
<tr>
<td>Table text load testing for ELB</td>
<td>Identity federation management</td>
<td>Time to genesis</td>
<td>Compatibility under different situations</td>
<td>API security</td>
<td>Availability and business continuity</td>
</tr>
<tr>
<td>Communication latency over SSL</td>
<td>Connectivity and reliability with 3rd parties</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-tenant penetration</td>
<td>Reliability and availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Integration Testing &
Test Automation
Integration Test Challenges and Approaches

Integration Test automation challenges

- Multiple systems behave differently
- Some systems do not allow automated deletion of test data
- Unpredictable delays in updating various systems.
- Different types of environments for testing
  - Perform web UI testing and runtime testing at once
Integration Test Automation With Cloud

- The outer harness is the runtime automation framework and the inner harness is the UI automation framework.
- Switching between runtime component and UI component testing.
- Combined test results and test logging.

Sahi as the UI framework
Java + TestNG as the Runtime framework
### Automation Frameworks

<table>
<thead>
<tr>
<th>Testing Type</th>
<th>Tools/Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Testing – (Web UI)</td>
<td>Sahi / Selinium scripts</td>
</tr>
<tr>
<td></td>
<td>Sahi based in-house test framework – WaFeR</td>
</tr>
<tr>
<td>Performance Testing</td>
<td>Neoload</td>
</tr>
<tr>
<td></td>
<td>JMeter</td>
</tr>
<tr>
<td>API Integration Testing</td>
<td>SOAP UI</td>
</tr>
<tr>
<td>Integration Testing</td>
<td>Test-NG + Java + Sahi combined approach</td>
</tr>
</tbody>
</table>
In-House Flow-Driven UI Test Automation Framework – WaFeR
Motivation

- **Emergence of Cloud**
  - SaaS: Several Web-driven Applications
  - PaaS: Web platform to build SaaS
- **Presence of dynamic objects in the UI**
- **Database-driven applications**
  - Table-based output screens
  - Dynamic URLs get generated with database sequence
- **Frequent changes to UI in Agile environment**
  - Changes can be expected on every sprint

**Requirement for extensible flow-driven test framework to automate web-application testing**
Progress Products That Use This Framework for Web UI Testing

Data Model | Workflow & Logic Model | User Interface Model | Permissions Model

Organization Model | Integration Model | Application Directory | Hybrid Mobile App

PROGRESS® Rollbase®

PROGRESS® DataDirect Cloud®
Proposed Flow-Driven Framework: Three Step Flow

1. Record test in the preferred User Interface (UI) testing tool (e.g. Sahi, Selenium)
2. Identify properties of the test flow (e.g. buttons, tabs etc.)
3. Test lines for testing the web application under consideration gets identified
4. Add pre assertions and post assertions for different test lines identified above
Proposed Automation Framework: Components

- Excel templates to define test suites
- Test Data (excel templates) is called from here

Built in components of the framework

Generic object names are defined here. It allows us to update object names at one place
e.g. _USERNAME

Excel Template is used to define all actions and related assertions for a given test

Test Result Repo

Test Execution Module

Test Data

Framework library (main script) which initiates the test

Driver Script

Driver Data

Web UI Testing tool

Generic Libraries

Product Specific Libraries

Extended Framework

Product-specific libraries and properties

Properties
Welcome to Pacific
Come on in. The water is warm!

Username: gaiyer
Password: ********

I forgot my Username
I forgot my Password

Log In

Don’t have a Pacific Account? Register now.
Example Test Case
Example Test Case
Example Test Case
Sample Script File for Previous Scenario

<table>
<thead>
<tr>
<th>SeqID</th>
<th>Flow</th>
<th>ObjectType</th>
<th>ObjectName</th>
<th>Data</th>
<th>GotoAction</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACT</td>
<td>link</td>
<td>VAR_RB_URL</td>
<td></td>
<td>navigateTo</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ACT</td>
<td>function</td>
<td>fncRBLogin</td>
<td><em>RB_USERNAME</em></td>
<td>exec</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ASST</td>
<td>RTData</td>
<td>fncRBLoginRV</td>
<td>true</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ACT</td>
<td>link</td>
<td>New Application</td>
<td></td>
<td>click</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ACT</td>
<td>link</td>
<td>Guide Me Through It</td>
<td></td>
<td>click</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ACT</td>
<td>textbox</td>
<td>txtAppName</td>
<td>ExchangeApp</td>
<td>setValue</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ACT</td>
<td>link</td>
<td>Next</td>
<td></td>
<td>click</td>
<td></td>
</tr>
</tbody>
</table>
Advantages of the Proposed Model

- Considerable improvement in test development
- Works seamlessly with existing test scripts
- Extensible to testing any web-driven/Cloud-based applications
- Entire framework can be deployed and executed from any platform
  - Local test machine
  - Cloud platforms such as AWS
- Reduced effort and manual intervention for maintenance
Conclusions

- Cloud gives unique opportunities and challenges for QA
- New test dimensions
  - Security, Multi-tenancy, etc.
- Faster test execution
  - Agile Testing
  - Leveraging automation tools
  - New automation frameworks
Visit the Resource Portal

- Get **session details & presentation downloads**
- Complete a **survey**
- Access the latest Progress **product literature**

www.progress.com/exchange2014