3 tips to move from performance testing to performance engineering

Mukulika Kapas, Silvia Siqueira & Vicky Villalobos

August 12, 2015
Today’s Speakers

Silvia Siqueira
Senior Product Marketing Manager
HP Software
Today’s Speakers

Mukulika Kapas
Senior Product Manager
HP Software
Today’s Speakers

Vicky Villalobos
Product Manager
HP Software
Housekeeping

- This “LIVE” session is being recorded
  Recordings are available to all Vivit members

- **Session Q&A:**
  Please type questions in the Questions Pane
Webinar Control Panel

Toggle View Window between Full screen/window mode.

Questions
“Working software is the primary measure of progress.”

Principles behind the Agile Manifesto
What is your release cycle?

- Quarterly
- Monthly
- Weekly
- Less than a week
How often do you run performance tests?

- Once per release
- More than once per release
- Once across multiple release
- Every sprint
What is the average time that you have for performance testing?

- Less than a week
- 2 weeks
- A month
- More than a month
Performance engineering is the key to meeting your business demands...
Simple apps but still complex to test for performance

TIME to execute the test

User experience
The journey…
From performance testing to performance engineering

Performance testing

When: Late Stage
Why: Risk-assessment
Who: Quality Assurance
What: Load Testing

Performance engineering

When: Continuous
Why: Business improvement
Who: DevTest, DevOps
What: App Optimization

We help our customers outperform their user experience!
Evolution to Performance Engineering

High performance value delivered

**Load Testing**
Take down your servers

**Performance Testing**
User performance

**Performance Tuning**
Continuous testing

**Lifecycle Virtualization**
Virtualize your dependencies

- SLA based
- Wide range of technologies
- Client side performance
- Correlated with load testing
- Network conditions underestimated
- Analyze operations data to realistically address performance profiles
- Eliminate 3rd party dependencies
- Adjust to modern challenges with location-aware applications

**Performance Engineering**
High performance value delivered

- Continuous business feedback and improvement
- Built-in and automated performance
- Optimized applications for business and customer value
- Collaborative and interactive team focused on quality

© Copyright 2015 Vivit Worldwide
3 tips to move from performance testing to performance engineering
Tip #1
Accelerate Dev Test by continuous testing

Shift left and include performance tests
Helping Dev/Test load test in Continuous Integration (CI): HP LoadRunner provides developers with tools and APIs to infuse performance testing earlier

- **Build load** test scripts during development integrating with Visual Studio or Eclipse
- **Run Unit tests** in Load Scenarios
  - Support for nUnit, jUnit and Selenium tests as part of LoadRunner or Performance Center scenarios
  - Add performance tests as build steps in Jenkins
- **PC REST API** to support Continuous Delivery processes
Continuous testing / Accelerate Dev Test
Integration with Dev tools – Unit test

Add LoadRunner APIs to Unit Tests in Visual Studio or Eclipse

Execute Unit Tests using the LoadRunner Engine

Verify Output within the IDE

Create LoadRunner scenario from the IDE

Integration with Jenkins: Automate test scheduling, execute tests and view results as part of the build plan
Continuous testing / Accelerate Dev Test
Integration with Dev tools – Unit test

Add LoadRunner APIs to Unit Tests in Visual Studio or Eclipse

Execute Unit Tests using the LoadRunner Engine

Verify Output within the IDE

Create LoadRunner scenario from the IDE

Integration with Jenkins: Automate test scheduling, execute tests and view results as part of the build plan
Continuous testing / Accelerate Dev Test
Integration with Dev tools – Unit test

Value: Accelerate application delivery by integrating performance testing in Agile and hybrid environments

© Copyright 2015 Vivit Worldwide
Continuous testing / Accelerate Dev Test
Integration with Dev tools – Unit test

Add LoadRunner APIs to Unit Tests in Visual Studio or Eclipse
Execute Unit Tests using the LoadRunner Engine
Verify Output within the IDE
Create LoadRunner scenario from the IDE
Integration with Jenkins: Automate test scheduling, execute tests and view results as part of the build plan

Value: Accelerate application delivery by integrating performance testing in Agile and hybrid environments

Available only on LoadRunner
Continuous testing / Accelerate Dev Test

Integration with Dev tools – Jenkins

Value: Accelerate application delivery by integrating performance testing in Agile and hybrid environments.

Available with LoadRunner and Performance Center

Add LoadRunner APIs to Unit Tests in Visual Studio or Eclipse

Execute Unit Tests using the LoadRunner Engine

Verify Output within the IDE

Create LoadRunner scenario from the IDE

Integration with Jenkins: Automate test scheduling, execute tests and view results as part of the build plan
Continuous testing / Open APIs

- REST API to support Continuous Delivery processes:
  Schedule, Run, Stop, Collate, Analyze tests via REST API
  Create Monitors, Topologies
  Connect to Continuous Integration solutions

Available only with Performance Center

NOTE: all product views are illustrations and might not represent actual product screen shots
Simple – Single TruClient Protocol

One script to test them all...

This is a rolling (up to 3 year) roadmap and is subject to change without notice

Agile/Continuous/DevOps

All product views are illustrations and might not represent actual product screens
Simplified end user experience

Performance and usability enhancements

- New Performance Center Dashboard
- Topology/Timeslot view and updated Run-time settings
- Chrome and Safari support in Performance Center
- New PC Reports - Test trending, protocol granularity
- Performance Improvements - VuGen correlations and Performance Center uploads
- Documentation Enhancements
Tip #2
Realistic testing using production feedback
How do we do performance test intelligently?

80% of people use only 20% of the application
Compare tests with real production flows

Stage One: Collect production metrics
- Configure & collect logs and metrics in production and compare with test
Compare load test to production

Fix testing discrepancies

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Scripts</th>
<th>Hits Distribution [%]</th>
<th>Total</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy_Cart_Mapped</td>
<td>generate_transactions</td>
<td>5.35</td>
<td>4.97% (-1.22%)</td>
<td>14.29 (15.71%)</td>
</tr>
<tr>
<td>Add2Cart_Mapped</td>
<td>generate_transactions</td>
<td>4.68</td>
<td>7.04% (-0.33%)</td>
<td>14.29 (10.09%)</td>
</tr>
<tr>
<td>Logon4Form_store_Mapped</td>
<td>generate_transactions</td>
<td>11.06</td>
<td>11.03% (-0.33%)</td>
<td>14.29 (20.87%)</td>
</tr>
<tr>
<td>Logon_to_store_Mapped</td>
<td>generate_transactions</td>
<td>11.06</td>
<td>11.03% (-0.33%)</td>
<td>14.29 (20.87%)</td>
</tr>
<tr>
<td>Search_page_food_Mapped</td>
<td>generate_transactions</td>
<td>22.13</td>
<td>21.99% (-0.64%)</td>
<td>14.29 (-35.06%)</td>
</tr>
<tr>
<td>Search_page_toys_Mapped</td>
<td>generate_transactions</td>
<td>28.58</td>
<td>27 (1.50%)</td>
<td>4.29 (-47.08%)</td>
</tr>
<tr>
<td>Search_page_pets_Mapped</td>
<td>generate_transactions</td>
<td>17.1</td>
<td>16.97% (-0.72%)</td>
<td>14.29 (-15.97%)</td>
</tr>
</tbody>
</table>
Automatically generate tests based real user behavior

Stage Two: Automatically create tests

- Run data mining and analytics algorithms to identify common user flows, urls, peak and average user loads, geographical distribution and client types
- Automatically generate test scenarios, scripts, network virtualization profiles

All product views are illustrations and might not represent actual product screens
Tip#3
Root cause analysis
Early performance testing
Focus on single user performance

Sprint 5

- **C1**: User story: For example, XYZ component response times should be <10 ms
- **G1**: User story: For example, ABC component response times should be <20 ms
- **C2**: User story: For example, GUI elements response times should be <100 ms
- **G2**: User story: For example, business process response times should be <100 ms
- **C3**
- **B1**: User story: For example, business process response times should be <100 ms

Legend:
- **C**: Component (non-GUI)
- **G**: GUI element
- **B**: Business process
- **Stable**
Early performance testing
Focus on single user performance

Slowest performing methods

Application performance optimization using HP Diagnostics
Application monitoring under load
Network virtualization + performance

- Local User (No NV)
- WiFi (Remote Branch)
- 4G
- 3G

<table>
<thead>
<tr>
<th>Activity</th>
<th>Local User</th>
<th>WiFi</th>
<th>4G</th>
<th>3G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch App</td>
<td>1.2</td>
<td>3.2</td>
<td>6.3</td>
<td>12.8</td>
</tr>
<tr>
<td>Log In</td>
<td>1.3</td>
<td>5.2</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>View Account</td>
<td>4.3</td>
<td>7.4</td>
<td>12.3</td>
<td>22</td>
</tr>
<tr>
<td>Sign Out</td>
<td>1</td>
<td>1</td>
<td>2.3</td>
<td>2.8</td>
</tr>
</tbody>
</table>
Native mobile app support

- Integrate with HP Mobile Center
- Record & Replay using TruClient directly on your device
- Load your application using LoadRunner and test for single-user experience with TC Native Mobile
Tip #1  Accelerate dev test by continuous testing  
        Shift left and include performance tests

Tip #2  Realistic testing using production

Tip #3  Root cause analysis / feedback
Evolution to performance engineering

High performance value delivered

- SLA based
- Wide range of technologies
- Client side performance
- Correlated with load testing
- Network conditions underestimated
- Analyze operations to realistically address performance profiles
- Eliminate 3rd party dependencies
- Adjust to modern challenges with location-aware applications
- Continuous business feedback and improvement
- Built-in and automated performance
- Optimized applications for business and customer value
- Collaborative and interactive team focused on quality

Load Testing
Take down your servers

Performance Testing
User performance

Performance Tuning
Continuous testing

Lifecycle Virtualization
Virtualize your dependencies

Today

Upgrade to newer versions, Performance Center or SaaS
Add Diagnostics, Network Virtualization, Service Virtualization
## HP performance testing solutions

### Performance testing solutions

- **StormRunner Load**
- **LoadRunner**
- **Performance Center on-premise**
  - Delivered on SaaS

**Monitor with SiteScope**

### Profiler solution to pinpoint root cause

- **Diagnostics**

### Detect network behavior

- **Network Virtualization**

### Virtualize test environment

- **Service Virtualization**
Why HP performance testing?

Large installed base
10,000+ customers

More platforms supported
50+ protocols

Comprehensive and affordable
Price reduced: no more controller fee, pay as you go, VUDs as low as 0.56 cents

Size and staying power
#17 Fortune 500
300K staff

Largest community
Millions of users WW

TruClient Technology
Unique, easy and fast to simulate transactions and test

Top analyst rankings
Gartner Group and IDC application test reports

Network Virtualization
Exclusive offering
4 FREE offerings…

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HP StormRunner Load Free Trial</strong></td>
<td><a href="https://saas.hp.com/signup/try/stormrunner-load">https://saas.hp.com/signup/try/stormrunner-load</a></td>
</tr>
</tbody>
</table>
Thank you

Fill out the survey or send us an email to request a Demo of the next release & Roadmap*

Email: vicky.villalobos@hp.com

*Under NDA
Additional Resources

To compare Performance Center products and select the right trials to meet your current business needs go to:


Also see the Application performance engineering e-book at:

Thank you

• Complete the short survey and opt-in for more information from HP Software.

All attendees that opt-in will have a chance to win an HP Stream!

www.hp.com

www.vivit-worldwide.org