"Performance and Security Testing in Agile Development"

Presented by:

Tracy DeDore
Hewlett-Packard

Brought to you by:
Tracy DeDore
Hewlett-Packard

Tracy DeDore is the Agile Solution Marketing Manager for HP’s BTO Software Application organization. She is a twenty-four year veteran of HP and has held a variety of software-related roles including software programming, 2nd level software support and software product marketing, as well as ten years managing network and mobility solution development and marketing.
Incorporating Performance and Security Testing in Agile Development

June 9, 2010

Tracy DeDore

Agenda

• The promise of Agile
• Agile delivery challenges
• Why QA is critical to the success of Agile
• Agile testing challenges
• Manual and automated testing
• Change-friendly component testing framework
• Performance testing in every sprint
• Service virtualization to remove dependencies and constraints
• Security testing in every sprint
• Summary
The promise of Agile

- Accelerate time-to-market
- Reduce costs
- Increase project success rates without sacrificing quality

Agile delivery

To realize the promise of Agile, it must move from a development practice to a delivery practice

- Developers
- Project Managers
- Business Analysts
- QA
- Performance Engineers
- Security Specialists/Penetration Testers
Key Agile management challenges

- Manage agile projects across all stakeholders
- Remove latency from Dev and QA processes
- Maximize collaboration across the Agile team
- Enable discovery, sharing, and re-use of key assets and artifacts to accelerate work
- Provide real-time visibility into tasks and progress to speed issue resolution and improve decisions

The challenges of Agile delivery

- “Will the business benefit from Agile?”
- “Why is Agile not working for us?”
- “Developers go one way, QA another. How do we better collaborate?”
- “No more point solutions. I need a unified vision.”
- “We’ll go Agile, even if no one else does.”
- “Just get rid of the process overhead and let us work!”
- “Requirements, QA – these guys don’t get it.”
- “Where do we fit?”
  - “We need a simpler way to capture reqts.”
  - “I know change is part of Agile, but how do I know the right features made it in?”
- “It’s not clear what Agile means for us.”
- “Some think Agile means ‘No more QA!’”
- “We're building in sprints, but still testing at the end.”
- “We're building so fast we have no time to performance test.”
  - “I don’t know if performance is improving or getting worse with each sprint.”
  - “How can we test earlier, when the environment isn’t even in place?”
- “We’re building so fast we have no time to performance test.”
- “I don’t know if performance is improving or getting worse with each sprint.”
- “How can we test earlier, when the environment isn’t even in place?”
**Bottom line:**
Moving to Agile is a big change

- Radically changes Planning, Dev, and QA practices
- Roles blur and responsibilities change
- Collaboration and concurrency are paramount
- Requires more than process change
- Functional, performance and security testing baked into every iteration
- Automation is critical to success

---

**Why the QA Director (and team) are key to Agile success**

"You can never be agile without automated testing solutions."

Gartner AADI Conference, Dec. 09, Matt Hotle, VP Distinguished Analyst

"As companies move to Agile, and as portfolios shift from version 1 into maintenance, the coding costs may drop, but the cost to test the application continues to rise. Thus, it is key to find ways to reduce the costs to test software and drive automation."

SOA Testing: Confronting the Nightmare of Testing Shared Services”, Application Architecture, Development and Integration Conference, December 7-9, 2009
Thomas Murphy, Gartner Research Director
Agile delivery – as it should be

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQMT</td>
<td>RQMT</td>
<td>RQMT</td>
<td>RQMT</td>
<td>RQMT</td>
<td>RQMT</td>
</tr>
<tr>
<td>DESIGN</td>
<td>DESIGN</td>
<td>DESIGN</td>
<td>DESIGN</td>
<td>DESIGN</td>
<td>DESIGN</td>
</tr>
<tr>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
</tr>
<tr>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
</tr>
</tbody>
</table>

Time-boxed for focus | Hands-on with stakeholders | Surfaces issues sooner

Rigorous, cumulative testing | Designed for change | True measure of progress

Agile delivery – as it too often is

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQMT</td>
<td>RQMT</td>
<td>RQMT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
</tr>
<tr>
<td>DESIGN</td>
<td>DESIGN</td>
<td>DESIGN</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
</tr>
<tr>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>CODE/UT</td>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
<td>SYS. TEST</td>
</tr>
</tbody>
</table>

Time-boxed for focus | Hands-on with stakeholders | Surfaces issues sooner

Rigorous, cumulative testing | Designed for change | True measure of progress
Scrummerfall. *n.* The practice of combining Scrum and Waterfall together in a single project so as to ensure failure at a faster rate than with Waterfall alone.

**Why?**

Some reasons the Agile vision goes unrealized

- Unit testing mistaken for system test
- System test deferred, as in Waterfall
- Performance and security testing deferred, as in Waterfall
- Regression testing not seen as viable in “sprint-time”
- Difficult to coordinate geographically distributed project teams
- Organizational impacts under-estimated
Why?
Reasons the Agile vision goes unrealized

-- Unit testing is mistaken for system test

“Agile is still a relatively new topic in the realm of software testing... the role of the traditional software test and QA organization hasn’t been well-defined.”

Don’t Let Short-Term Agile Create Long-Term Pain, Gartner Apr. 09

underestimated

40% of unplanned downtime is caused by application failures, costing an average of $100k per hour for mission-critical apps

Gartner, 2008
The cost of poor quality

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>Typical cost of testing in a development project</td>
</tr>
<tr>
<td>56%</td>
<td>Number of defects introduced at the requirements phase</td>
</tr>
<tr>
<td>82%</td>
<td>Amount of effort required to fix poor requirements</td>
</tr>
<tr>
<td>1</td>
<td>The no1 leading cause of IT waste is poor defect mngt and rework</td>
</tr>
<tr>
<td>100x</td>
<td>Cost to repair a defect in production vs. requirements</td>
</tr>
</tbody>
</table>

The value of effective quality management

Source: NIST 2002 RTI Project 7007.011
## Agile testing challenges

- Early defect discovery
- Test the highest value / highest risk user stories
- Regression testing is critical; impossible without automation
- Support unit, service, integration, GUI, system, and business process testing in one platform
- Remove pervasive system dependencies/constraints that delay testing and increase environment costs
- Test complex, heterogeneous enterprise environments and modern technologies

## Customers need a Quality Management solution that enables their business objectives

- Eliminate rework and reduce cost across the enterprise
- Improve time to market
- Mitigate risk
Agile project management

Agile Dashboard and Reporting
- Burn-up/burn down charts
- Cross-sprint velocity charts
- Agile development progress dashboard
- Agile testing progress dashboard

Agile Development Management
- Risk based story and task management
- User story management
- Task management
- Effort estimation

Agile Project Planning
- Define and automate Agile processes
- Reduce and accelerate process
- Release / sprint / backlog management
- User story and task definition
- Resource Planning

Agile Quality Management
- Testing task management
- Test execution management
- Defect reporting and management
- Test metrics and Quality progress

Functional, performance, and security requirements

Release planning through the hardening sprint

Functional, performance, and security should be addressed starting with the release planning phase
Modern Applications + Agile = Challenge

Multiple components, multiple sprints, multiple moving parts

End to End Business Process Testing
(GUI / Business Layer / MIScld)

Application Level Testing
(GUI Testing, User Acceptance Testing)

Business Layer Testing
(Web Services, APIs, Components, Unit, Integration)

Manual testing must be organized and immediate

QA should be able to manually test faster within the iteration
Automate as much and as earlier as possible

Regression testing at the end of each iteration (or part of continuous build)

QA should be able to automate on the available application layer as early as possible

Traditional serial testing

Functional testing today

Create order test
- Login
- Create Order
- View Order
- Logout

Delete order test
- Login
- Search Order
- View Order
- Delete Order
- Logout

Update order test
- Login
- Search Order
- Update Order
- View Order
- Logout
Build a change-friendly component testing framework

Create once, reuse components in multiple tests

- Application architecture?
- New technologies: degrade performance, introduce risk?
- Upgrade the IT infrastructure?
- User interface design: degrade performance, introduce risk?
- Expected increase in usage: degrade performance?
- Test data needed to accelerate testing?
Effective user stories facilitate performance and security testing

- Involve performance and security engineers in user story review to identify areas of high risk

For example:
- data privacy
- high data throughput
- new technologies (Flash, Ajax…)
- high degree of customization to packaged apps
- architectural changes
- new components
- non-standard toolkits

- Prioritize performance and security requirements based on risk

Automated load testing

Replaces real users with thousands of “virtual” users
- Generates accurate, measurable and repeatable load on the system from a single point of control
- Pinpoints bottlenecks in the system
Performance Testing automation accelerates the testing cycle

Incorporate Performance Testing into every sprint

- Assess performance requirements during system architecture and release planning
- Define acceptance criteria for each performance user story
- Develop test cases/data in parallel with coding
- Virtualize dependent services and resources
- Deliver user stories to performance testers as signed off
- Ensure continuous feedback among all stakeholders
- Share performance test assets across projects
- Schedule performance tests to run off-hours
Performance Testing in Agile – HOW?

• Client-side performance awareness
• Monitor performance trends per iteration
• Monitor SLA performance per iteration
• Component/"Headless" performance testing
• Share/reuse test assets across iterations
• Store/connect test results to components

Pervasive dependencies and constraints can kill agility

System availability constraints delay testing efforts and increase test environment costs
How to get agility in constrained environments?

Combine functional and performance testing with service virtualization to simulate the realistic and dynamic behavior of dependent systems.

Before

After

Effective user stories facilitate performance and security testing

• Involve performance and security engineers in user story review to identify areas of high risk

For example:
• data privacy
• high data throughput
• new technologies (Flash, Ajax…)
• high degree of customization to packaged apps
• architectural changes
• new components
• non-standard toolkits

• Prioritize performance and security requirements based on risk
Security risks have never been greater

Attacks

Industrialized identity theft, and illicit information markets
Wide variety of regulations by industry and geography

Variety of mitigation methods under development (including compliance)
Regulations/compliance begin to come into force
New ones under development
Maturity Security Programs

Loose collaboration among groups

Individual gain

Internal measures

Wide variety of regulations by industry and geography

Industrialized identity theft, and illicit information markets

The approach

The costs of a security attack are huge

$202 \times 30,000 = 6.65 \text{ M}

Total average cost of a data breach per compromised record*
Average # of compromised records per breach^*
Average Total Cost per breach*

<table>
<thead>
<tr>
<th>Company</th>
<th>Records Affected*:</th>
<th>Cost:</th>
<th>Organization Cost:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Services</td>
<td>12,500,000</td>
<td>$202</td>
<td>$2.53 Billion</td>
</tr>
<tr>
<td>Government</td>
<td>25,000,000</td>
<td>$202</td>
<td>$5.05 Billion</td>
</tr>
<tr>
<td>Major Retailer</td>
<td>94,000,000</td>
<td>$202</td>
<td>$19 Billion</td>
</tr>
</tbody>
</table>

* Ponemon Institute, 2008 Annual Study: U.S. Cost of a Data Breach
^ The Open Security Foundation

No industry is exempt
Applications are the target

Applications: ✖️ Unprotected and ignored
Servers: ✔️ Protected by intrusion prevention air
Network: ✔️ Secured by firewall

From scans of 31,373 sites, over 85% showed a vulnerability that could give hackers the ability to read, modify and transmit sensitive data.

-- Web Application Security Consortium

Web security risks are complex

Security professionals are overwhelmed by applications

Organizations must learn to bridge the gap

Developers/QA are overwhelmed by security
Ingredients for application security success

Application Security is everyone’s responsibility
- Business, Development, QA & Security Teams
- Educate and empower

Build security in
- Repeatable & predictable
- Best practices
- Enterprise policies and standards

Enterprise Security Platform
- Automated solutions
- Built in security knowledge
- Communication

People

Process

Technology

Security Testing in Agile - HOW?

- Static source code analysis
- Step or path-specific business process testing
- Black box/dynamic Web Application Scanners
- Standard, repeatable security testing into standard QA practices
- Leverage automation tools
## Incorporating Performance and Security Testing in Agile Development

- Change Agile from a development practice to a delivery practice
- Build a change-friendly component framework
- Bake functional, performance and security testing into every iteration
- Automate as soon as possible
- Virtualize dependent systems/resources

## Q&A
For more information...

http://www.hp.com/go/agile

HP Agile Development Blog:  

Future of Testing Blog:  

Requirements Management: Blog  

LoadRunner/Performance Center Blog:  

Application Security Blog:  
http://www.communities.hp.com/securitysoftware/blogs/

Outcomes that matter.