Quality Everywhere
How quality fits in every step of the software process

UVA Software Testing
Fall 2018

Corey Vaudo
Senior Vice President, Quality Assurance

Conrad Rybka
Senior Quality Assurance Engineer
Agenda

Who we are

Quality Definition

Quality Everywhere

Q & A
Who we are
Applied Predictive Technologies

Mission
Help drive evidence-based business decisions

Clients
Deployed globally by over 300 organizations

Expertise
Combine business strategy, math, and large databases to inform decisions

Product
Cloud based software platform utilizing same underlying technology across industries
APT – The Concept

We enable business experimentation

---

**Step 1: Design a Test**
- Choose a group that is representative of the entire network so decision-makers can apply test findings broadly.
- Test the action with the select group.

**Step 2: Measure the Impact**
- Compare performance of the test group to a similar control group that didn’t receive the action to isolate the action’s true impact.
- Understand which variations of the program work best.

**Step 3: Optimize Rollout**
- Identify key factors driving performance.
- Build a predictive model to understand which groups will respond profitably.

---

APT Targeted Rollout: $12.1MM

Full Rollout: $7.8MM
APT – Business experiment examples

We tackle a variety of high stakes “experiments” across industries

Promotion Planning – is my promotion driving traffic or just eroding margin?

Menu Development – what entrees should I remove from my menu? What should I add?

Market Basket Profiling – How do different customers react to a promotion? How can we target promotions by customer segment?

Customer Incentive Programs – what rate should I offer on a CD to a new customer?

Network Planning – where should we build the next Holiday Inn?

Online-to-store Advertising - how much does online advertising affect my in-store sales? Do online sales cannibalize in-store sales?
Our approach requires expertise in numerous areas

**Big Data**
- Host over 500 individual SQLServer databases
- Host over 1 PB of data
- Conduct “sku” level analysis for the world’s largest retailers

**Analytics**
- Hold numerous patents for advanced analytic techniques
- Utilize both leading statistical packages and develop in-house algorithms

**Web Dev**
- Host software entirely over the web utilizing latest front-end technologies (e.g. Redux, React, etc)

**CI / CD**
- Deploy code multiple times per day
- Run tens of thousands of tests daily
- Utilize git, Jenkins, nUnit, Chef, Selenium & numerous other world-class “testing & deployment” technologies
Stereotypical approach to quality

Product Management team develops “requirements”

Engineers build software to “spec”

QA team confirms software is high quality

Software delivered to users
Quality Everywhere
Quality is built-in everywhere throughout the development process.
Quality is built-in everywhere throughout the development process.
Quality in the Requirements Process

Quality is built in the requirements process in multiple dimensions

- Prototyping
Quality in the Requirements Process

Quality is built in the requirements process in multiple dimensions

- Usability Sessions

Developer watching videotape of usability test.
Quality in the Requirements Process

Quality is built in the requirements process in multiple dimensions

- Requirements testing

<table>
<thead>
<tr>
<th>Priority</th>
<th>Item</th>
<th>Details / Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>I want to select a metric most relevant to my business question so that I can best answer it. (metric selection)</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>I want to select a specific timeframe so that I can focus on the relevant period for my business question. (timeframe selection)</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>I want to select a visualization that best displays the answer for my relevant business question. (visualization selection)</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>I want to summarize the metric and metric format used to construct an output so that I can quickly understand the underlying data for a given output. (metric summary)</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>I want to summarize the timeframe used to construct an output.</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>I want to choose a metric format that best answers my business question. (metric format selection)</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>I want to summarize the metric and metric format used to construct an output so that I can quickly understand the underlying data for a given output.</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>I want to choose whether to include a benchmark so that I can better contextualize my portfolio's performance. (benchmark toggle)</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>I want to choose one or more attributes to split my output so that I can analyze metric performance segmented by those attributes. (split selection)</td>
<td></td>
</tr>
</tbody>
</table>
Quality is built-in everywhere throughout development process
Quality is shared amongst all developers

Numerous developer practices drive software quality

- Pull Requests and Code reviews
Quality is shared amongst all developers

Numerous developer practices drive software quality

- Pair programming

“When Terry said we should consider pair programming, I don’t think she meant we had to take it this far.”
Quality is shared amongst all developers

Numerous developer practices drive software quality

- Test Driven Development
Quality is built-in everywhere throughout development process
Code is “integrated” every hour

“Continuous Integration” tools ensure all changes are “merged” and tested together frequently
Quality is built-in everywhere throughout development process.
Automated Regression Testing

When you fix one bug, you introduce several newer bugs
Automated Regression Testing

Manual test only once, but test automatically forever

Test Result

0 failures (±0)

All Tests

<table>
<thead>
<tr>
<th>Package</th>
<th>Duration</th>
<th>Fail</th>
<th>(diff) Skip</th>
<th>(diff) Pass</th>
<th>(diff) Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apt.Platform</td>
<td>77 ms</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Apt.Platform ABTesting.Tests.Client</td>
<td>1.9 sec</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Apt.Platform ABTesting.Tests.Client.Integration</td>
<td>0.42 sec</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Apt.Platform ABTesting.Tests.Core</td>
<td>47 ms</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Apt.Platform ABTesting.Tests.Web</td>
<td>4.2 sec</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Lines Of Code: 10,469
Files: 199
Directories: 87
Lines: 15,796

Functions: 599

Duplications: 0.4%

Unit Tests Coverage: 82.8%

Line Coverage: 82.8%
Quality is built-in everywhere throughout development process.
Configuration Management

Server consistency is a key driver of “software quality”
Quality is built-in everywhere throughout development process

Requirements Gathering

Development

Continuous Integration

Regression Testing

User Feedback & Reporting

Client Users

Internal Users

Analytic & Front-End Testing

Servers

Security Testing

Performance Testing

Integration Testing

Monitoring & Alerting
Vulnerability Scanning

Testing for security vulnerabilities is essential when dealing with sensitive data

### Checkmarx

<table>
<thead>
<tr>
<th>Scan Results (43 Risks)</th>
<th>Open Code Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High Risk Vulnerabilities (0)</strong></td>
<td></td>
</tr>
<tr>
<td>Reflected XSS All Clients</td>
<td>0</td>
</tr>
<tr>
<td>Client DOM XSS</td>
<td>0</td>
</tr>
<tr>
<td><strong>Medium Risk Vulnerabilities (13)</strong></td>
<td></td>
</tr>
<tr>
<td>HttpOnlyCookies In Config</td>
<td>0</td>
</tr>
<tr>
<td>No Request Validation</td>
<td>0</td>
</tr>
<tr>
<td>Client DOM XSRF</td>
<td>11</td>
</tr>
<tr>
<td>Client Cross Frame Scripting Attack</td>
<td>1</td>
</tr>
<tr>
<td>Parameter Tampering</td>
<td>1</td>
</tr>
<tr>
<td><strong>Low Risk Vulnerabilities (30)</strong></td>
<td></td>
</tr>
<tr>
<td>Missing X Frame Options</td>
<td>0</td>
</tr>
<tr>
<td>Client DOM Open Redirect</td>
<td>14</td>
</tr>
<tr>
<td>Client Insecure Randomness</td>
<td>6</td>
</tr>
<tr>
<td>Improper Exception Handling</td>
<td>4</td>
</tr>
<tr>
<td>Client Potential DOM Open Redirect</td>
<td>2</td>
</tr>
<tr>
<td>DebugEnabled</td>
<td>2</td>
</tr>
<tr>
<td>Off By One Error</td>
<td>1</td>
</tr>
<tr>
<td>Divide By Zero</td>
<td>1</td>
</tr>
<tr>
<td>CustomError</td>
<td>0</td>
</tr>
</tbody>
</table>

### Scan Details

- **Scan Start**: Saturday, September 9, 2017 1:08:22 PM
- **Scan Time**: 00h:18m:34s
- **Scan Type**: Full
- **Lines of Code**: 252089
- **File Count**: 1560
Quality is built-in everywhere throughout development process.
Performance testing

A functional application is only valuable if it is performant
Performance testing

A functional application is only valuable if it is performant.
Quality is built-in everywhere throughout development process
Front-end Integration Tests

Test against live versions of the application
Load Testing

Test against live versions of the application
Quality is built-in everywhere throughout development process

Requirements Gathering

Development

Continuous Integration

Regression Testing

User Feedback & Reporting

Client Users

Internal Users

Analytic & Front-End Testing

Servers

Security Testing

Performance Testing

Integration Testing

Monitoring & Alerting
Monitoring and Alerting

Extensive monitoring of production system minimizes downtime
Quality is built-in everywhere throughout development process.

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- User Feedback & Reporting
- Client Users
- Internal Users
- Analytic & Front-End Testing
- Servers
- Security Testing
- Performance Testing
- Integration Testing
- Monitoring & Alerting
Front-end & analytic testing

Extensive test planning and review lead to more effective testing
Front-end & analytic testing

Testing Blitzes allow for testing from multiple perspectives
Front-end & analytic testing

Analytic validation is essential to providing a trustworthy tool

### Analytic Test Plan

Math changes: Expectation, impact, and lift are calculated from the already aggregated top-level this and last year pre- and post-period values.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Tester</th>
<th>DB</th>
<th>Description</th>
<th>Expected</th>
<th>Analytic</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Alex Berkow</td>
<td>Regression</td>
<td>Normal met/cat</td>
<td>New analysis numbers lie out</td>
<td>No aggregate: <img src="https://testing.apptplatform.com/hweb/ActivityOutputs/7227?OutputsDashboardViewModel.OpenOutputId=45896" alt="Link" /> &lt;br&gt; Aggregate: <img src="https://testing.apptplatform.com/hweb/ActivityOutputs/7228?OutputsDashboardViewModel.OpenOutputId=47302" alt="Link" /> &lt;br&gt; Click here to expand...&lt;br&gt; SSLY analytic cases.xlsx</td>
</tr>
<tr>
<td>10</td>
<td>Alex Berkow</td>
<td>UDM Met/Met</td>
<td>New analysis numbers lie out</td>
<td>No aggregate: <img src="https://testing.apptplatform.com/hweb/ActivityOutputs/7300?OutputsDashboardViewModel.OpenOutputId=48491" alt="Link" /> &lt;br&gt; Aggregate: <img src="https://testing.apptplatform.com/hweb/ActivityOutputs/7345?OutputsDashboardViewModel.OpenOutputId=47613" alt="Link" /> &lt;br&gt; Click here to expand...&lt;br&gt; SSLY analytic cases.xlsx</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Alex Berkow</td>
<td>UDM Met/Met</td>
<td>New analysis numbers lie out</td>
<td>No aggregate: <img src="https://testing.apptplatform.com/hweb/ActivityOutputs/7407?OutputsDashboardViewModel.OpenOutputId=47911" alt="Link" /> &lt;br&gt; Aggregate: <img src="https://testing.apptplatform.com/hweb/ActivityOutputs/7408?OutputsDashboardViewModel.OpenOutputId=47910" alt="Link" /> &lt;br&gt; Click here to expand...&lt;br&gt; SSLY analytic cases.xlsx</td>
<td></td>
</tr>
</tbody>
</table>
Quality is built-in everywhere throughout development process

Requirements Gathering

Development

Continuous Integration

Regression Testing

User Feedback & Reporting

Client Users

Internal Users

Client Users

Analytic & Front-End Testing

Servers

Security Testing

Performance Testing

Integration Testing

Monitoring & Alerting
Front-end & analytic testing

Multiple deployment levels allows for internal user feedback

MMMM. I BELIEVE I HAVE DETECTED AN ISSUE WITH LARRY’S CODE.
Quality is built-in everywhere throughout the development process.
User Feedback and Reporting

Monitoring usage and collecting user feedback helps close the loop between development team and client.