Quality Everywhere

How quality fits in every step of the software development process

UVA Software Testing - Fall 2019
Corey Vaudo
Conrad Rybka
Agenda

Who We Are
Defining Quality
Quality Everywhere
Q & A
Who We Are
Mastercard Data & Services

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
Test & Learn

Enabling 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.

Test vs. Control Performance

Full Rollout: $7.8MM
APT Targeted Rollout: $12.1MM
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 Development**
- 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
Defining Quality
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.

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Servers
- Usage
- Analytic and Front-end Testing
- User Feedback & Reporting
Quality in the requirements process

Prototyping
Quality in the requirements process

Usability Sessions

Developer watching videotape of usability test.
## Quality in the requirements process

### Use Cases

Listing out the drilldown use cases, prioritized and grouped by similar/related actions. The priority is informed by the following aspects:

- frequency of use
- importance to the output generation
- dependence on other use cases - i.e. if one use case depends on the selection of another use case, it should generally be lower in priority - see Settings Hierarchy section below.

The priority can then be used to help inform both the placement of the features to achieve these use cases as well as the relative emphasis of each feature on the drilldown page.

<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 the development process:

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Servers
- Analytic and Front-end Testing
- Usage
- User Feedback & Reporting
Quality is shared amongst all developers

Pull Requests and Code Reviews
Pair Programming

Quality is shared amongst all developers
Quality is shared amongst all developers

Test Driven Development

- Add Test
- Watch Test Fail
- Write Code
- Run Tests
- Refactor
Quality is built-in everywhere throughout the development process.

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Servers
- Usage
- Analytic and Front-end Testing
- User Feedback & Reporting
Continuous integration tools ensure that all changes are merged and tested together frequently.
Quality is built-in everywhere throughout the development process

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Servers
- Usage
- Analytic and Front-end Testing
- User Feedback & Reporting
Automated regression testing

When you fix one bug, you introduce several newer bugs
Automated regression testing

Manually test only once, test automatically forever

<table>
<thead>
<tr>
<th>Location</th>
<th>Count</th>
<th>2017</th>
<th>2016</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jenkins</td>
<td></td>
<td>5674</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sonar</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Files</td>
<td>199</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Functions</td>
<td>599</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
</tr>
<tr>
<td>Duplications</td>
<td>0.4%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Coverage</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
</tr>
<tr>
<td>Line Coverage</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
</tr>
<tr>
<td>Unit Tests Coverage</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
<td>82.8%</td>
</tr>
<tr>
<td>Lines</td>
<td>65</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Blocks</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Files</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Directed Lines</td>
<td>15,795</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Directories</td>
<td>87</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Quality is built-in everywhere throughout the development process

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Security Testing
- Servers
- Analytic and Front-end Testing
- Usage
- User Feedback & Reporting
Vulnerability scanning

Static and dynamic code analysis
Quality is built-in everywhere throughout the development process

- User Feedback & Reporting
- Development
- Continuous Integration
- Regression Testing
- Analytic and Front-end Testing
- Servers
- Performance Testing
- Requirements Gathering
Front-end page load tests

Performance testing

<table>
<thead>
<tr>
<th>Test run: WEBPAGETEST-127.0.0.1</th>
<th>Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>Requests</td>
</tr>
<tr>
<td>Load Time</td>
<td>First Byte</td>
</tr>
<tr>
<td>1.572s</td>
<td>0.263s</td>
</tr>
</tbody>
</table>

RUM First Paint: 0.440s
DOMContentLoaded: 0.509s - 0.509s (0.000s)
loadEvent: 1.574s - 1.692s (0.028s)

Waterfall View

- DNS Lookup
- Initial Connection
- SSL Negotiation
- Time to First Byte
- Content Download
- 3xx response
- 4xx response

- Start Render
- 1stPaint
- DOMContentLoaded
- OnLoad
- Document Complete

- CPU Utilization
- Bandwidth (0 - 20.000 Mbps)
Performance testing

Analytic code performance monitoring
Quality is built-in everywhere throughout the development process.

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Analytic and Front-end Testing
- Usage
- User Feedback & Reporting
- Servers
Server consistency is a key driver of quality
Quality is built-in everywhere throughout the development process

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Integration Testing
- Usage
- Analytic and Front-end Testing
- User Feedback & Reporting
- Servers
Front-end integration tests

Test against live versions of the application
Can the application perform under significant load
Quality is built-in everywhere throughout the development process

- **Requirements Gathering**
- **Development**
- **Continuous Integration**
- **Regression Testing**
- **Servers**
- **Usage**
- **User Feedback & Reporting**
- **Analytic and Front-end Testing**
- **Monitoring and Alerting**
Monitoring and alerting

Extensive monitoring of production systems minimizes downtime
Quality is built-in everywhere throughout the development process.
Extensive test planning and review leads to more effective testing
Testing blitzes allow for testing from multiple perspectives.
Analytic validation is essential to providing a trustworthy tool.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Tester</th>
<th>DB</th>
<th>Description</th>
<th>Analytic</th>
<th>Actual</th>
<th>Status</th>
<th>Regression test</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>@Alex Berlow</td>
<td>ChucK_Regression</td>
<td>Normal multcat</td>
<td>New analysis numbers tie out</td>
<td>No aggregate: <a href="https://testing-apiplatform.com/hdweb/ActivityOutputs/72277?OutputDashboardNewModel/OpenOutputId=10999">https://testing-apiplatform.com/hdweb/ActivityOutputs/72277?OutputDashboardNewModel/OpenOutputId=10999</a></td>
<td>2795109</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BSLY analytic cases.xlsx</td>
</tr>
<tr>
<td>10</td>
<td>@Alex Berlow</td>
<td>UDM MIVNet</td>
<td>New analysis numbers tie out</td>
<td>No aggregate: <a href="https://testing-apiplatform.com/hdweb/ActivityOutputs/75007?OutputDashboardNewModel/OpenOutputId=45941">https://testing-apiplatform.com/hdweb/ActivityOutputs/75007?OutputDashboardNewModel/OpenOutputId=45941</a></td>
<td>2811254</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BSLY analytic cases.xlsx</td>
</tr>
<tr>
<td>11</td>
<td>@Alex Berlow</td>
<td>UDM MIVNet</td>
<td>New analysis numbers tie out</td>
<td>No aggregate: <a href="https://testing-apiplatform.com/hdweb/ActivityOutputs/74079?OutputDashboardNewModel/OpenOutputId=17911">https://testing-apiplatform.com/hdweb/ActivityOutputs/74079?OutputDashboardNewModel/OpenOutputId=17911</a></td>
<td>2817476</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BSLY analytic cases.xlsx</td>
</tr>
</tbody>
</table>
Quality is built-in everywhere throughout the development process

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Servers
- Analytic and Front-end Testing
- Usage
- User Feedback & Reporting
Multiple deployment environments allow for internal user feedback.
Quality is built-in everywhere throughout the development process

- Requirements Gathering
- Development
- Continuous Integration
- Regression Testing
- Servers
- Analytic and Front-end Testing
- Usage
- User Feedback & Reporting
User feedback and reporting

Monitoring usage and collecting feedback helps close the loop between development teams and their clients.
FAQ

Q & A