Testing Challenges in an Agile Environment

Roland Baecker, Sr. Director QA
Roland brings 25 years experience of software development and quality assurance for large scale systems to the QA team at SpotX.

As Director of QA, Roland is focused on increasing the quality and testing efforts in an agile development approach, building out test automation, in house developed testing frameworks as well as developing a release management approach which allows to deploy daily multiple builds to production.

Prior to SpotX, Roland has had numerous positions as a software developer, QA engineer and Test Manager at Alcatel Lucent, T-Mobile, LocalMatters, Intelepeer and IPCommerce/Vantiv. He received a diploma in Electrical Engineering from University of Applied Science Frankfurt/Main (Germany). He lives in Denver, Colorado.
Who We Are

- Holistic video monetization platform for publishers
- Founded in 2007, 275+ employees globally
- First to market with video RTB in 2010
- Connected to 50+ DSPs
- Integrated with over half of comScore top 100 pubs

5+ Billion
Ad decisions per day

Reaching
600+ Million
Uniques every month

Serving impressions in
190+ Countries

Integrated with 450+ Publishers
The SpotX Platform
Status April 2014

- SpotX does not have any QA Organization
- QA is done by Development
- Establish a QA team at SpotX with the following requirements
  - Suitable for Agile / fast paced Environment
  - QA coverage for up to 10 Scrum Teams. Each team typically 4-5 developers plus 1 Dev Lead
Typical Challenges for QA in an Agile environment

- Mini Waterfall
- Continuous Integration – Constantly changing software
- Requirements are smaller and can be changed as development progresses
- Inadequate Testing coverage
- Test Automation
- .....
What needs to be considered for a functional QA Organization?

- Process
- Environments
- Resources
- Which Teams are required
Process

• Jira as
  – Requirement Management
  – Defect Tracking and
  – Test Case Management System

• Confluence as Document Management System
Process - Continued

• Each releasable item (User Story/Bug) has the following documents associated with it
  – Subtask for Handoff to QA (optional)
  – Product Acceptance
  – Test Plan
  – Test Cases
  – Ready for Release Checklist
  – Gerrits
    • +2 for Code Review
    • +1 for QA Verified
User Story

[Dev] Load MOAT only when we have an impression event

Acceptance Criteria:
- Call trackAd function right before our mock VPND object dispatches adImpression

Issue Links
depends on: TCDT-4895 Test Plan for Most Track Ad fix
links to: PFR Checklist

Sub-Tasks
1. Make dev changes
2. Have MOAT review changes
3. Product Acceptance
4. Handoff to QA
5. Create Documentation

Gerrit Reviews
There are no open Gerrit changes.
There are 4 closed Gerrit changes:
Most: Mock VPND object and wait for AdImpression
MOAT: Call mock On AdImpression
MOAT: Mock VPND object and wait for AdImpression
MOAT: Mock VPND object and wait for AdImpression
Change 10293 - Needs Verified

Description:
Mock VPAID object and wait for AdImpression

Using a mock vpaad object to pass to mock, so that event listeners actually work. Wait to call MOATAnalytics.trackAd until an ad impression event has been fired.

Change-Id: Id dac198a5f469210c199523193936dc83f3f5ba5cb
JIRA: YKS-2774, YKS-3265

Author: Eric Kenney <kenney@spotxchange.com>
Commmit: 3a9bca36bc3d513b00c1343f2ba20ed9e08840f
Parent(s): 133fac6af471f68514039885937d8a8e265ac53
Change-Id: Id dac198a5f469210c199523193936dc83f3f5ba5cb

Files:
- Commit Message
- AS3_AdPlayer/spotxchange/ad/core/tracking/TrackingManager.as
- AS3_AdPlayer/src/com/spotxchange/ad/AdManager.as
- AS3_AdPlayer/src/com/spotxchange/ads/AdManager.as
- AS3_AdPlayer/src/com/spotxchange/ads/AdManager.as
- AS3_AdPlayer/src/com/spotxchange/ads/MoatVPAID.as
- AS3_AdPlayer/src/com/spotxchange/ads/MoatVPAID.as
- AS3_AdPlayer/src/com/spotxchange/ads/MoatVPAID.as
- AS3_AdPlayer/src/com/spotxchange/ads/MoatVPAID.as
- AS3_AdPlayer/src/com/spotxchange/ads/MoatVPAID.as
- AS3_AdPlayer/src/com/spotxchange/ads/MoatVPAID.as

Conflicts With (3):
- IDOD: Add additional logging
- Make the SWF Smaller
- IDOD: Start Beacon not firing and DODE: Smaller

Code-Review: +2
Has-Changelog: +1
QA- Verified: +1
Verified: +1

Date: Jun 17, 2015 2:01 PM
Updated: Jul 15, 2015 4:45 PM
# Ready For Release Checklist

2015/07/28 - Admin Tool Fix - Playtime preserved after switching ad source

Created by Peter Go, last modified by Joe Litwack on Jul 22, 2015

<table>
<thead>
<tr>
<th>Team</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2015/07/28 - Admin Tool Fix - Playtime preserved after switching ad source</td>
</tr>
<tr>
<td>JIRAs</td>
<td><a href="https://jira/browse/YWS-1330">https://jira/browse/YWS-1330</a></td>
</tr>
<tr>
<td>Application Branches</td>
<td>master; spotx-web-admin</td>
</tr>
</tbody>
</table>

**Before Going to RE**

<table>
<thead>
<tr>
<th>Initial</th>
<th>Description</th>
<th>Role</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Unit tests are written and passing</td>
<td>Dev</td>
<td>Change #10470 has some failing UTs but these failures pre-date this change - Steve from Film has UT fixes to be checked in. Change #10597 is not yet set up with unit tests (per Joel).</td>
</tr>
<tr>
<td>✔️</td>
<td>Test plan written and executed in LOD</td>
<td>QA</td>
<td>Way to go Dom!</td>
</tr>
<tr>
<td>✔️</td>
<td>Test plan reviewed with team</td>
<td>QA</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>All issues are accepted by PO team</td>
<td>Dev</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Automated test suite executed in LOD</td>
<td>QA/Dev</td>
<td>N/A - not a candidate for automated tests</td>
</tr>
<tr>
<td>✔️</td>
<td>All code review items are completed</td>
<td>Lead</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>All issues have +2</td>
<td>Lead</td>
<td>Gerrits: <a href="https://gerrit.hq.booyahnetworks.com:8080/#/q/topic:-playtime-admin">https://gerrit.hq.booyahnetworks.com:8080/#/q/topic:-playtime-admin</a></td>
</tr>
<tr>
<td>✔️</td>
<td>All issues have changelog</td>
<td>Any</td>
<td><a href="https://gerrit.hq.booyahnetworks.com:8080/#/c/10597/">https://gerrit.hq.booyahnetworks.com:8080/#/c/10597/</a></td>
</tr>
</tbody>
</table>
Environments

• At least 1 Test Environment per scrum team

• Multiple Staging environments for release management

• Multiple Staging environment for automation

• Sandbox environments for End-to-End testing for business

• Demo environment for sales
Resources

• 1 QA Engineer per Scrum Team

• Release Engineers for Production Deployments

• Automation Engineers to increase Test Coverage

• Dev/Ops Engineers to support Production Support

• PM/Scrum Master to coordinate between Teams
Teams

9 Scrum Teams in parallel

=>

9 QA Engineers

7 Automation Engineers

3 Release Engineers

3 Dev/Ops Engineers

+ new development hires

2 PM
Testing Workflow

- Test Env
- Prod Env
- Sandbox
- Staging Env
- Release Mgr

Business ETE Tests

Automated Testing
Scrum Teams

• 9 Scrum Teams in parallel

• Classic Agile Software Lifecycle
  – Weekly Product Design / Grooming
  – Daily Standups
  – Estimation Meetings
  – Plan to Plan
  – Sprint Demo’s
  – Sprint Planning
  – Retroperspective
  – Project Status Meetings for Epics across different teams
Scrum Teams - Continued

- 3 week Sprints

- Each Team: 4-5 Developers + 1 QA Engineer + 1 Dev Lead
  - Mark items as RFR as they are ready and hand them over to the PM
  - Don’t wait to the end of the Sprint
  - In case User Stories have not been completed in current Sprint, Sub Tasks move over to new Sprint
Release Management Team

• Pick up RFR items as they come in

• Bundle RFR items together based on features or same packages

• Deploy new release on a staging environment

• Execute Functional, Integration and Automation Tests in staging environment

• Deploy and test in sandbox environment
  – Have Business signing off on new releases for Web Applications and API’s
  – In Sanbox, Web Applications and API’s are pointing to Production DB’s
Release Management Team - Continued

• Sign off on release

• Prepare Release Procedure

• Support Systems with Production Deployment

➤ Daily (except Friday’s) between 1 – 3 releases to production
What is the definition of Ready For Release (RFR)

- Code complete
- Unit Test pass
- Code Review complete
- Test Plan and Test Cases defined
- Test Cases executed and pass
- RFR Checklist complete
Version Control – Git-Flow Workflow

• We follow the "git flow" workflow
  https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

• On a high level,
  
  – RFR items and scheduled for release items in a stable development branch “master”
  
  – Each scrum team is developing new projects in an “epic” branch (branch off master)
  
  – still have the flexibility to merge single gerrits into master when RFR (smaller changes when it does not make sense to create a separate epic branch)
  
  – Release branch reflecting was is getting released and mirroring production
Version Control – Git-Flow Workflow - Continued

- Master branch has to be stable at all time
- Epic branches are getting rebased with Master branch daily
  - RFR items which are scheduled for a production release are getting propagated to all scrum test environments daily
  - Immediately knowledge when master is broken
- New, RFR release items are getting tested automatically by all scrum teams
Required Tests before Releasing to Production

Summary

• RFR => sign off by Scrum Team

• Merged in Master and rebased to all scrum team environments => automatically tested by all scrum teams again

• Deployed on a staging environment and sign off by the Release Team

• Successful automation run => Daily
Release Calendar
How to deploy in production

• Release Procedures follow general steps:

  – Remove server out of LB
  – Deploy new build on server
  – Smoke test new build
  – Put server back in LB and bake
  – Roll new build to multiple server / one datacenter / all datacenter
Planned improvements for production rollout

• Currently new features are immediately live after rollout. Based on the data volume (5 billion requests per day), we see issues within minutes/hours. However:

• In the future, releases should be rolled out with new features disabled. Once a new release has been in production for “long” enough, a new feature can be turned on slowly

• In the future, a new feature should be turned on for “Beta” customers first
Challenges

• Right frequency and size of builds

• Too many small releases
  – High cost on release management and systems
  – Production instability

• Less, but bigger releases
  – Production issues are hard to localize
  – Hard to get a release completed
  – Reverts of single features too complicated
  – Rollback reverts to many new features.
Challenges - Continued

- Constant, small releases with fixed schedule plus ability for additional releases based on business needs. For example
  - Monday: Web Application Server
  - Tuesday: Publisher/Advertiser Interfaces
  - Wednesday: Backend Server
  - Thursday: SpotX Platform
Automation Team

• Selenium Webdriver / Java
• Maven (project build and management)
• TestNG
• Various open Source Java libraries
• RestAssured (API)
• Internal developed frameworks
• Jenkins
Nightly Runs out of Jenkins
Automation Environments

• Build and deploy “Production” + any planned release to a staging environment daily. Test Automation will show us if planned release can be pushed to production

• Build and deploy any future release to a staging environment for Test Automation Sprint work.
Challenges

• Well defined, maintainable and reliable set of automated test cases

• Well defined baseline for test results

• Ability to have exact test results early, every morning

• Communication between Scrum, Automation and Release Team

• Test results of nightly runs determine production deployments
Production Support

- Previously Production issues have been routed to Scrum teams
- Production Issues have interrupted Sprint work. Scrum teams couldn’t complete sprint as committed
- Team of Dev/Ops/QA takes care of incoming production issues. 90% of all incoming production issues are now handled by Production Support team.
- Only critical or new feature requests are routed in scrum teams
- Production Support is entry for new developers to get knowledge to our system and the code base
Project Manager

• What is the role of the PM in the workflow
  – Ensure that only user stories are marked as RFR which meet criteria
  – Work with scrum teams to identify dependencies between user stories
  – Work with scrum teams to implement, forecast and manage Epics and multi-scrum team projects
  – Forecast with Release Manager possible release dates and communicate to Product Management
  – Work with Business to plan and execute End-to-End tests on Sandbox environment
How does SpotX QA manage the challenges of Agile

• We avoid “Mini Waterfall” by releasing daily what is RFR

• Releases are independent from Sprints and the typical end of Sprint crush

• Scrum teams feel more comfortable to move tasks to next sprint in case its not complete

• QA does not have to test entire Sprint work on last two days of Sprint
How does SpotX QA manage the challenges of Agile

- We have changing or not well defined requirements, but
  - QA is involved in all decisions at any time
  - Each User Story needs PO acceptance to make sure that we develop and test what Product wants

- We have constantly changing software, but with
  - testing new features as part of the scrum team
  - merging master in epic branches and propagating to LOD environments
  - Integration and functional testing in stage
  - End-to-End testing in sandbox
  - daily automation runs

➢ we have the required testing coverage for daily releases in production
How does SpotX QA manage the challenges of Agile

- With multiple, small releases daily we
  - immediately can identify production issues and have the ability to take required actions
  - reduce dependencies and have a better way to get new functionality in production than deploying “big” releases once every 2 weeks, monthly, quarterly, ....
  - separate sprint schedules from release management
How does SpotX QA manage the challenges of Agile

• With multiple staging environments for automation, we are able to
  – run daily automated tests for production releases
  – work in parallel on future releases

➢ Consistently having a functional Automation Suite in place

➢ Having the test coverage in place we need to push daily to production
THANK YOU