CI/CD – An Essential Ingredient for Mobile First Initiatives
Agenda

• AGILE Adoption, Continuous Integration (CI) & Continuous Delivery (CD) – InfoStretch Findings
• Mobile design challenges
• Continuous Integration and Delivery
  • What? Why? How?
• A real-life case study
• Q&A
InfoStretch Findings – Agile & CI/CD Adoption

• Transformation at varying maturity levels
• Much higher adoption in web applications
• Mobile is still a challenge
Mobile Design Challenges

• Mobile Apps live or die by App store ratings

• Margin for error is too low
  • Could talk over a week duration to push to production app store
  • Cannot afford human errors with deployment, artifacts/config management etc.
    • Example: PROD app signed with QA Provisioning Profile

• Device & OS Fragmentation
  • Test coverage to ensure quality
  • Longer testing cycles due to lack of automation
Agile Delivery Pipeline Without Automation

Key Challenges:

- No early feedback
- Inefficient use of engineering time
- Release planning & scheduling risks
- Delivery pipeline not optimized
What is CI/CD?

“Fully automated delivery pipeline from code commit to production release”
CD landscape – Mobile Apps Example
Key Benefits

• Build the right product
• Go-to-market faster
• Get early market feedback
• React and respond to change quickly
• Facilitate Innovation
• Improve reliability & consistency
• Reduce cost and improve ROI
• Competitive Advantage
# Case Study – Automobile Data Broker

## Before

**CI IMPLEMENTATION**

- SCM was not streamlined
- Manual builds – Need to be scheduled
  - Manual deployment of Apps to devices
- Longer QA cycles – No Test Automation

## After

**CI IMPLEMENTATION**

- SCM workflows – Feature, Dev, Master branches
- Automated the build process ensuring continuous build availability
- Automated deployment of builds to cloud based Mobile Devices
- Automated functional tests
- Reporting engine to deliver continuous results to key stakeholders

<table>
<thead>
<tr>
<th>654</th>
<th>114</th>
<th>300+</th>
</tr>
</thead>
<tbody>
<tr>
<td>654 development builds</td>
<td>114 parameterized QA builds</td>
<td>Automated 300+ test scenarios</td>
</tr>
</tbody>
</table>
CI/CD Tools

InfoStretch CI/CD Offerings

User Roles
- Developer
- Product Manager/Developer
- QA Engineer
- Release Engineer
- Operations Engineer

Build Automation
Activities
- Requirements
- Scripted Builds
- Build Once - Deploy Anywhere
Tools
- JIRA / Rally
- Jenkins / CloudBees
- Maven / Gradle
- Artifactory / JFrog

Test Automation
Activities
- Unit Tests / API Tests / Functional Tests / System Tests
- TDD / BDD
- Analytics / Security
Tools
- JACoCo, Clover
- Crucible, SonarQube
- Cucumber, JBehave
- AppDynamics, Fortify

Deployment Automation
Activities
- Zero touch Continuous Deployment
- Distributed Build Machines
Tools
- Docker, Electric Cloud, OpenStack
- PuppetLabs, Chef

Dashboard & Reporting
- Product > Sprint > User > Story > Build
- Real Time Execution Details
- Drill Down Reports
- Integration with ALM Tools
- Auto Notification of Pipeline Status

Web Application, Mobile Application (Native App, Responsive Web App)
- Platform-as-a-Service (PaaS)
- IBM Bluemix, RedHat OpenShift, Microsoft Azure, Amazon AWS

©2016 InfoStretch Corporation. All rights reserved.
Thank You

Info@infostretch.com
408.727.1100
Appendix
# Different tools set to build CI/CD

<table>
<thead>
<tr>
<th>Components</th>
<th>Tools</th>
<th>Cost</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application analytics</td>
<td>Nagios, New Relic, Splunk</td>
<td>Medium to high</td>
<td>Getting information about customer behavior requires writing code to collect that data</td>
</tr>
<tr>
<td>Artifact repository (AKA component/package manager or asset manager)</td>
<td>Sonatype Nexus, Apt and Yum (open source)</td>
<td>low</td>
<td>Low. The technology is stable, well proven, and requires minimal changes in processes and developer behavior when integrated into automated build process.</td>
</tr>
<tr>
<td>Automated functional testing</td>
<td>CA Application Test, Cucumber, HP UFT, Microsoft, IBM RTW, Borland SilkTest, Parasoft, SmartBear, Selenium,</td>
<td>medium</td>
<td>Agile adoption grows is newer AFT tools that enable test-driven development and behavior-driven development (Cucumber, Fit/Fitnesse). A primary differentiator of the modern tools is their suitability for frequent use across multiple small change sets</td>
</tr>
<tr>
<td>Build automation</td>
<td>Gradle, Maven, Microsoft Build, Rake</td>
<td>low</td>
<td>Build automation tools are software applications that automate the compilation and integration of executable software components from source code</td>
</tr>
</tbody>
</table>

Reference: TechRadarTM: Continuous Software Delivery, Q2 2015 by Kurt Bittner and Diego Lo Giudice, May 1, 2015 | Updated: May 21, 2015
## Different tools set to build CI/CD

<table>
<thead>
<tr>
<th>Components</th>
<th>Tools</th>
<th>Cost</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content management</td>
<td>Acquia, Adobe, Wordpress</td>
<td>low</td>
<td>The value of CMS is to disconnect content from its delivery, thus allowing non-developers to edit content without having to change.</td>
</tr>
<tr>
<td>Continuous delivery management</td>
<td>Go.CD (open source), Electric Cloud ElectricFlow, Electric Cloud ship.io, IBM UrbanCode Release, Serena Release Manager</td>
<td>Medium</td>
<td>They orchestrate the application delivery workflow from code check-in to deployment of the application into one or more production environments.</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Atlassian Bamboo, CircleCI, Electric Cloud, Cloudbees (Jenkins), JetBrains TeamCity, Travis CI</td>
<td>low</td>
<td>Continuous integration tools orchestrate the CI process starting from the point code is checked into a version management system, kicking off a build and integration process, and initiating test environment provisioning, deploying the build into the testing environment and initiating automated testing</td>
</tr>
<tr>
<td>Environment management</td>
<td>Amazon, Ansible, CA Technologies, Chef, Docker, HP, IBM, Plutora, XebiaLabs</td>
<td>High</td>
<td>Environment management tools automate the provisioning and management of environments based on versioned configuration information.</td>
</tr>
</tbody>
</table>

Reference: TechRadarTM: Continuous Software Delivery, Q2 2015 by Kurt Bittner and Diego Lo Giudice, May 1, 2015 | Updated: May 21, 2015
## Different tools set to build CI/CD

<table>
<thead>
<tr>
<th>Components</th>
<th>Tools</th>
<th>Cost</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life-cycle integration</td>
<td>Borland Connect, CollabNet, TaskTop Sync</td>
<td>Low</td>
<td>Life-cycle integration tools tie together tools from multiple vendors into a single integrated tool chain, ensuring data consistency and integrity. They also provide reporting and visibility across the integrated tool chain.</td>
</tr>
<tr>
<td>Performance/load testing</td>
<td>Borland, CA Technologies, Dynatrace, Gatling (open source), HP, IBM, Neotys, Parasoft, SmartBear, Soasta, OpenSTA</td>
<td>Medium to high.</td>
<td>Tools are relatively inexpensive, but modeling workloads requires expertise and understanding real-world usage scenarios. As performance testing shifts more to the left, tools are becoming more palatable to developers.</td>
</tr>
<tr>
<td>Service virtualization</td>
<td>CA LISA, HP Service Virtualization, IBM Rational Test Workbench, Parasoft, Tricentis Tosca Virtualize, Tosca Orchestrator</td>
<td>Medium</td>
<td>Service virtualization tools simulate the behavior of a system or service by providing an alternate simulated implementation of that system or service’s APIs.</td>
</tr>
<tr>
<td>Static analysis</td>
<td>CAST, HP Fortify, Rogue Wave Klocwork, SonarQube</td>
<td>Low.</td>
<td>Static analysis tools analyze source code for variation from a set of defined coding standards. They are used to find errors in the code, including architectural anti-patterns and security vulnerabilities.</td>
</tr>
</tbody>
</table>

Reference: TechRadarTM: Continuous Software Delivery, Q2 2015 by Kurt Bittner and Diego Lo Giudice, May 1, 2015 | Updated: May 21, 2015
# Different tools set to build CI/CD

<table>
<thead>
<tr>
<th>Components</th>
<th>Tools</th>
<th>Cost</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test data management</td>
<td>Grid Tools, Informatica, IBM Optim, Tricentis Tosca Test Suite</td>
<td>Medium</td>
<td>Test data management tools create and manage test data.</td>
</tr>
<tr>
<td>Test management</td>
<td>Borland Silk, HP Quality Center, IBM Rational Quality Manager, MS Test Professional, Parasoft, SmartBear, Tricentis Tosca Test Suite</td>
<td>medium</td>
<td>Test management tools manage the software testing process, typically including management and creation of test cases, test automation artifacts including testing environment configurations and test data, and test results.</td>
</tr>
<tr>
<td>UI-based test automation</td>
<td>Accenture, Eclipse</td>
<td>medium</td>
<td>UI-based test automation generates events such as clicks or keystrokes and observes the changes in the GUI.</td>
</tr>
<tr>
<td>Unit testing</td>
<td>JUnit, NUnit, Parasoft, Microsoft, TypeMock</td>
<td>Medium</td>
<td>verify that the code of a class, method, function, statements, conditions, etc. work in conformance to the developer’s design.</td>
</tr>
<tr>
<td>Version management</td>
<td>Git, IBM, Microsoft, Serena, Subversion, Perforce</td>
<td>low</td>
<td>Version management tools provide the ability to store separate versions of files, controlling visibility and access, as well as managing variation through branching and merging.</td>
</tr>
</tbody>
</table>

Reference: TechRadar™: Continuous Software Delivery, Q2 2015 by Kurt Bittner and Diego Lo Giudice, May 1, 2015 | Updated: May 21, 2015