The Last Mile: Continuous Delivery

Darryl Bowler, CollabNet
Brian Dawson, CollabNet
Presenters - The Last Mile: Continuous Delivery

Darryl Bowler, Senior Systems Architect, Services, CollabNet

With more than fifteen years of IT experience, Darryl Bowler leads the consulting services business for CollabNet Lab Management, delivering high-value virtualization and test lab automation consulting services to major enterprises. Darryl is an accomplished senior systems architect, specializing in software development methodologies, virtualization, and cloud computing, combined with application lifecycle management and agile best practices.

Brian Dawson, Senior Consultant, CollabNet

As a Senior Consultant at CollabNet, for 5 years Brian Dawson has helped software development organizations large and small improve the Application Lifecycle Management process through the use of TeamForge. Prior to CollabNet, Brian spent 12 years in software development. Ten of those years were spent at Sony Computer Entertainment where he worked as the Director of Tools and Technology and oversaw the development and deployment of TeamForge as a worldwide collaboration portal.
Agenda

- Continuous Delivery Defined
- Implementation of Continuous Delivery
  - Participants
  - Tools
- The Early Stages
  - Plan-Develop-Commit
  - Task-Based CD for Governance
- Functional/Acceptance Testing
  - Behavior Driven Development
- Release / Deployment stage
  - The New User Story
  - Infrastructure as code
  - Making DevOps Real
“CD is about moving away from making the software ready as a separate activity, and instead developing in a way that means the software is always ready for release.”

- Kief Morris
Continuous Delivery - Defined

- Continuous Delivery extends the CI process from build to include test and deploy phases
  - Ensure fluid coordination and communication between
  - Customers, Managers, Developers, Testers, Operations, etc

“Continuous delivery of software needs processes that encompass the entire delivery process, from concept to use.”
— Unknown
Implementation of Continuous Delivery - Participants

Customer
- Needs
- Priority

Product Manager
- Features
- Design
- Plan

Developer
- Estimate
- Implement
- Fix

Quality Assurance
- Script
- Test
- Report

Operations
- Install
- Config
- Maintain

Customer
- Use
- Respond

Communication/Feedback

Automation

Version Control

Environment Application Configurations

Environment Application Configurations

Testing
- Configure environment
- Deploy binaries
- Smoke tests
- Regression

Implementation
- Configure environment
- Deploy binaries
- Smoke tests
- Regression

Production
- Configure environment
- Deploy binaries
- Smoke tests

Source Code

Changsets Tests

Changesets

Changesets Tests

Developers

Code Metrics Build Failures Test Failures

Quality Assurance

Machine Provisioning
Stack Provisioning
Application Deployment

Operations

Machine Provisioning
Stack Provisioning
Application Deployment

Artifact Repository

Binaries Metadata Reports

Binaries

Metadata Reports
Implementation of Continuous Delivery - Interactions

Customer
- Needs
- Priority

Product Manager
- Features
- Design
- Plan

Developer
- Estimate
- Implement
- Fix

Quality Assurance
- Script
- Test
- Report

Operations
- Install
- Config
- Maintain

Customer
- Use
- Respond

Communication/Feedback
Automation

Version Control
Environment
Source Control
Changesets
Tests

Artifacts Repository
Binaries
Metadata Reports

Environment Configuration
Testing
Package environment
Binaries

Copyright © 2012 CollabNet, Inc. All Rights Reserved
Implementation of Continuous Delivery - Tooling

- Planning Tools
  - Task Management
- Build Management
  - CI Server
- Testing Tools
- Configuration Management
- Virtualization
  - Data Center
- Tracking/Defect Management
- Version Control
- Code Quality
- Artifact Repository
  - Dependency Management
- Reporting

Integration/Traceability/Visibility
Continuous Delivery Needs Functioning Delivery Pipeline

All Stages are important and interdependent..

Thus.....
Continuous Delivery Needs Functioning Delivery Pipeline

GIGO
GARBAGE IN. GARBAGE OUT.

“Quality Control” from planning->develop->commit…
The Steps of the Delivery Pipeline

- Capture customer needs
- Right-Size
- “Appropriate” details
  - Not too much
  - Not too little
- Decompose into “trackable” units
  - App Dev
  - Database
  - Config
- TDD/BDD, create acceptance test upfront
- Develop code and test automation to the acceptance test
The Steps of the Delivery Pipeline

- Visibility
- Integration
- Communication
- Documentation
The Steps of the Delivery Pipeline

- Create links to work units
  - Changes are business driven
- Proper comments
  - Documented, visible
The Steps of the Delivery Pipeline

- Gate the commits
  - Changes have to be controlled, and deliberate
  - Know what is in the release

```
commit -m "Adding activation/deactivation method stubs." /Users/Brian
Sending /Users/Brian/Documents/workspace/Registration System
Transmitting file data ...
Committed revision 13.
commit -m "Updating activation/deactivation method stubs." /Users/Brian
Sending /Users/Brian/Documents/workspace/Registration System
Transmitting file data ...
svn: Commit failed (details follow):
svn: Commit blocked by pre-commit hook (exit code 1) with output:
Valid commit.
artf1355 is a valid object ID.
<table>
<thead>
<tr>
<th>Pre-Commit Hook</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>artf1355</td>
</tr>
</tbody>
</table>
```
• SVN feature or dev branches…maybe
• DVCS forks…maybe
• Help “package” changes but increases complexity
  • Multiple build jobs
The Steps of the Delivery Pipeline

• Run functional test on Story units, instead of individual changes
  • Gate promotion
  • Don’t send incomplete part down the line
• Report output
  • Inform team

Build Reports

Stories By Status

Stories in Pre-Promotion

<table>
<thead>
<tr>
<th>Priority</th>
<th>Artifact ID : Title</th>
<th>Assigned To</th>
<th>Submitted By</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>No results found</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Showing 1-0 of 0

Stories on Stage

<table>
<thead>
<tr>
<th>Priority</th>
<th>Artifact ID : Title</th>
<th>Assigned To</th>
<th>Submitted By</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Highest</td>
<td>artf6548 : [Sample] Story One</td>
<td>Jenkins Service Account</td>
<td>TeamForge Administrator</td>
</tr>
</tbody>
</table>

Showing 1-1 of 1

All Builds

<table>
<thead>
<tr>
<th>Priority</th>
<th>Artifact ID : Title</th>
<th>Assigned To</th>
<th>Submitted By</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>artf1073 : dl_scrips-1D</td>
<td>None</td>
<td>Jenkins Service Account</td>
</tr>
<tr>
<td>None</td>
<td>artf1079 : JEE-Webapp-33</td>
<td>None</td>
<td>Jenkins Service Account</td>
</tr>
</tbody>
</table>

20. remainingEffort => 0
21. reportedReleaseId =>
22. resolvedReleaseId =>
22. status => In Development
24. statusClass => Open
25. title => Update MemberList Data
26. version => 1.0
comment =>
Testing artf6548
"This build operation is blocked. artf6548 is expected to have a status of Ready for UN"](http://wiki/collabnet)

Copyright © 2012 CollabNet, Inc. All Rights Reserved
Example Task-Based CI/CD With Governance

Development (Commit Stage) -> UAT Promotion (Functional Stage)

Story entered and assigned → Developer "picks" story, sets status, makes changes and commits with "association" → Validate story status, Fail if not "In Dev" → Developer updates story status to "Ready for UAT" → Poll change set & story status

Build Begins → CI Server (Jenkins) parses change-set and related stories

CI Server (Jenkins) parses change-set and related stories → Validate story status, block if not "Ready for UAT" → CI Server (Jenkins) promotion to UAT completes → Auto Deploy to UAT

Auto Update story status, Environment, etc "In UAT Environment" → Dev->UAT Promotion Complete → "Promotion Report" available, QA Notified → Functional Test Executed
Poll Question #2

Which tools are integrated in your delivery pipeline?
Acceptance Testing + Behavior Driven Development (BDD) Stage 2

A user story is not complete unless it passes the acceptance tests - XP

- Is an evolution of TDD
- Strong collaboration with stakeholders regarding behavior of software
  - Devs, QA and Business

**DISCUSS** with Business Owner what is the intended behavior of the user story? Define the behavior as a story

**DISTILL** tests into a common understood language.
  - Technical implementation abstraction
  - Given, When and Then

**DEVELOP** the code and map

**DEMO** the implementation.
1: Story resulting from discussion and distilled into a common language

2 and 3: Develop

4 and 5: Demo
A New View of a Sprint?

A sprint is the basic unit of development in Scrum

But... Development of a production product requires more than just software development

- Visibility of true level of effort
- Path way to production readiness
- Ensure all dependencies are completed
As a user, I want to search for my customers by their Last name and SSN.
Infrastructure As Code

Is the concept that infrastructure should be treated as source code

- Aligns software developments practices with infrastructure
  - Tags, branches and releases
  - Manage life-cycle
    - Test, Dev, UAT, Production
  - Continuous Testing
# site.pp

class named {
    package { caching-nameserver: ensure => installed }
    package { bind-chroot: ensure => installed }

    file { "/var/named/chroot/etc/named.conf":
        source => "puppet:///public/e-securenetworks.named/named.conf",
        ensure => present,
    }

    file { "/var/named/chroot/var/named/e-securenetworks.net.db":
        source => "puppet:///public/e-securenetworks.named/e-securenetworks.net.db",
        ensure => present,
    }

    file { "/var/named/chroot/var/named/10.168.1.db":
        source => "puppet:///public/e-securenetworks.named/10.168.1.db",
        ensure => present,
    }

    file { "/var/named/chroot/var/named/10.168.2.db":
        source => "puppet:///public/e-securenetworks.named/10.168.2.db",
        ensure => present,
    }

    file { "/var/named/chroot/var/named/named.root":
        source => "puppet:///public/e-securenetworks.named/named.root",
        ensure => present,
    }

    service { named: ensure => running,
        enable => true,
        subscribe => File["/var/named/chroot/etc/named.conf", "/var/named/chroot/var/named/e-securenetworks.net.db", "/var/named/chroot/var/named/10.168.1.db"],
    }
}
What are your 2012 focus areas?
• Cross functional visibility

• Full tractability to even system changes to support functionality

• Realistic project planning for all production requirements

• Enhanced Burn Down

• Enhanced Roll-Up
Making Automated Deployments Real

Agile Project Development
- Manager
- Developer
- Product/Sprint Planning
- Stories

Source Code Management
- SCM Server
- Change Sets
- Checks in
- Checks out
- Code Review

Automated Build/Test
- Build/Tag
- Unit Tests
- Functional Tests
- Static Code Analysis
- Code Coverage
- License Compliance

Deployment Services
- Runtime Dependencies
- Governance Process
- Change Management

Artifact Publication
- Artifact Repository
- Dependency Management
- Governance Process
- Artifact Reuse
- Reporting

CI Servers
- Binary
- Libraries

www.collab.net
Copyright © 2012 CollabNet, Inc. All Rights Reserved
Summary

- **Continuous Delivery**
  - Software always ready to release
  - Involve collaboration from planning to production
  - Does not prescribe “How”...is different for every team

- **“Governing” the pipeline is important**
  - Prevent downstream issues which affect releasability

- **Traceability is essential**
  - Smooth hand-off from Plan->Dev->Production

- **TDD/BDD**
  - Enables cross functional collaboration...
  - Clearly defines expectations between Dev/Business

- **Infrastructure as code**
  - Clearly defines expectations between Dev/Operations
The Steps of the Delivery Pipeline

Build 57

Plan: JEE-Webapp  Build: 57  Status: Success  Duration: 00:01:22

Completed on: 18.04.2012 01:34:49.396 +0700
Cause: Started by an SCM change.

Test Results

No test results generated.

Artifacts

No artifacts.

Changes

• task [artf1355]: Allow members to be activated and deactivated vs deleted and restored  bdawson
  /src/main/java/collabNet/controller/MemberRegistration.java
Commit to: https://calt.hhs.gov/svn/repos/bnt_example/trunk/src

Enter a comment for the commit operation.

Adding stub for activate/deactivate membership

Configure Comment Templates...

Changes

- bnt-example-src/main/java/collabNet/controller
- MemberRegistration.java

Artifact artf1355

- Allow members to be activated and deactivated vs deleted and restored
- Status: Under Consideration
- Submitted By: TeamForge Administrator
- Submitted: Apr 12, 2012 2:31 AM

Description

When the member Bob falls if fails make payment the Admin can deactivate his account.
When Bob becomes current with payments the Admin can reactivate his account.

Attributes

- Group: None
- Status: Under Consideration
- Priority: 2 - High
- Assigned To: bdawson
- Planning Folder: None
- Calculate Effort: 0
- Remaining Effort: 0

Context: Tracker

Artifactory cost:

Artifact changed: Allow members to be activated and deactivated vs deleted and restored
- Status: Under Consideration
- Submitted By: TeamForge Administrator
- Submitted: Apr 12, 2012 2:31 AM

Description

When the member Bob falls if fails make payment the Admin can deactivate his account.
When Bob becomes current with payments the Admin can reactivate his account.

Attributes

- Group: None
- Status: Under Consideration
- Priority: 2 - High
- Assigned To: bdawson
- Planning Folder: None
- Calculate Effort: 0
- Remaining Effort: 0

Context: Tracker

Artifactory cost: