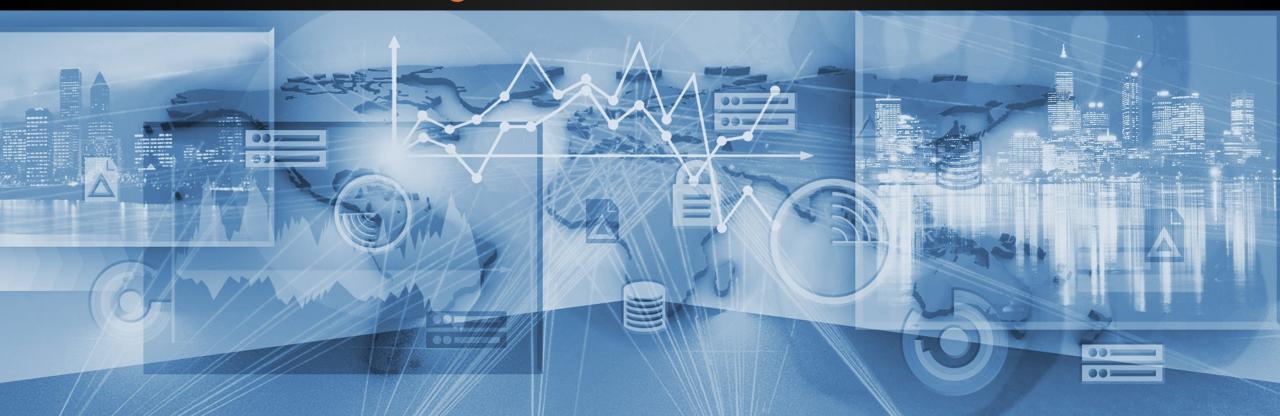


C++Now 2015 – Lightning Talks

Gwendolyn Hunt Senior Software Engineer Tripwire R&D

Surviving and Thriving in a Multiplatform, Multilanguage, Multiproduct, Multisite Continuous Integration Environment





Surviving and Thriving in a Multiplatform, Multilanguage, Multiproduct, Multisite Continuous Integration Environment

In other words:

- Know where your source came from
- Know what you are Building
- Know what you are Testing
- Know what you are Shipping
- Know what you are Supporting



Assumptions

- Either racks full of discrete machines or Private Cloud Infrastructure
- Sane artifact versioning scheme
 - git short hash
 - revision major.minor.patch(.micro)
 - build number



Item 1 - Artifact Repository

Example: Artifactory

- Secure storage of your built components
 - third-party lib source
 - idl/idl bindings
 - libs, components, applications
 - installers, distribution packages
- Tool Chain utilities
 - cmake modules
 - gradle plugins
- Test Support
 - locally built or modified Ruby gems
 - VM Provisioning scripts (Chef recipes, cookbooks, etc.)



Item 2 - Source Repository

Example: local instance of github

- Source for
 - Components
 - Libs
 - Applications
 - Installers
- ◆ C++ Core moving to git flow
 - Development branch nightly continuous integration tests
 - Master branch released code, less frequent integration tests
 - Feature branches
 - short-lived targeted development for new feature
 - Merged into development branch



Item 3 - Automated Build Infrastructure

Example: Jetbrains's TeamCity

- dedicated VMs
 - build agents organized by platform
 - Tool chain
 - Bitness
 - ♦ Windows 18 platforms: WinXP, Vista, Win7, Win 8, Win8.1, Server 2003, Server 2008, Server 2012 plus 4 POS platforms
 - ◆ Linux 8 platforms: RedHat and CentOS 5&6 (pending SuSE, RHEL7 CentOS7, Oracle Linux)
- Builds component, lib or application
 - Triggered by source commit
 - executes acceptance test after build
 - publishes artifact when acceptance test passes



Item 4 - Full Spectrum Testing

- C++ unit tests
 - google test, boost test short running must pass before code review
- C++ acceptance tests Ruby RSpec, Cucumber
 - short acceptance (10 min) executed post build, pre publishing to artifactory
 - long acceptance (60+ minutes) nightly development branch tests on every platform on Jenkins
- Automated and Ad hoc System, Scale and Performance tests
 - pulls artifacts from Artifactory
 - Chef (or Puppet) used for provisioning test infrastructure



Item 5 - Use the Right language for the Task

- Retrieve Artifacts
 - Java, Ruby build target use gradle plugin
 - ◆ C++ build target cmake maven resolver
- Create build environment
 - cmake
- Acceptance tests
 - Happy path: Ruby Cucumber
 - Complex normal and abnormal behavior:
 - Ruby Rspec
 - Ruby Rake Run tests without build environments



Item 6 - Collect Metrics

About everything

- Artifacts
 - Dependencies
 - Licenses and license attributions
- Build history
- Test history
 - acceptance
- Review daily
 - Are required to understand all failures
 - Tests
 - Test Harness failures