

ThoughtWorks®

*Continuous Delivery
Addressing the Last Mile*



Agenda

- * The problem
- * Continuous Delivery and how it can help
- * Making it a reality

The Cycle

- * New idea about how to improve “things”
- * Testable hypothesis about the improvement
- * Quickly and safely make and deploy the change
- * Check whether the hypothesis holds
- * Repeat

Continuous Delivery Objective

- * Release any time at the push of a button
- * Deployments should be BORING!
 - Excitement of releases from features
- * Requires automating pretty much everything
 - Environments
 - Configurations
 - Build
 - Testing
 - Deployment

Caveat

✂ Just because you can, doesn't mean you have to release everything all the time

Environments

✂ Script the provisioning of environments

✂ No “works of art”

Configuration Parameter Management

- ✂ Use wisely and consistently
- ✂ Always check in to version control
- ✂ Identify by version and environment
- ✂ Understand version relationships between applications

Build

- * Can always rebuild a complete system from version control automatically

Testing

- ✂ All levels of testing should be automated to the extent possible
- ✂ Early and often integration and performance testing
- ✂ Automated smoke and regression functional suites that provide production safety net

Deployment

- ✂ Not just automated builds but a build and deployment pipeline
- ✂ Automated promotion across environments (possibly controlled by manual approvals)
- ✂ Movement of binaries after the first stage
- ✂ Deploying to production not a ceremony because same process used all along

Continuous Delivery Benefits

* Deployments are:

- Stress free
- Fast
- Drastic reduction in errors
- Not an “adventure”

* Easier to debug production issues

- No “works of art”

Caveat

✂ Just because you can, doesn't mean you have to release everything all the time

Making it Happen

- * Testable hypothesis
 - The “why” of the features”
- * Decreased risk and increased cycle time allows for more experimentation
- * Also requires code to be readily changed
 - Internal software quality
 - Evolutionary architecture
 - Architect and design for testability

<http://www.thoughtworks.com>
<http://www.rebeccaparsons.com>
rjparson@thoughtworks.com

QUESTIONS?

ThoughtWorks®