Continuous Delivery

A methodology that allows you to deliver software faster and with lower risk

Too often, agile processes stop with Continuous Integration (CI) practices. However, CI does NOT equal Continuous Delivery (CD). True Continuous Delivery is present at every stage of the lifecycle.

Three key traits are evidence that actual Continuous Delivery practices are in use:

- Responded to questions about using basic CI practices:
  - Have automated >30% of infrastructure changes.
  - Frequently confirmed software updates are shippable.
  - Have basic monitoring in place.

- Teams can perform push-button deployments on-demand.

- Teams have continuous visibility into production readiness.

Achieving Continuous Delivery is Hard

For many, Continuous Delivery is still a work in progress.

Continuous Delivery Still Primarily in Development

Continuous Delivery isn’t a hard sell for developers. Continuous Delivery Still Primarily in Development

Culture and Lack of Time: The Major CD Barriers

CloudBees, Inc.

289 South San Antonio Rd

Los Altos, CA 94022

info@cloudbees.com

www.cloudbees.com

© 2015 CloudBees, Inc. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of CloudBees, Inc.

Continuous Delivery

Starts with Continuous Integration

CI is a jumpstart to enabling CD and definitely an important component to achieving true CD. But for CI to be really effective it needs to be extended beyond the code delivery processes to true CD.

Which CI server do you use for your development projects?

66% of respondents still use an infrastructure as code tool to respond to code changes in the software delivery process.

Continuous Delivery

The DevOps Way is Catching On

Continuous Delivery supports the DevOps philosophy of better collaboration across the organization.

Deployment Times Leave Room for Improvement

Once a code change is committed, how long does it take to get to the customer?

Some key traits to consider:

- Automated CI in place.
- Continuous visibility into production readiness.
- Push-button software deployment on-demand.

Achieving textbook Continuous Delivery is hard

When respondents were further filtered by the three key traits of CD, only 8% met all three criteria.

The not-so-good news:

The good news:

- 66% of respondents still have an infrastructure as code tool to respond to code changes in the software delivery process.

- 59% of respondents currently have a DevOps team.

- 27% of respondents have automated >30% of infrastructure changes.

The Major CD Barriers

- 6% use manual scripts for at least half of their infrastructure changes.

- 58% have some form of assembly automation.

- 73% have an automation pipeline in place.

- 52% manually deploy code to production.

- 60% use an infrastructure as code tool.

- 48% have a DevOps team.

- 38% use an infrastructure as code tool.

- 33% have automated >30% of infrastructure changes.

- 41% use basic CI practices.

- 32% have some form of assembly automation.

- 28% have an automation pipeline in place.

- 26% manually deploy code to production.

- 22% use an infrastructure as code tool.

- 20% have automated >30% of infrastructure changes.

- 19% use basic CI practices.

- 18% have some form of assembly automation.

- 17% have an automation pipeline in place.

- 14% manually deploy code to production.

- 12% use an infrastructure as code tool.

- 11% have automated >30% of infrastructure changes.

- 10% use basic CI practices.

- 9% have some form of assembly automation.

- 8% have an automation pipeline in place.

- 6% manually deploy code to production.

- 5% use an infrastructure as code tool.

- 4% have automated >30% of infrastructure changes.

- 3% use basic CI practices.

- 2% have some form of assembly automation.

- 1% have an automation pipeline in place.

- 1% manually deploy code to production.

- 1% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.

- 0% have automated >30% of infrastructure changes.

- 0% use basic CI practices.

- 0% have some form of assembly automation.

- 0% have an automation pipeline in place.

- 0% manually deploy code to production.

- 0% use an infrastructure as code tool.