

Continuous Delivery and Continuous Integration

Overview

- Definitions
- Background/History
- Continuous Delivery
 - How to practice Continuous Delivery
- Continuous Integration
- Continuous Integration Tools
- Continuous Delivery Summary

Definitions

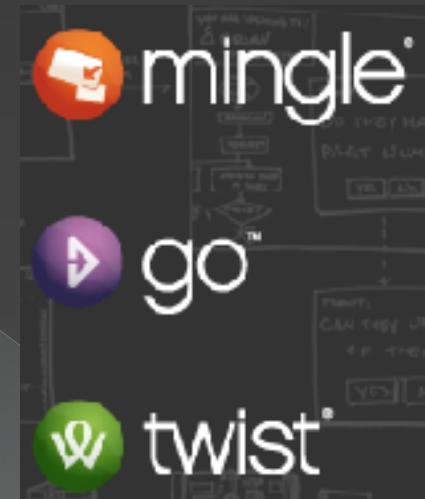
- **Agile Development Process:**
"An **iterative and incremental** (evolutionary) approach to software development which is performed in a **highly collaborative manner by self-organizing teams** within an effective governance framework with "just enough" ceremony that produces **high quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders.**" – Scott Ambler

Definitions Cont.

- **Agile Manifesto:** Principles that outline the Agile Process
- **Continuous Delivery:** New software development process that aims to build software that is always ready to be deployed into production
- **Continuous Integration:** Technique where every developer on a project must integrate their work daily with every other developer

Background/History cont.

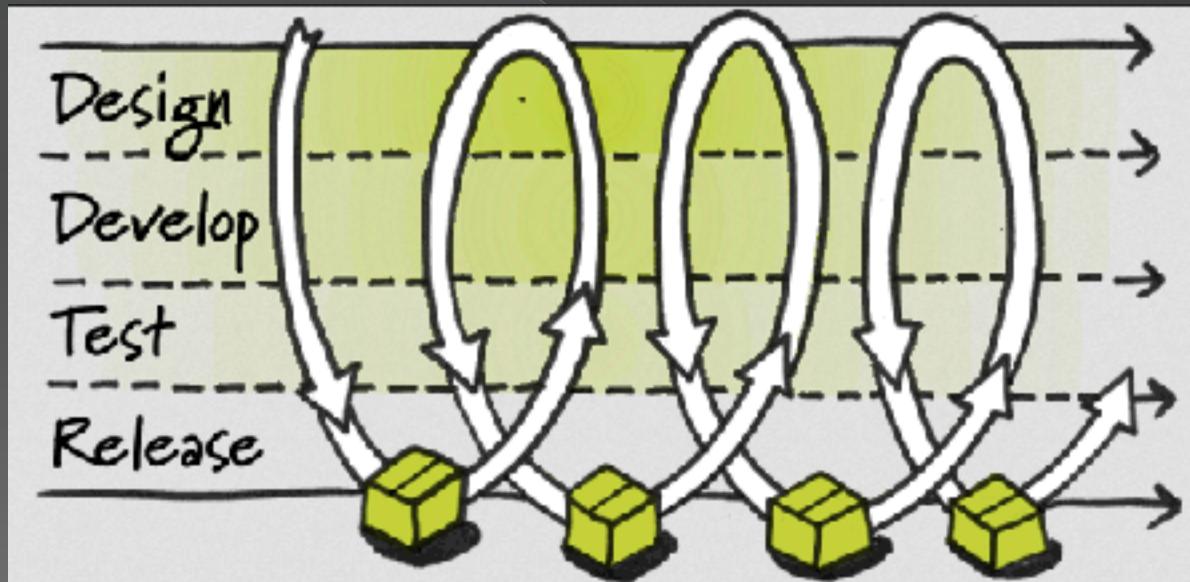
- ThoughtWorks Studios



- Continuous Delivery: Book written by Jez Humble and David Farley in 2010

Continuous Delivery

- Recall: Build software that is always ready to be deployed into production



Continuous Delivery cont.

- Is this a new idea?
- 1st principle of the Agile Manifesto:
 - > “Our highest priority is to satisfy the customer through early and **continuous delivery** of valuable software.”
- Goal: Release to production more often
 - > Monthly
 - > Weekly
 - > Daily

Continuous Delivery cont.

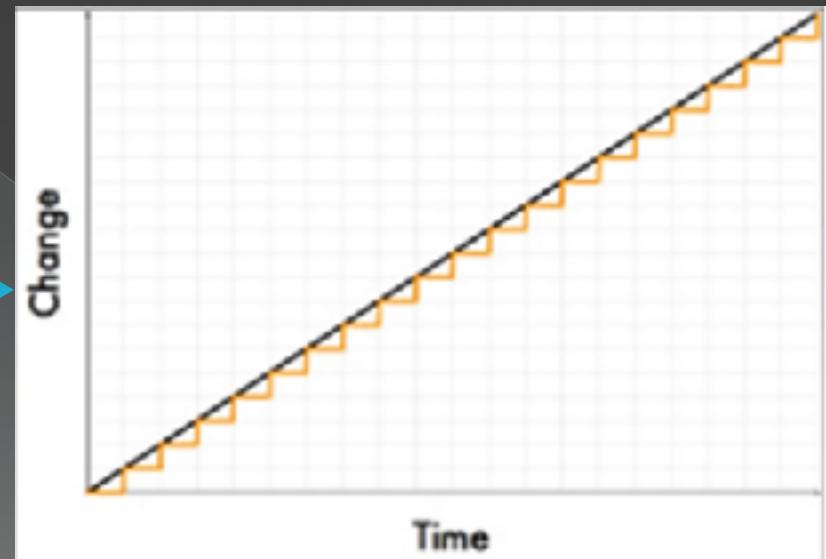
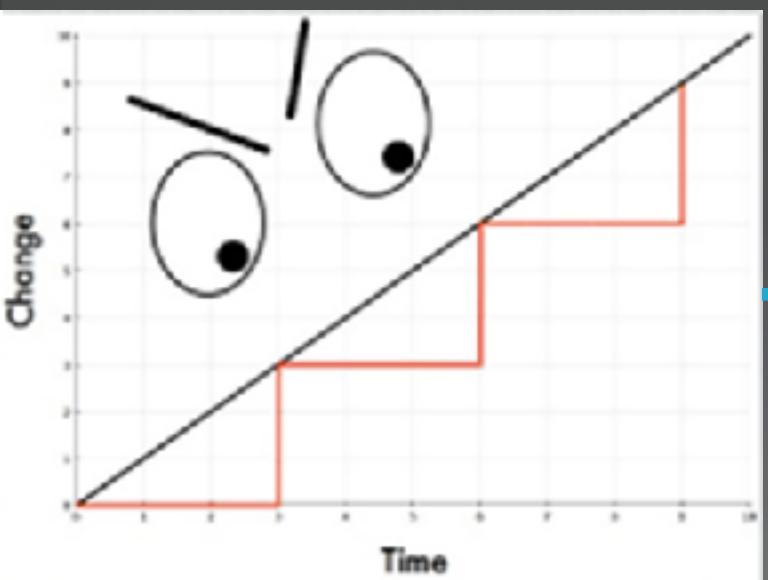
- ◉ Pros: Receive many benefits from the Agile process as well as others
 - › Build the right product
 - Constant feedback from customers
 - Write thorough acceptance tests
 - › Earlier benefits
 - Get product out before competitors
 - Find bugs earlier

Continuous Delivery cont.

- > Ability to react quickly to change
 - Not a huge amount of time and money can be abandoned to adopt new requirements
- > Save money and time
 - Save money and time if project goes bad
 - Save money and time with automation
 - As deploying to production becomes easier, more time can be spent developing valuable features

Continuous Delivery cont.

- Pros cont:
 - > Reliability



Practice Continuous Delivery

- Basics to start practicing Continuous Delivery
 - Configuration Management
 - What needs to be kept under CM?
 - Everything!
 - Code
 - Tests
 - Configuration Files
 - Build Scripts
 - Environments
 - Documentation
 - Etc.

CM Branching

- Stay away from branching except in special cases
 - › Branch on releases
 - › Horror story example
- You must always check into the trunk!
 - › Otherwise you're not continuously integrating

CM Branching

- What if your project team is releasing to production every week, but you're working on a feature that will take longer than the release cycle to implement?
- First option:
 - Gradually release feature into production
- Second option:
 - Feature toggle

Managing Environments

- Must have multiple environments when developing software
- Need to be able to duplicate environments with ease
- Environments configurations to take consideration of:
 - Operating systems including their framework and settings
 - Packages needed to be installed for the application to function properly
 - Network settings

Managing Environments

- Tools to help configure environments in an automated fashion:



Continuous Integration

- Vital step when practicing CD
- Recall: To practice CI, every developer on a project must integrate their work daily with every other developer
 - › Everyone on the team needs to practice this for it to work
- Continuous integration is **not a tool**, but a technique
 - › But there are many open source tools to help practice CI

Continuous Integration

- Things a CI server can do:
 - > After you check-in code it can:
 - Run build/test scripts
 - Notify developers of a failed build or failed tests
 - > Can show all past check-ins and if they failed or passed all the tests
 - > Can keep track of multiple projects
 - > Plus much more!!

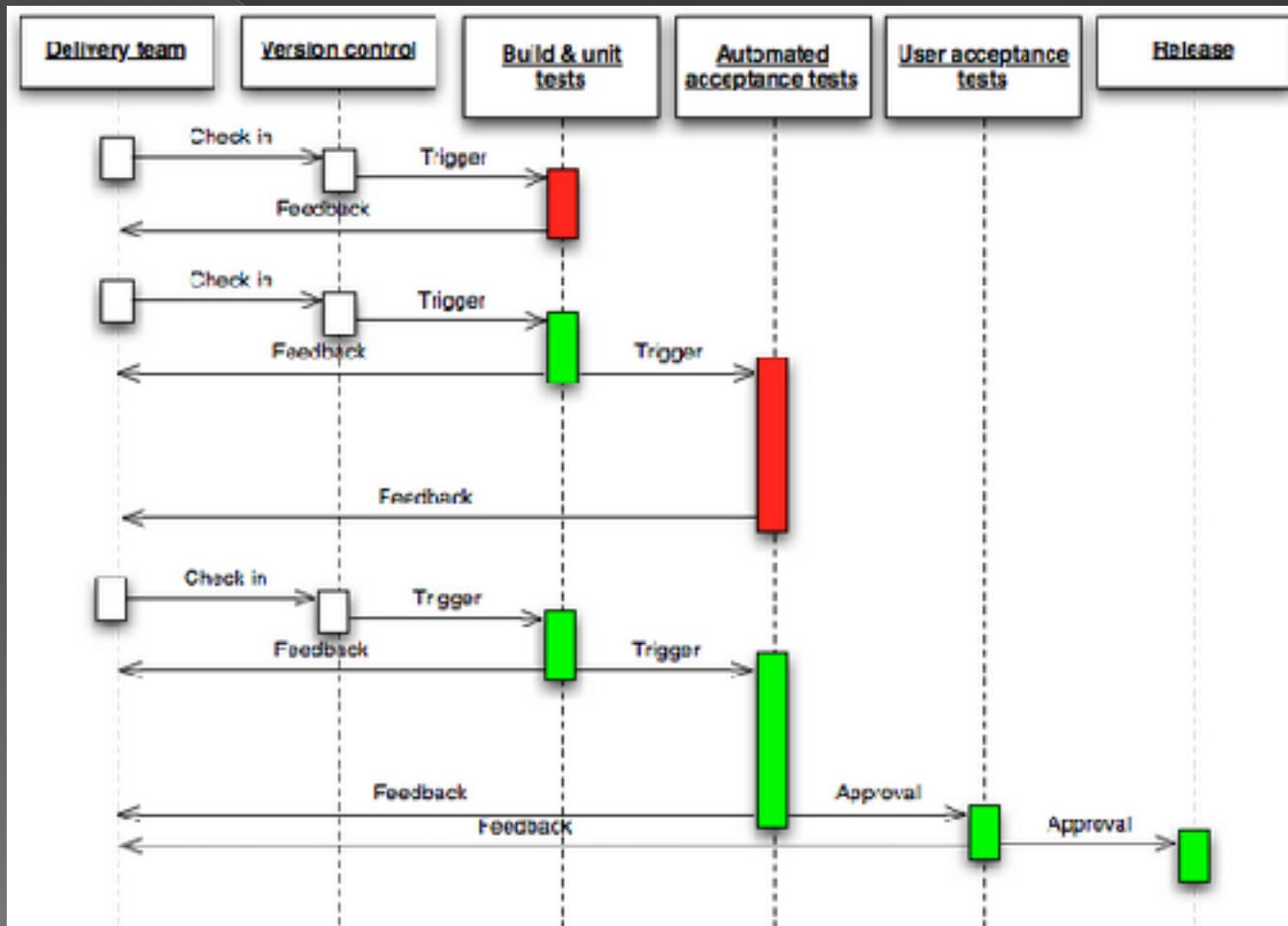
Demo Continuous Integration Tools

- Demo open source CI tools TeamCity and Hudson

Deployment Pipeline

- Heart of Continuous Delivery
- Will give immediate feedback for how ready your software is for production
- Automate everything(unit tests, acceptance tests, performance tests,...) to the point of just being able to click a button to deploy to production

Deployment Pipeline cont.



Deployment Pipeline in Go