Continuous Delivery

In the Enterprise

Class IV - Summary

- Continuous Integration Video Thoughts.
  - Comment with the class
- Build Automation Basics
  - Definition
  - Expected benefits
  - Common pitfalls
  - Tools
- Little Homework
  - Make your local CI work

Build Automation Definition

In the context of software development, build refers to the process that converts files and other assets under the developers' responsibility into a software product in its final or consumable form. The build may include:

- compiling source files
- packaging compiled files into compressed formats (such as jar, zip)
- producing installers
- creating or updating of database schema or data

The build is automated when these steps are repeatable, require no direct human intervention, and can be performed at any time with no information other than what is stored in the source code control repository.

Expected Benefits

Build automation is a prerequisite to effective use of continuous integration. However, it brings benefits of its own:

- Eliminating a source of variation, and thus of defects; a manual build process containing a large number of necessary steps offers as many opportunities to make mistakes
- Requiring thorough documentation of assumptions about the target environment, and of dependencies on third party products

Common Pitfalls

- The practice of build automation should not be confused with continuous integration: the latter consists of “executing” the build process as frequently as possible (ideally whenever a code change is checked into the source code control repository) and “verifying” the correctness of the resulting product, in particular by unit tests.
- In particular, continuous integration tools (CruiseControl, Jenkins, etc.) are a category distinct from build automation tools (make, Ant, Maven, rake, etc.)
- Being able to trigger some build operations from within a development environment (IDE) is usually not sufficient: as it is often the case that some build operations are not supported within the IDE, it must be possible to perform a build outside of the IDE.
- The duration of a build process should be under ten minutes, including the execution of automated tests; beyond this order of magnitude it will generally be difficult for the team to achieve continuous integration.

Tools

A sample list of common used tools for automate builds:

- Make
- Ant
- Maven
- Gradle
- Gulp / Grunt
- Rake
- Webpack
- npm

Let’s eat and digest...

- Let’s watch this video: https://www.youtube.com/watch?v=hjIlTaAMsbI
    - Understand it and discuss.

Homework

- Make your CI work in Jenkins.