Testing Spring

Beginner to Guru

Beyond Testing with CI / CD

CI - Continuous Integration

- **Continuous Integration** (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early. - ThoughtWorks
  - *"Continuous Integration doesn't get rid of bugs, but it does make them dramatically easier to find and remove."* - Martin Fowler, Chief Scientist, ThoughtWorks

CI Practices

- CI Practices per Martin Fowler:
  - Maintain a Single Source Repository
  - Automate Build
  - Make Your Build Self-Testing
  - Every Commit Should Build on Integration Machine
  - Fix Broken Builds Immediately
  - Keep the build fast
  - Test in a Clone of the Production Environment
  - Make it Easy for Anyone to get the Latest Executable Version
  - Everyone Can See What is Happening

Common CI Build Servers

- **Self-Hosted**:
  - Jenkins, Bamboo, TeamCity, Hudson
    - Fun-Fact - Jenkins forked from Hudson in 2010 due to a legal conflict with Oracle
- **Cloud Based**:
  - CircleCI, TravisCI, Codeship, GitLab CI, AWS CodeBuild
  - And many, many more

CD - Continuous Deployment

- Continuous Deployment will automatically deploy build artifacts after all CI tests have run.
- Should Happen with every Commit
- Completely Automated
- May Include a Staging Area from which Additional Automated Tests are run
- Easily Confused with Continuous Delivery
- Example: AWS CodePipeline

CD - Continuous Delivery

- Process to Automatically Deliver code changes directly to the Production Environment
- Involves a High Degree of Automation in Testing and Deployment
- Must have a *VERY* Mature Process
- Can be Difficult in Some Industries due to Regulatory Requirements
- This area is evolving.
  - Few Hard “Rules” - No Standard Way
  - “Best Practices” are maturing, and still a lot of lively debate!

SPRING FRAMEWORK

GURU