Continuous Integration

Improving Software Quality and Reducing Risk

Paul M. Duvall
CI Value

• Reduce risks
• Reduce repetitive manual processes
• Generate deployable software at any time and any place
• Enable better project visibility
• Establish greater confidence in the software product from the development team

Introduction

• Build software at every change
• Continuous database integration
• Continuous testing
• Continuous inspection
• Continuous deployment
• Continuous feedback
Build Software at Every Change

- Automate builds
- Perform single command builds
- Separate build scripts from your IDE
- Centralize software assets
- Create a consistent directory structure
- Fail builds fast
- Build for any environment
- Use a dedicated integration build machine
- Use a CI Server
- Run fast builds
- Stage builds

Continuous Database Integration

- Automate database integration
- Use a local database sandbox
- Use a version control repository to share database assets
- Give developers the capability to modify the database
- Make DBA part of the development team
Continuous Testing

- Automate unit tests
- Automate component tests
- Automate system tests
- Automate functional tests
- Categorize developer tests
- Run faster tests first
- Write tests for defects
- Make component tests repeatable
- Limit test cases to one asset

Continuous Inspection

- Reduce code complexity
- Perform design reviews continuously
- Maintain organizational standards with code audits
- Reduce duplicate code
- Assess code coverage
Continuous Deployment

- Release working software any time, any place
- Label a repository’s assets
- Produce a clean environment
- Label each build
- Run all tests
- Create build feedback reports
- Possess capability to roll back release

Travis

https://travis-ci.com/
https://docs.travis-ci.com/
Travis Introduction

- automatically build and test code changes
- provide immediate feedback on success of changes
- automate other parts of the development process
- automate other parts of delivery workflow
- process
  - clones GitHub repo to a new virtual environment
  - carries out specified tasks: build, test, inspect
  - build passes if no task fails, deploy code to S3, Heroku

Travis Supports

- GitHub public and private repositories
- Many languages
- Multiple operating systems
- Containers
- Isolated virtual machines give clean state for build/test
- Services to test databases, messaging, headless browsers, etc.
Travis-CI for CS 314

- pom.xml (Maven configuration)
- .travis.yml
- build, test, inspect, deploy

Pull request build flow

A pull request is created
GitHub tells Travis CI
the build is mergeable
Hooray!
Your build passed!
Travis CI updates the
PR that it passed
You merge in
the PR goodness