CI/CD
Lecture 3-Continuous Integration

Omkarendra Tiwari

August 21, 2022

Frederick Brooks

Highlights

- ACM Turing Award-1999
- 8-bit byte
- System Interrupts
- Mythical Man Month (Book)

A scientist builds in order to learn; an engineer learns in order to build.
-Fred Brooks

Image:https://amturing.acm.org/award_winners/brooks_1002187.cfm

Outline

Developer's day

- New feature to next feature

Continuous Integration

- What, Why, When, and How

Continuous Integration Practices

- Overview and Discussion

CI: Beginning of Automation Pipeline

Developer's day

1. Develop a feature
2. Integration locally
    1. Pull the changes by others, if any
    2. Resolve the conflicts, if any
3. Build
4. Unit test
5. Commit to central repository
6. Build on integration server
7. Testing
8. Feedback
    1. Passed: Move to next feature
    2. Failed: Improve the code

Continuous Integration (CI)

What

- A DevOps software development Practice
- A culture of integrating changes frequently
- An automated process of Integration, Build, and Unit tests

Why

- Shorter feedback time
- Quicker to fix bugs
- Faster release/iteration
- Reduced risk

When/How

- Culture : Agile/XP
- Discipline : commit per hr/day
- Tools : Git, Jenkins

Elements of CI Workflow

Developer

- Develop a feature
- Unit test
- Build locally
- Push changes

Remote Repository

- Store version history

Continuous Integration Server

- Continuously monitor for new commits
- Checkout new changes
- Build (Make executable and Test)
- Notify Developers and others

CI Practices: Overview

How to make it effective

- Single source repository
- Automate build
- Make your build self-testing
- Commit frequently; everyday if possible
- Every commit should build the mainline
- Fix broken builds immediately
- Make your build fast
- Test in a clone of production environment
- Make it easy for everyone to get the latest executable
- Visible progress

CI Practices: Discussion

Single source repository

- Use a VCS
- Place all files in it
  - Source code
  - Test cases
  - Third party libraries
  - Install scripts
- Maintain single mainline branch
- If needed, allow another for bug fixes

Automate the Build; Self-Testing

- Build: Transformation from source code to executable
- Compilation catches a few bugs

  However
- Including Tests would increase the scope
- Automate the execution of Tests
- A failed Test should result in Build fail

CI Practices: Discussion

Frequent Integration

- Commit/integration is communication to others
- Requires breaking the task in smaller chunks
- Conflicts/Bug resolution is quicker

Everyone can see the state

- Communication is important
- State of Machine build is of particular interest
- Who/what changes were made
- Is any build in progress

Build on Integration Machine

- Local build is not sufficient
- Ensure to build on Integration machine

Build Fast

- Saved time in Build affects each commit/developer
- Usual bottleneck is Testing-as it requires external services (e.g. Database)

Further Readings

Highly Recommended

- https://www.martinfowler.com/articles/continuousIntegration.html

Recommended

- https://continuousdelivery.com/