PNSQC

OCTOBER 9-11 2023

AMP IT UP:

TRANSFORMING QUALITY

Brent Clausner

Software Continuous Integration with Hardware

Hello

- Who I am
- Introduction
- DevOps Current Practices
- Software for Hardware
- Emulation
- Code Branches
- Drawbacks
- Wrap up

Myself

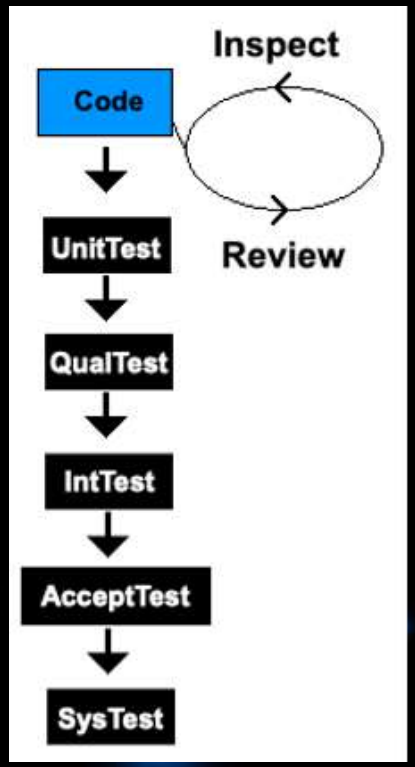
- Education
  - PTI
  - Point Park University
- Work
  - DevOps Engineer
- Hobbies
  - Video Games
  - Jogging
  - Technology

Introduction

- Testing is important
  - Regular basis
  - Every code submission
- Hardware
  - Equipment Costs
  - Contention for use
  - Downtime for hardware refresh
- Functional Dependencies
  - Test all areas
  - Use all current changes

Current DevOps

- Develop Code
  - Build
  - Pipeline for Static Analysis
- Inspect/Review
- Test
  - Unit Testing
    - Language based frameworks
  - Integration Testing
  - System Testing



Current DevOps Cont'd

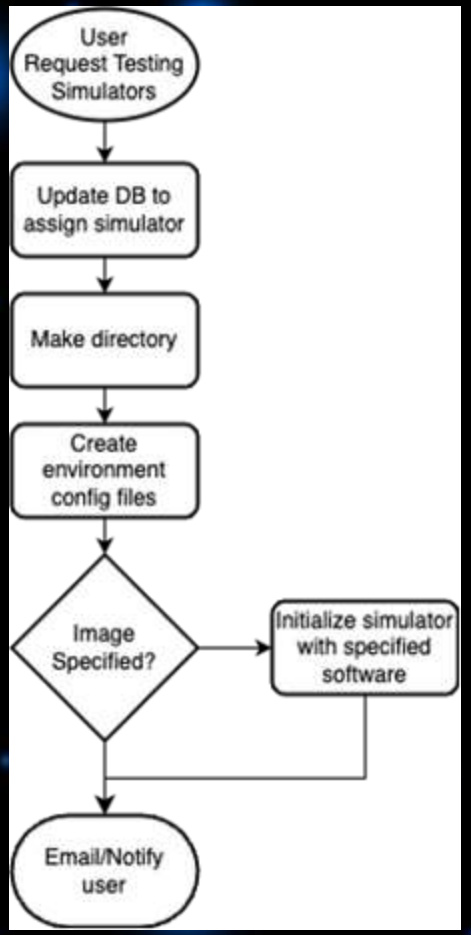
- Architecture(s)
  - x86_64 or arm64 (Examples)
  - Cross platform builds
  - Parallel building
- Functional Areas
  - Most features touch several areas
- Example
  - REST API
    - Authentication (Every endpoint)
    - Database update
    - Bring service up
    - Responses

Software for Hardware

- Connecting
  - rconsole
- Loading Image onto the Hardware
- Timing
  - Initialization Time
  - Asynchronous Commands
- Capturing output
- Errors
  - Console only errors
- Hardware inside pipeline

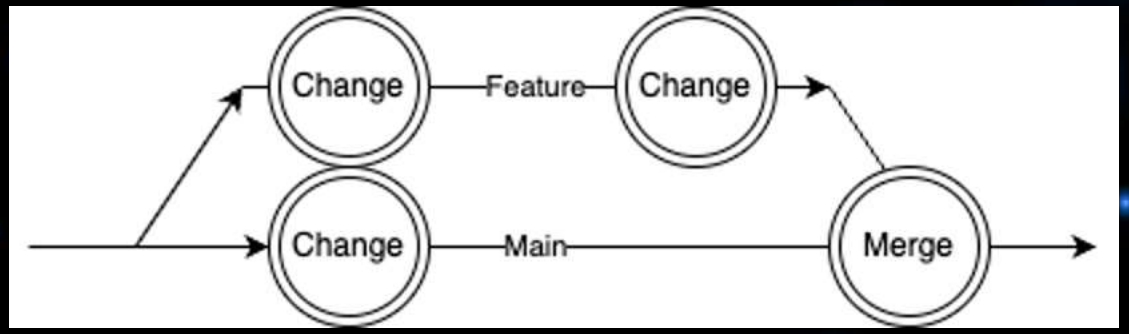
Emulation

- Cost savings
  - Hardware
  - Time savings
  - Team sharing
- Scale
- Sample Testbed Creation
- Templates
  - Benefit multiple teams
  - Expand to test suites
  - Changes benefit all suites
  - Test in parallel threads



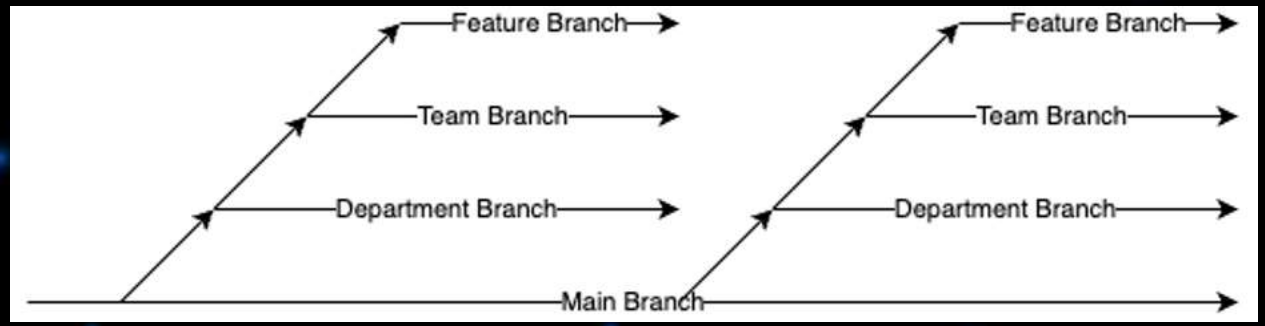
Code Branches

- Single Main
  - Simple
  - Easy of use
  - Most small projects
  - Backing out changes
  - Merges
  - Testing on each submission
  - Sync from main
- Requires tagged version branches
  - Long term support



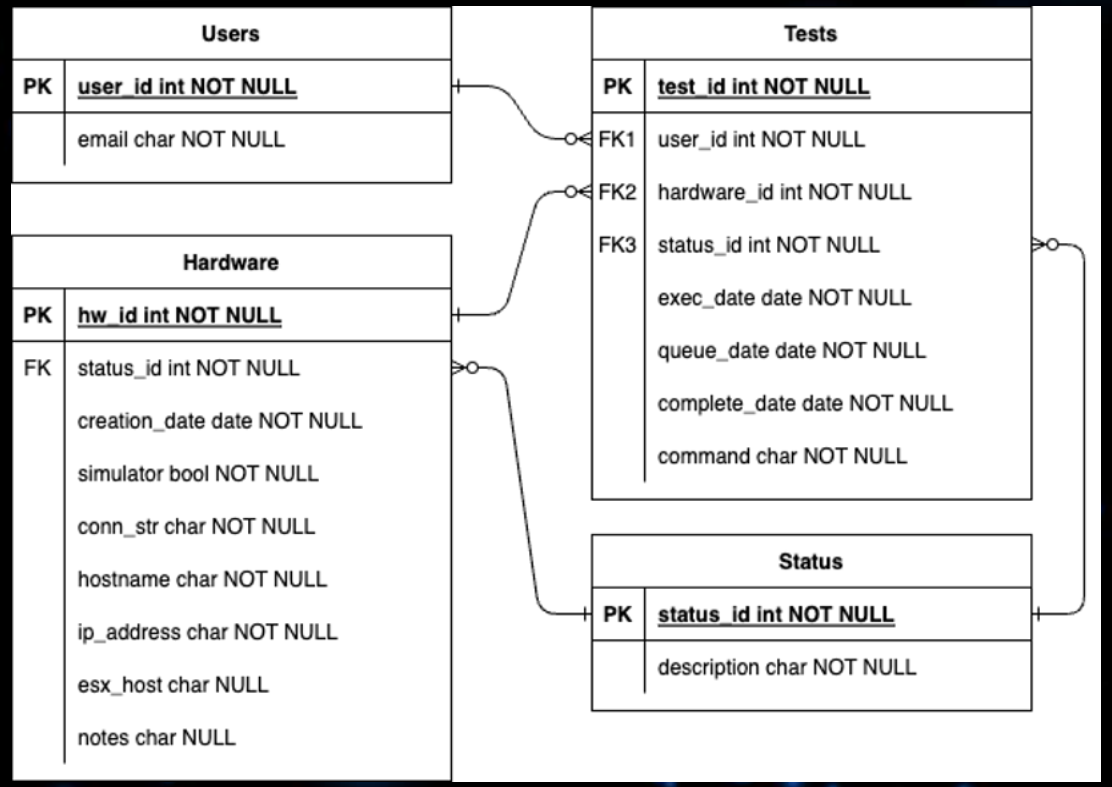
Code Branches Cont'd

- Complex branching
  - More teams more problems
  - Slow to merge up
  - Cherry picking changes
  - Problem changes impact
  - Testing takes longer
- More teams
- Testing every Change
  - Large workload for concurrent changes
  - Sequential Queue



Code Branches Cont'd

- Single main branch
  - Smaller feature branches
- Testing Periodically
  - Time based cron
  - Find bad changes via sorting algorithms
  - Bulk good changes validated quicker
- Handle large amounts of changes
- Better picture of quality
- Use Emulators for scaling testing
- Sample DB Schema



Drawbacks

- Delays
  - Intermittent failures
  - Infrastructure failures
  - Multiple bad changes
    - Prevent any changes during the find phase
- Don't wait to submit before vacation
- Custom Infrastructure
  - Team to support
  - Time investment
- Time to be approved
  - Can be shortened as Developers learn the system

Wrap up

- Write tests for areas you care about
- Reduce costs
  - Emulators
  - Test more frequently
- Prevent bad changes quicker

Questions?

???

References

- Tim Menzies, William Nichols, Forrest Shull, & Lucas Layman (Retrieved on 2023, May 29) Are delayed issues harder to resolve? Revisiting cost-to-fix of defects throughout the lifecycle. Retrieved from: https://link.springer.com/article/10.1007/s10664-016-9469-x
- David Davis (Retrieved on 2023, May 30) What is VMware ESXi Server and Why do I Need It? Retrieved from: https://www.pluralsight.com/blog/it-ops/what-is-vmware-esx-server-and-why-you-need-it
- Install GitLab Runner (Retrieved on 2023, May 30) Retrieved from: https://docs.gitlab.com/runner/install/index.html
- Stable Diffusion web UI: https://github.com/AUTOMATIC1111/stable-diffusion-webui

THANK YOU

Brent Clausner

Software Continuous Integration with Hardware