Integration

Moorthy

Integration

Created by Moorthy / @mskmoorthy and modified by Wesley Turner / @wd-turner

Reading Material

- ctest
- github continuous integration
- github templates

Integration

- Systems Integration - Hardware, Firmware, Software
- Hardware Integration - components wired together
- Software Integration - Final step before acceptance and Testing
- We will deal with only Software Integration.

Purpose

- Combining two or more software units
- Verifying combination by testing
- Heterogeneous components
- Mix of functions and operating modes

Why

- New Problems will inevitably surface.
- If done poorly all problems will surface at the same time.
- Cascade of interdependencies.

Types Of Integration

- Phased Integration ("big bang" integration - each unit separately and combine them all)
- Incremental Integration (Develop a functional "skeleton" system - design, code, test and debug a small new piece, integrate it before adding any other piece

Benefits of incremental

- Errors easier to isolate, find, fix reduces developer bug-fixing load
- System is always in a relatively working state - good for developers and customers.
- System complexity can be understood incrementally.
- Users can develop a better understanding of their requirements.

Drawbacks of Incremental

- May need to create "stub" versions of some features that have not yet been written.
- With many developers, the incremental integration may become a "big-bang" integration.

Types of Integration

- Top Down Integration - Start with Outer UI layers and work inward
- Bottom-up Integration - Start with low level layers and work outward.
- "Sandwich" Integration - connect top level UI with crucial bottom-level classes (add middle layers later as needed)
- Continuous Integration - Maintain a single source repository (automate the build, make your build self testing, every one commits to main line every day.

More on Continuous Integration

- Every commit should build mainline on an integration machine
- Keep the build fast
- Test in a clone of the production environment
- Make it easy every one to get the latest executable
- Automate Deployment

Daily Builds

- Daily Build: Compile working executable on a daily basis
- Continuous Integration Server - An external machine that pulls down your latest repo and builds all resources

Automated Tests

- Automated tests
- Smoke Tests

Stubs

- A controllable replacement for an existing software unit to which your code under test has dependency
- Useful for simulating difficult-to control events
- network/internet, database, threads, memory
- Also useful for legacy systems

"Mock" Objects

- Mock object is a fake object that decides whether a unit test has passed or failed by watching interactions between objects
- Useful for interaction testing
- Statement Coverage - All statements have been executed.

Software Support

- cmake with ctest
- travis ci - github dual license
- jenkins ci open source
- circle ci
- github actions
- gitlab

Walkthrough

- github actions

The End

by Moorthy