Continuous Integration of Mesa

Lessons learned at Intel

Defining Continuous Integration

Martin Fowler: “Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.”

Developer Tests

- Ensure the stability of the project
- Find your own bugs
- Defend your work against subsequent breakage
- Push with confidence

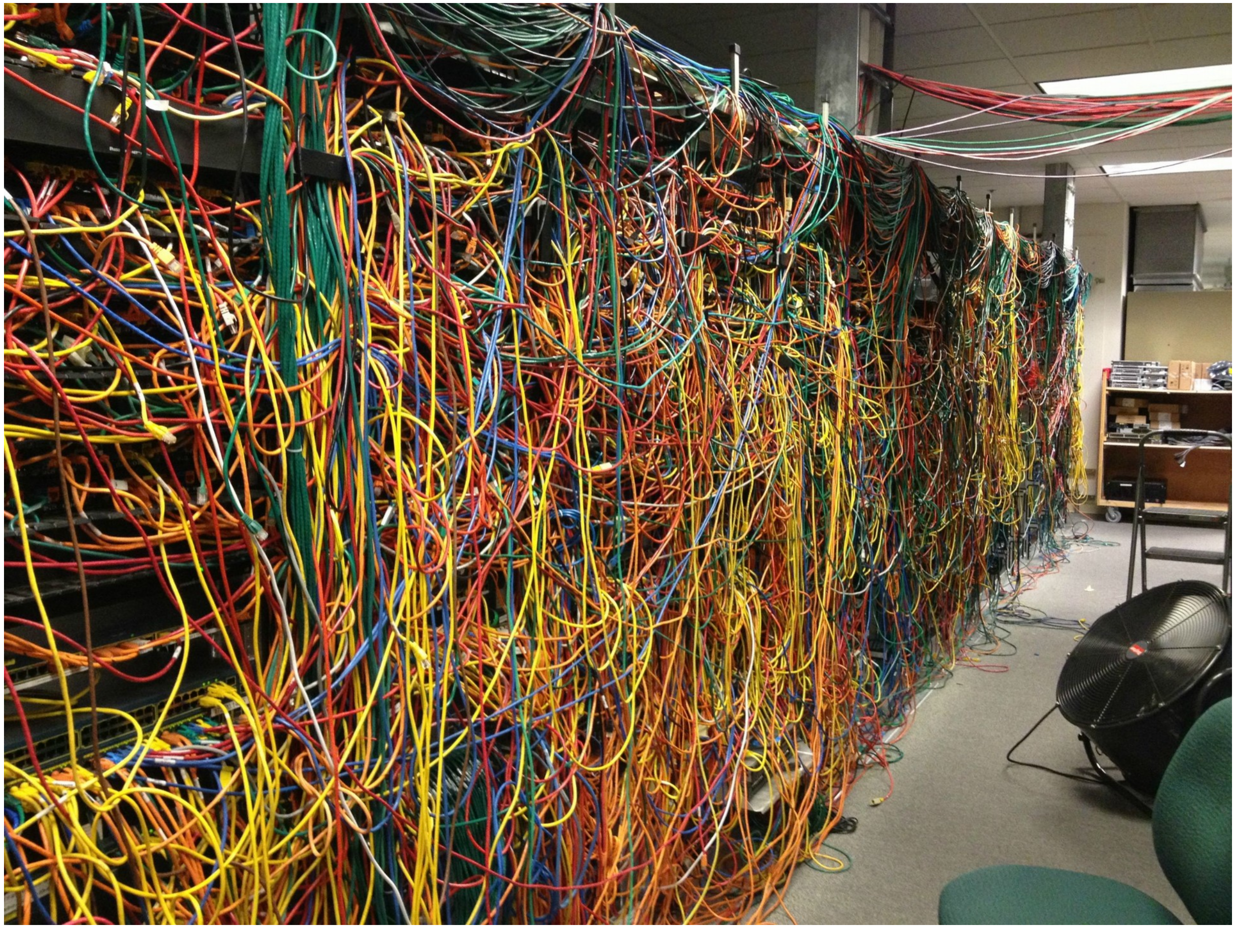
Developer Test Suggestions

- Use a test framework
- Make test a 1st class artifact of the development process
- Prioritize test reliability
- Prioritize test run time

CI Suggestions

- Top priorities should be reliability and run time
- Automate and standardize machine setup
- Support branches
- Leverage standard tools, don't depend on them
- Don't track bugs in CI

Intel's Mesa CI Lab



CV LAB
BAY #2

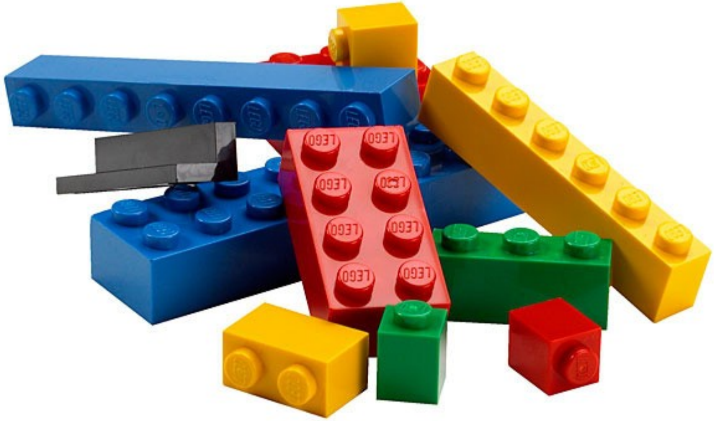
apw

Intel's Mesa CI Jenkins

- Developer branches
- Stable branches and Master
- Custom builds
- Daily/percheckin builds
- Component summary
- Long pole analysis

Intel's Mesa CI Automation

Component and Scheduler pattern



http://github.com/janesma/mesa_jenkins

Intel's Mesa Practices

- Reliability and run-time
- Automated machine setup
- Branch support
- Standard tools, and tool neutrality
- Bug tracking vs Regression identification

Discussion