# Continuous Integration (CI)

… with nightly builds, without deployment

# Roadmap

- Who am I and why
- Problem definition
- Basics (What we need)
- Benefits
- Best practices
- Tools
- Buildbot
- Buildbot in action

# Who am I and why

- Name: Rafael Sadowski
- Bachelor: Applied computer science
- UNIX / OpenBSD user
- 3 years pure C++ / Qt4 developer
- 2+ years buildbot administrator
- Experiences with large code base (larger than Firefox or Chromium)

# Problem!?

1. Slow release process
2. Poor visibility
    a. Project
    b. History
    c. Tests
3. Issues raised are harder to fix or find
4. Writing good, useful tests is hard!
    a. and running easy?

# Basics (What we need)

- Version control system
  - GIT, Subversion ....
  - Store (almost) everything in the VCS
    - Complete code, build scripts, tests, lib depends
- Automated Build
- Automated Test
  - Unit tests, performance ...
- Process / Rules
  - compiler
  - failure handling

# Benefits

- Automating the build and test process gives developers immediate feedback on their work.
- Tests run early
- No integrations points
- Tests can run on multiple platforms
- Measurable and visible code quality
- Easy to track
- Continuous automated regression unit test

# Best practices

- Single source repository
- Automate the build and test
- everyone commits every day
- Keep the build fast
- Transparency (see everything)

## Google Speed and Scale

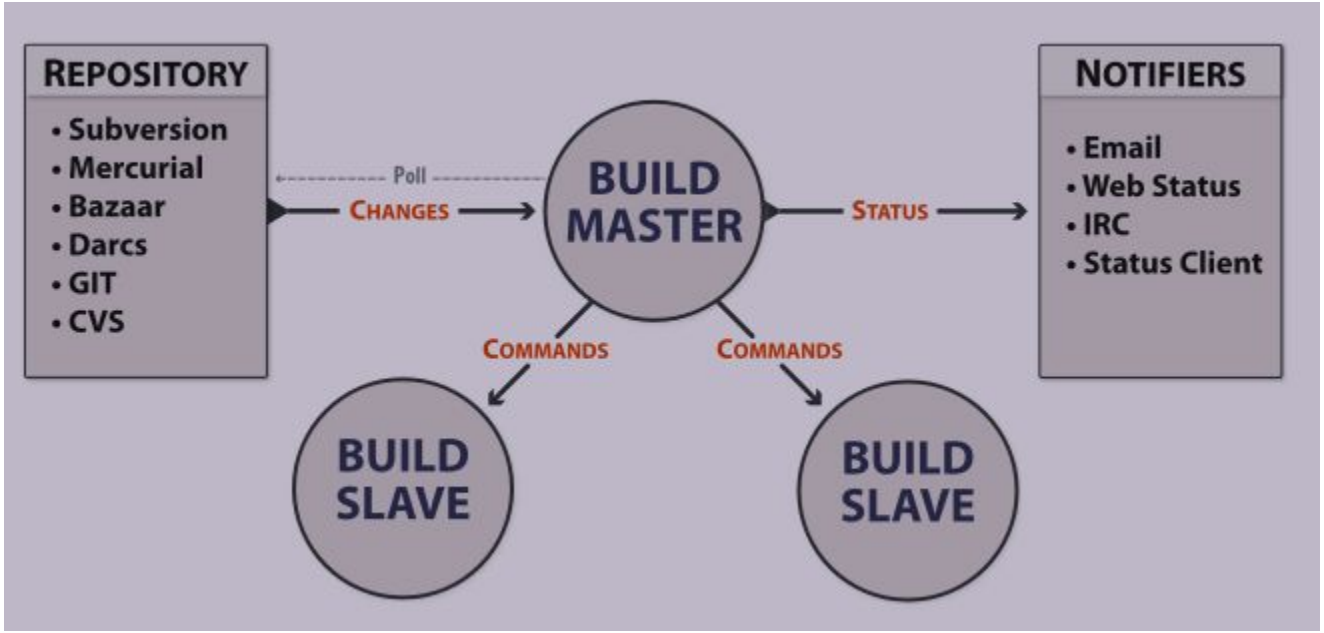
- 15000+ developers in 40+ offices
- 4000+ projects under active development
- 5500+ submissions per day on average
- Single monolithic code tree with mixed language code
- Development on one branch - submissions at head
- All builds from source
- 20+ sustained code changes per minute with 60+ peaks
- 50% of code changes monthly
- 75+ million test cases run per day

# Tools

- Jenkins/Hudson
- AnthillPro
- CruiseControl
- Team Foundation Server
- Buildbot

# Buildbot

Buildbot is an **open-source framework** for automating software build, test, and release processes.



# Why Buildbot

- Many CI tools, such as CruiseControl or Jenkins, are structured as ready-to-use applications.
  - Buildbot not! Good? Bad?
  - Good: **The fundamental design is not fixed and options are not limited.**
  - Bad: **Not ready-to-use and need time**
- Buildbot's design allows your installation to grow with your requirements
- Famous buildbot user: LLVM, Python, GDB, Mozilla,Google Chromium
  - http://trac.buildbot.net/wiki/SuccessStories

# Buildbot in action

LLVM: http://lab.llvm.org:8011/waterfall

GDB: http://gdb-build.sergiodj.net/waterfall

VideoLAN: http://buildbot.videolan.org/waterfall