

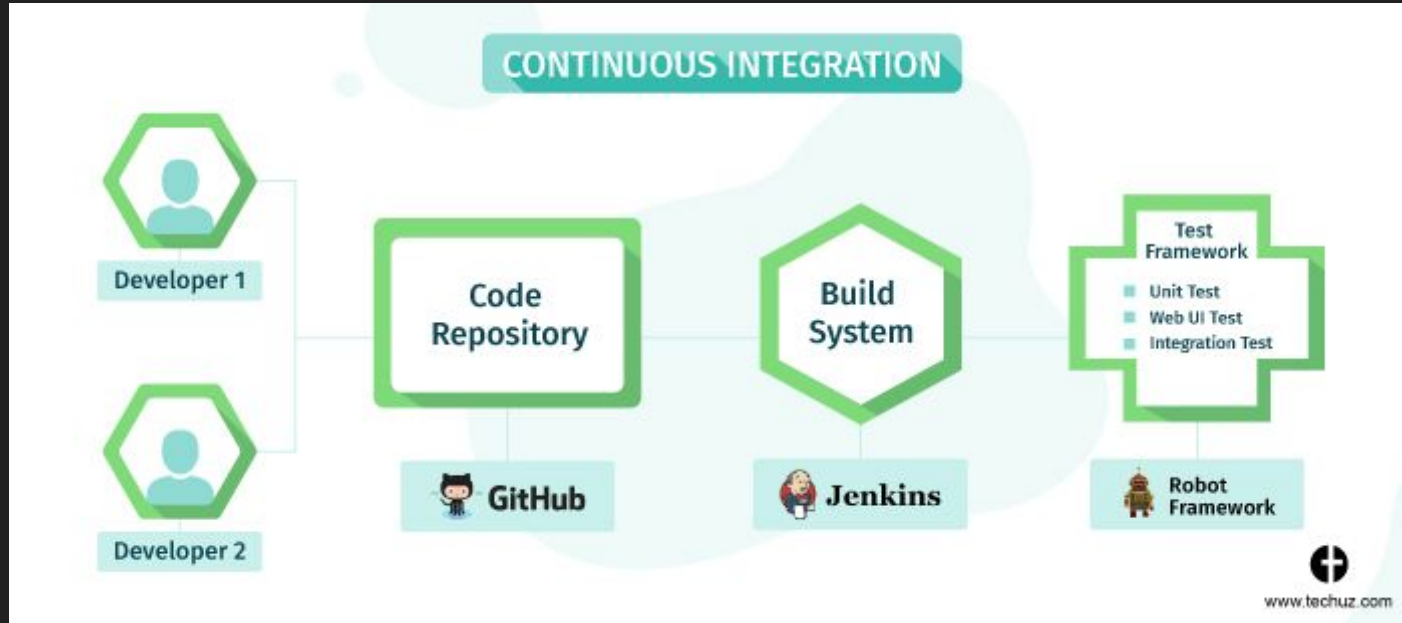
# Google's CI Practices

Michael Richter, Angela Mattamana, Paul Blackburn,  
Charlie Nguyen

# What is CI?

- CI is Continuous Integration
- The practice of integrating all developers' working copies to a shared mainline several times a day.
- Allows teams to detect problems early
- With CI, possible to more frequently cut release candidates.
  - Why? Every commit is a release!
  - No more QA/Staging environment

# CI



# History

- Grady Booch first proposed CI in his 1991
  - No integrating several times a day
  - Did not make it a part of build process for an entire application
- With advancement, automation of building applications became important
  - For version control
  - For cleaner code

# Industry Norms

- Jenkins
  - Job control
    - Master node
    - Builds are built on slave node(s)
  - Build trigger on commit
    - Unit tests must pass for commit to be accepted
  - More rigorous test suites may be run as part of a nightly regression test
    - Robot framework
    - Build at a certain time like a cron job



**Jenkins**

# Jenkins

The screenshot shows the Jenkins web interface. At the top left is the Jenkins logo. To its right is a search bar and the text "Jenkins Admin | log out". Below the search bar is a link "ENABLE AUTO REFRESH". On the left sidebar, there are several menu items: "New Item", "People", "Build History", "Manage Jenkins", "Credentials", and "My Views". The main content area features a table of build jobs. The table has columns for "S" (Status), "W" (Weather icon), "Name", "Last Success", "Last Failure", and "Last Duration". Two jobs are listed: "OpenShift Sample" and "Test OpenShift Job". Below the table, there are links for "Icon: S M L" and "Legend", and three RSS feeds: "RSS for all", "RSS for failures", and "RSS for just latest builds".

Jenkins Admin | log out

ENABLE AUTO REFRESH

[add description](#)

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		<a href="#">OpenShift Sample</a>	N/A	N/A	N/A	
		<a href="#">Test OpenShift Job</a>	14 hr - #1	N/A	12 sec	

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

**Build Queue** —

No builds in the queue.

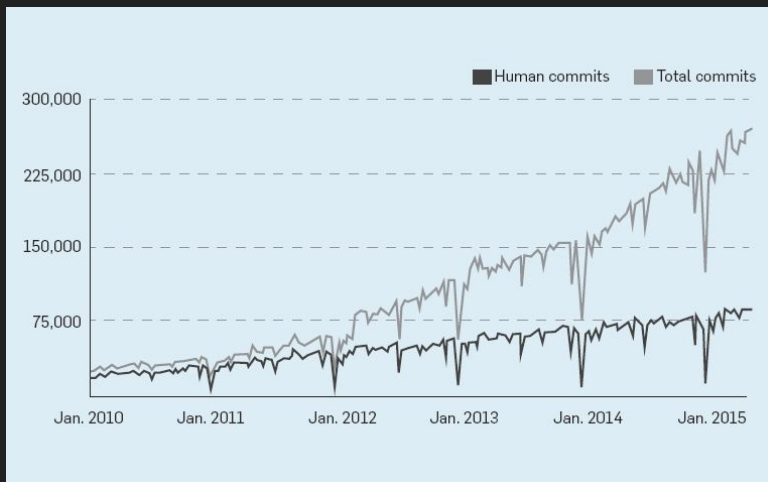
**Build Executor Status** —

[jenkins-slave-5-hz9if-0253fc22](#)  
1 Idle

[jenkins-slave-5-p2oih-6d33f3d0](#)  
1 Idle

# Google's CI Practice

- Stores billions of lines of code in a single repository
- Been this way for 16 years
- Repository has 1 billion files, 35 million commits, thousands of developers
- 86 TB of data
- Approximately 40,000 commits to codebase daily



# Advantages

- Unified version - one source of truth
- Extensive code sharing and reuse
- Simplified dependency management
- Atomic changes
- Large-scale refactoring
- Collaboration across teams
- Flexible team boundaries and code ownership
- Code visibility



# Concerns

- Have to create and scale tools for development and execution
- Code discovery
- Notifications
  - Firehose (unsolved so developers turn off notifications)
- Too many unit tests.
  - Bazel. The build tool can be asked “what depends on this code” in a way that scales.
  - Run only unit tests that are connected.
- Security
- Unreasonable for developer to have entire repo on their machine
  - Perforce (proprietary) uses a virtual file system

Questions?