Searching for Build Debt
Managing Technical Debt at Google

J.D. Morgenthaler, M. Gridnev, R. Sauciuc and S. Bhansali
Google's Build System Overview

- Single, Monolithic Source Repository
- Single, Global Build System
- Single Continuous Integration System
Google's Build System Debt

- Dependency Debt
- Visibility Debt
- Zombie Targets
- Dead Flags
- Other Discoveries
Dependency Debt

- Declared dependencies have to be manually kept in sync with source
- Over-declared dependencies waste resources
- Under-declared dependencies hinder progress
Example Under-Declared Dependency

//project:main calls Bar.getX()
//project:direct_dep uses Bar
//indirect/dependency:rule defines Bar

/project/BUILD:

foo_binary(name = "main",
            deps = ["//:direct_dep"])

foo_library(name = "direct_dep",
            deps = ["//indirect/dependency:rule"])
Example Over-Declared Dependency

```
//project:main

//project:direct_dep

//indirect/dependency:rule

calls Bar.getX()
stops using Bar
defines Bar

/project/BUILD:

foo_binary(name = "main",
            deps = ["":direct_dep")

foo_library(name = "direct_dep",
            deps = ["//indirect/dependency:rule"]
```
Remove Over-Declared Dependency Dependency

//project:main

calls Bar.getX()

//project:direct_dep

defines Bar

//indirect/dependency:rule

/project/BUILD:

foo_binary(name = "main",
            ## BROKEN
deps = ["direct_dep"])

foo_library(name = "direct_dep",
            deps = [])
Treatment Philosophy

- Automate
- Make it easy to do the right thing
- Make it hard to do the wrong thing
Treatment of Under-Declared Dependencies

- Educate engineers
- Automate addition of under-declared dependencies
- Use build system to prevent reoccurrence
Results

- Tools adopted by several large projects
- Engineer pushback
- Uncovered additional technical debt
Visibility Debt Cleanup Results

- Changed default target visibility to private - 2011
- Poor education increased change aversion
- Engineer pushback overcome by management
- Remaining debt slowly being paid down
Zombie Target Cleanup Results

- Daily tracking of long-term broken targets (<1%)
- Identification in code search UI
- Semi-automated cleanup had little impact
Dependency Debt Removal Detail

- Language-specific solution (Java)
  - Build system partitions classpath elements (jars) into direct and indirect based on dependencies
  - Extend `javac` to determine the jar from which each referenced class was loaded
  - Issue warning when indirect jars referenced
  - Enforce: `strict_java_deps` build rule attribute