Automatic Building of Java Projects in Software Repositories: A Study on Feasibility and Challenges

Foyzul Hassan*, Shaikh Mostafa, Edmund S. L. Lam, Xiaoyin Wang

Why Automatic Building?

Necessity of large corpus of built software

1. *Large Scale Program Analysis.*
    I. *Points-to analysis.*
    II. *Call-graph generation.*
    III. *Dependency Analysis. etc.*
2. *Mining of software artifacts.*

Single Project Build Requires to follow lot of steps. If we want to build 1000 projects, Bang!

Study Subjects

Top 200 Java Projects Based on Popularity

maven

gradle

Build Systems We Considered

How We Can Build Automatically?

Running Default Build Command

Ant	• *ant build*
Maven	• *mvn compile*
Gradle	• *gradle build*

Build With Default Build Command

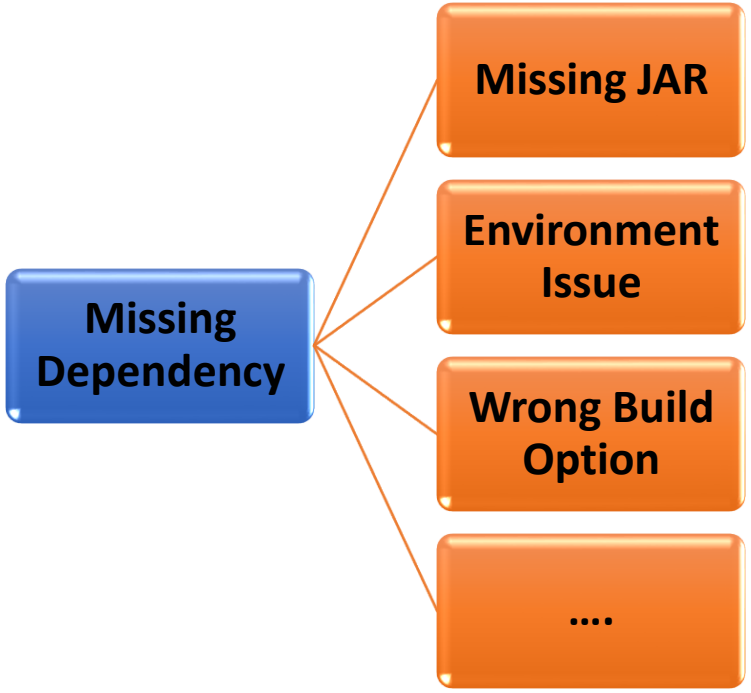
99 of 200 top Java projects cannot be built successfully with default build commands.

Other Study Vs Our Study

```
$ gradle build -x test --daemon
:compileJava

FAILURE: Build failed with an exception.

* What went wrong:
Could not resolve all dependencies for configuration ':compile'.
> Could not find mysql-connector-java-bin.jar (mysql:mysql-connector-java:5.1.16
).
```



Our Study

Our Study *"Automatic Building of Java Projects in Software Repositories: A Study on Feasibility and Challenges"* focuses on

Build Failure Analysis

I. Performed detailed manual analysis and building to find out and confirm the root causes of the build failures.

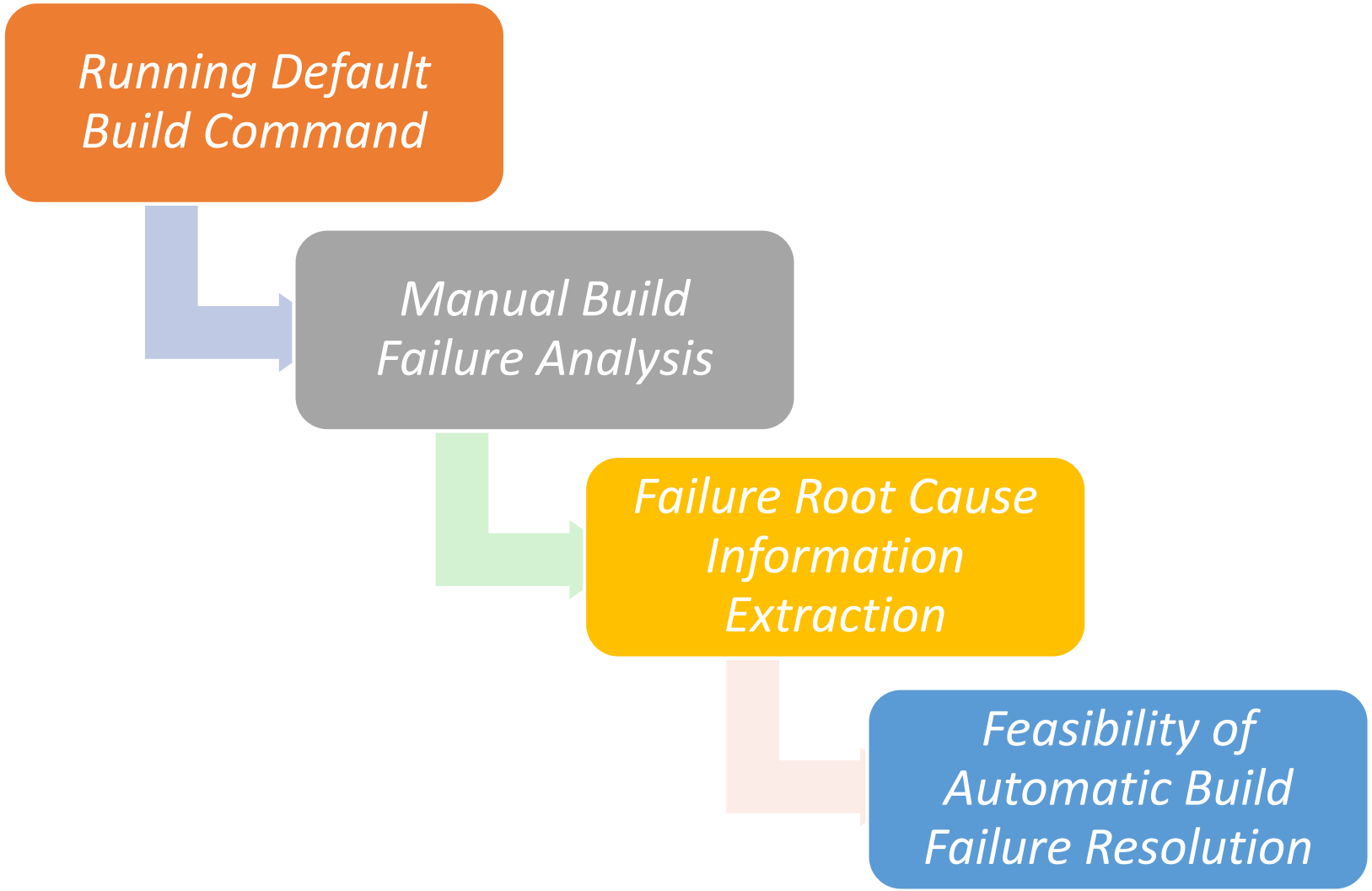
II. Build Failure Hierarchy based on manual analysis.

Build Failure Fix Analysis

I. Manual analysis on how build failures fix information can be extracted.

II. Feasibility of Automatic Building.

Overview of Our Study



Study Design

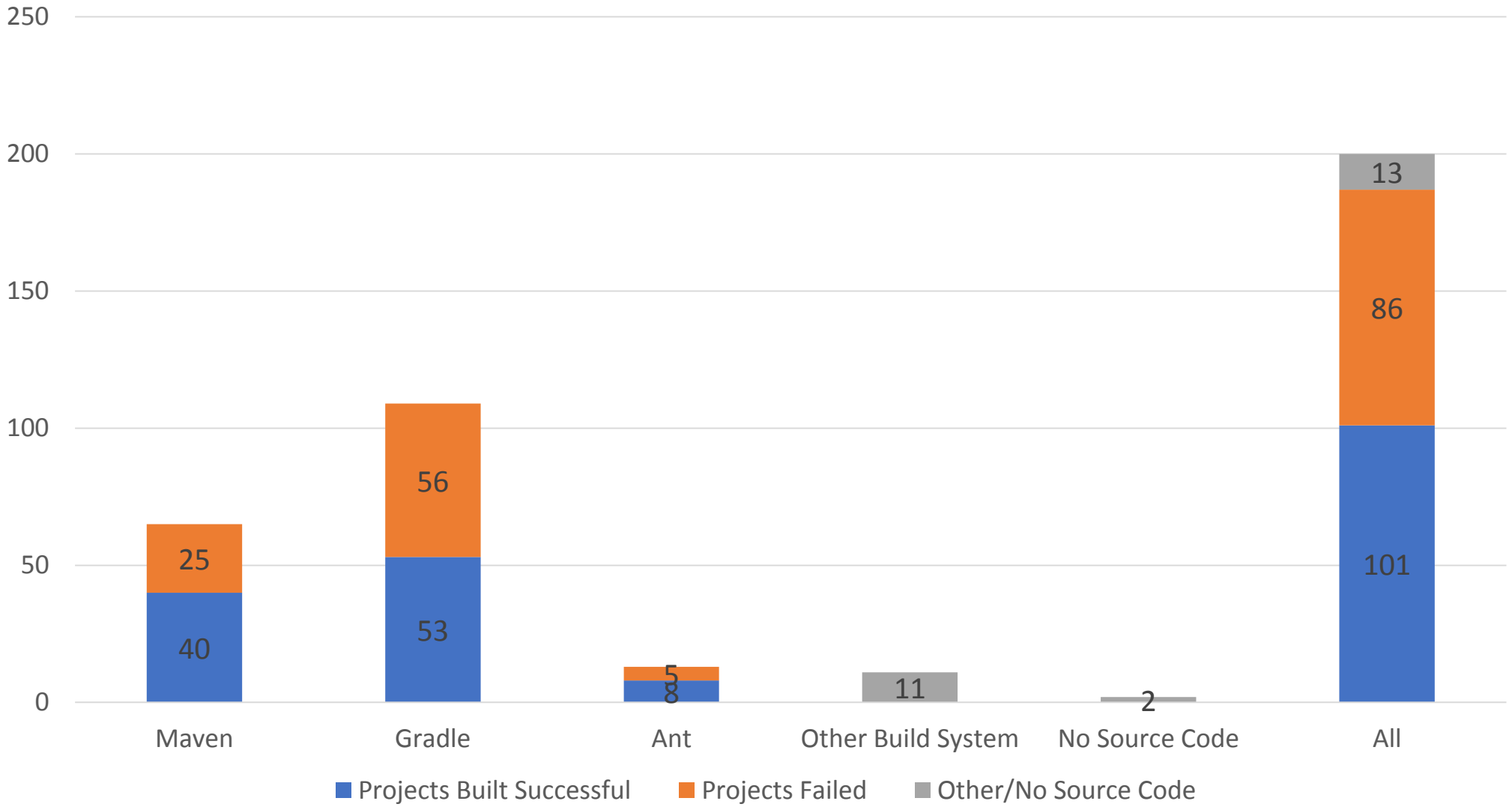
RQ1: What proportion of top Java projects can be successfully built with default build commands of popular build tools?

RQ2: What are the major root causes of the observed build failures?

RQ3: How easily can root causes of build failures be identified from readme files and build failure logs?

RQ4: What proportion of build failures can be (or have the potential to be) automatically resolved?

RQ1: Build Status With Default Build Command



RQ2: Major root causes of the build failures?

We classify the build failures to 3 general categories: environment issues, process issues, and project issues.

- Environment Issues

  Environment issues are build failures caused by the change of building environment.

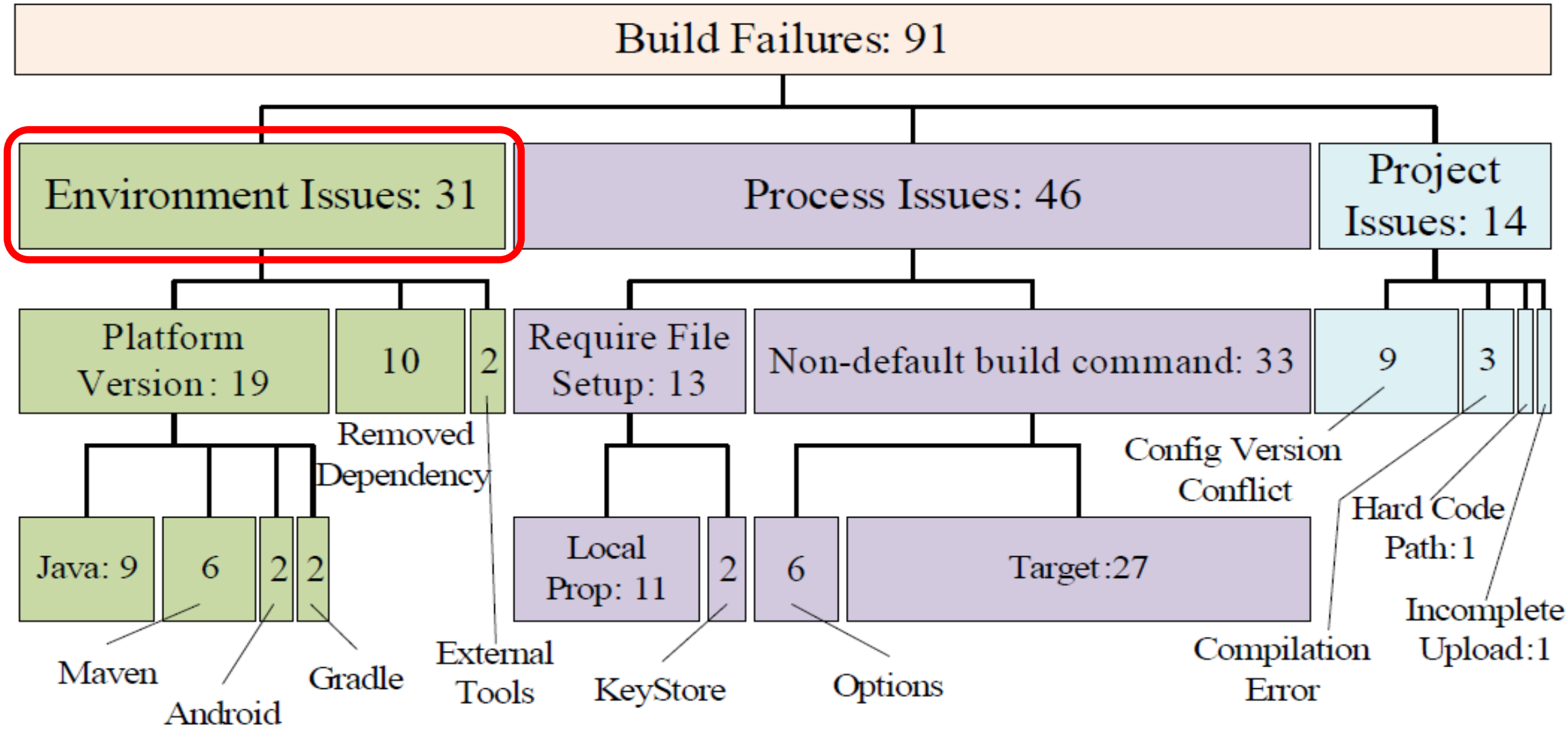
- Process Issues

  Process Issues are build failures caused by the requirement of additional steps in the building process.

- Project Issues

  Project issues are build failures caused by defects in the project itself.

Build Failure Hierarchy



Environment Issues Build Failures

Platform Version Issue

(elasticSearch/elasticSearch:42a7a55)

A problem occurred evaluating root project ’buildSrc’.
> Gradle 2.13 is required to build elasticsearch

External Tools Issue

(gocd/gocd: a3f77f9)

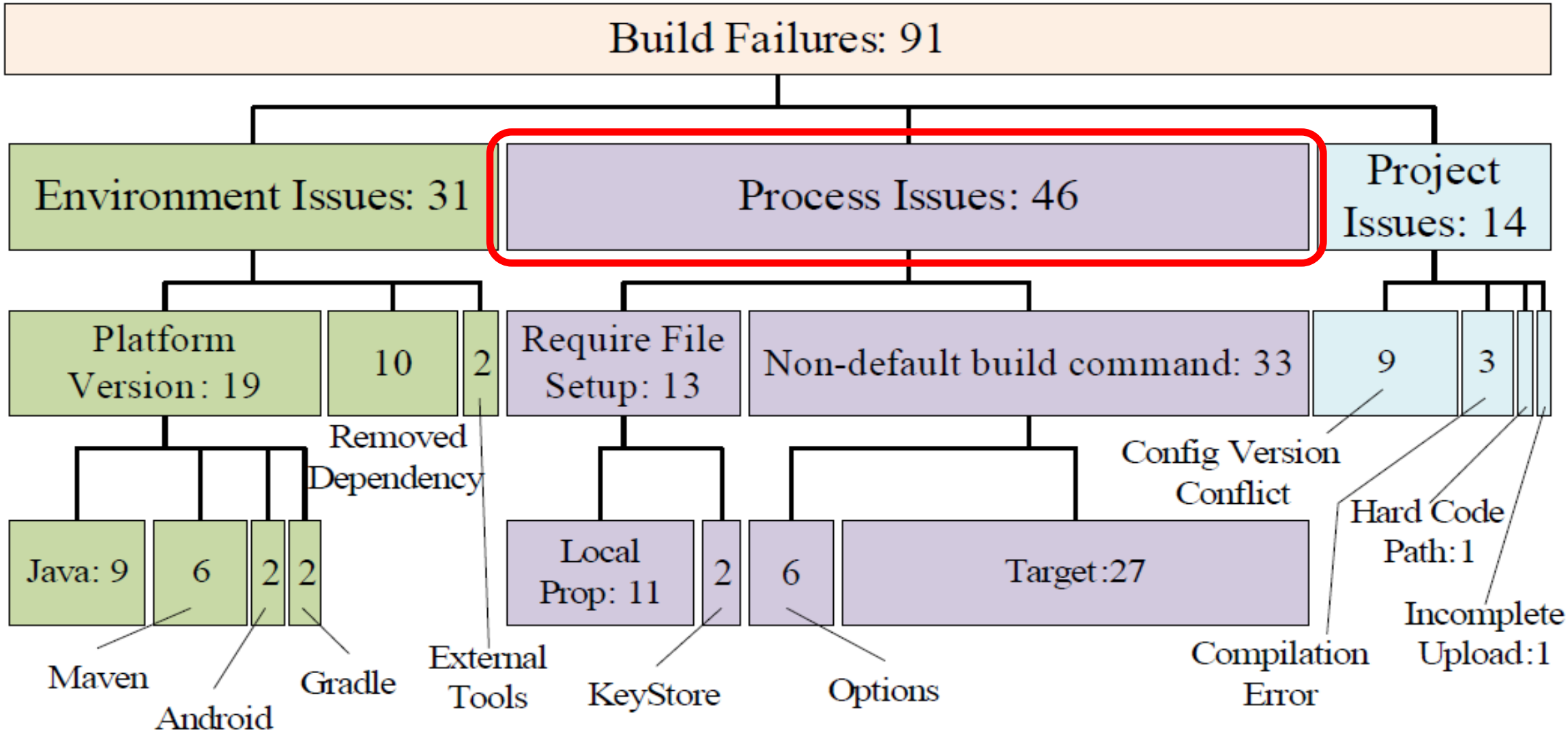
Execution failed for task ’:installers: agentPackageDeb’.
> A problem occurred starting process ’command ’fpm’’

Removed Dependency Issue

(Yalantis/Phoenix:188f2ec)

> Could not find com.android.tools.build: gradle:2.0.0-alpha1.
Searched in the following locations:
https://repo1.maven.org/maven2/com/android/tools/build/gradle/2.0.0-alpha1/gradle-2.0.0-alpha1.pom

Build Failure Hierarchy



Process Issues

Non-default Build Command. Expecting mvn clean install -P 'guice'

(roboguice/roboguice:d96250c)

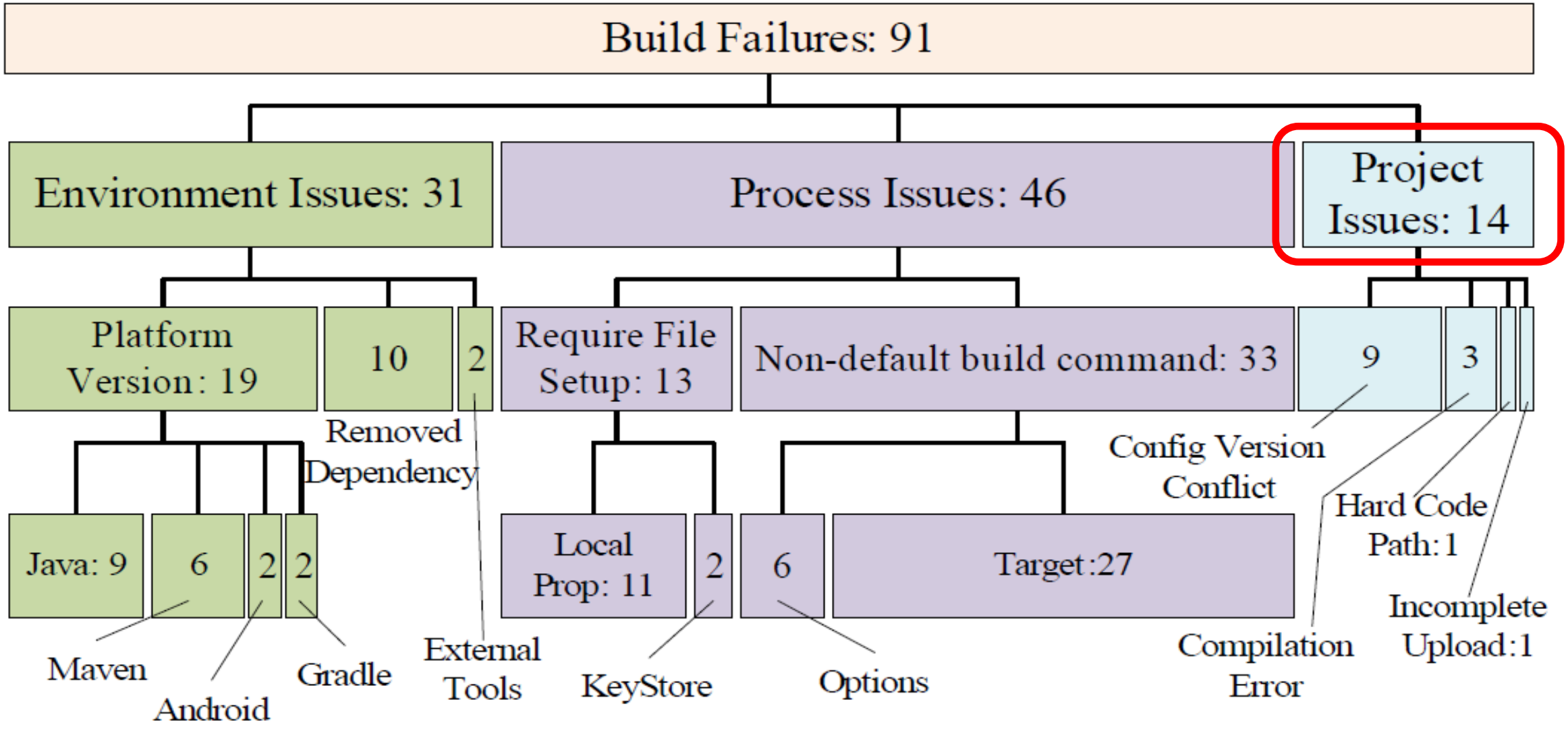
Exception in thread "pool-1-thread-1" java.lang.NoClassDefFoundError:
org/eclipse/aether/spi/connector/Transfer$State
at org.eclipse.aether.connector.wagon.WagonRepositoryConnector$GetTask.run(WagonRepositoryConnector.java:608)

Require File Setup Issue

(google/iosched: 2531cbd)

A problem was found with the configuration of task ':android:packageDebug'.
> File '/home/~/google_iosched/android/debug.keystore' specified for property 'signingConfig.storeFile' does not exist.

Build Failure Hierarchy



Project Issues(1/2)

Version Conflicts in Configuration Files

(google/iosched: 2531cbd)

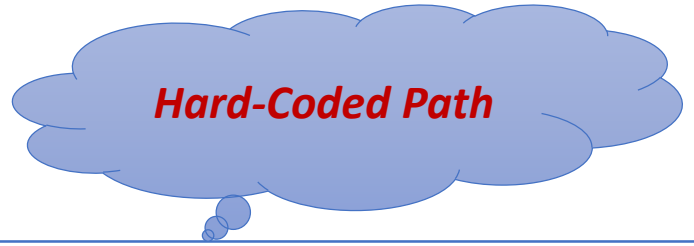
> Failed to apply plugin [id 'com.android.application']
> Gradle version 2.2 is required. Current version is 2.1. If using the gradle wrapper, try editing the distributionUrl in ~/gradle/wrapper/gradle-wrapper.properties to gradle-2.2-all.zip

Compilation Failure

(daimajia/AndroidSwipe-Layout: d7a5759)

> Compilation failed; see the compiler error output for details.
.../library/src/main/java/com/daimajia/swipe/SwipeLayout.java: 1327: error: illegal start of expression float willOpenPercent = (isCloseBeforeDragged ? ...

Project Issues(2/2)



(singwhatiwanna/dynamicload-apk: d262449)

A problem occurred configuring project ':doicommon'.
> The SDK directory '/home/~/153-singwhatiwanna_dynamic-load-apk/DynamicLoadApk/D:\adt-bundle-windowsx86_64-20130219\sdk' does not exist.

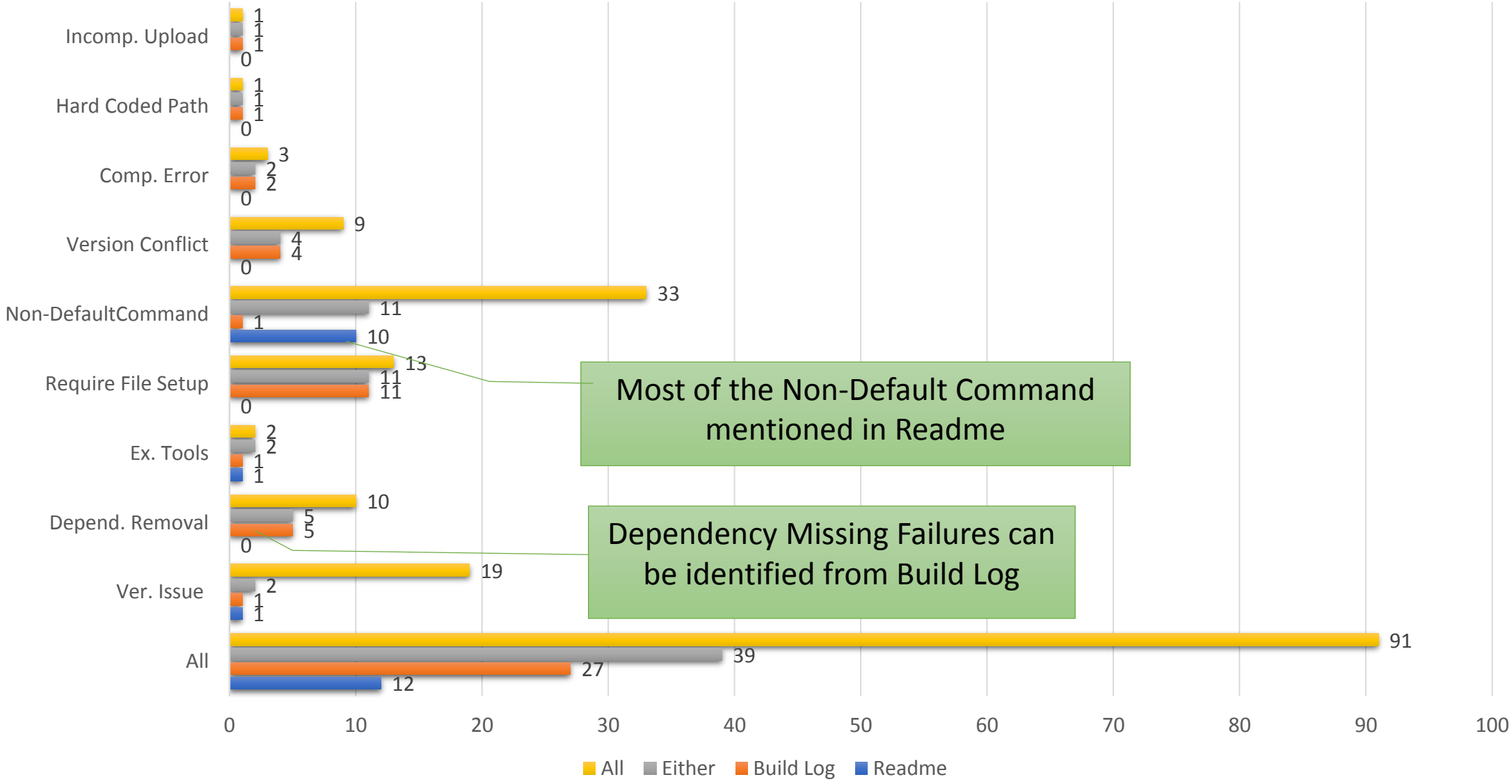
(android/platform_frameworks_base: e011bf8)

> com.android.ide.common.process.ProcessException: org.gradle.process.internal.ExecException: Process 'command '/home/~/android-sdk-linux/build-tools/21.1.2/aapt'' finished with non-zero exit value 1

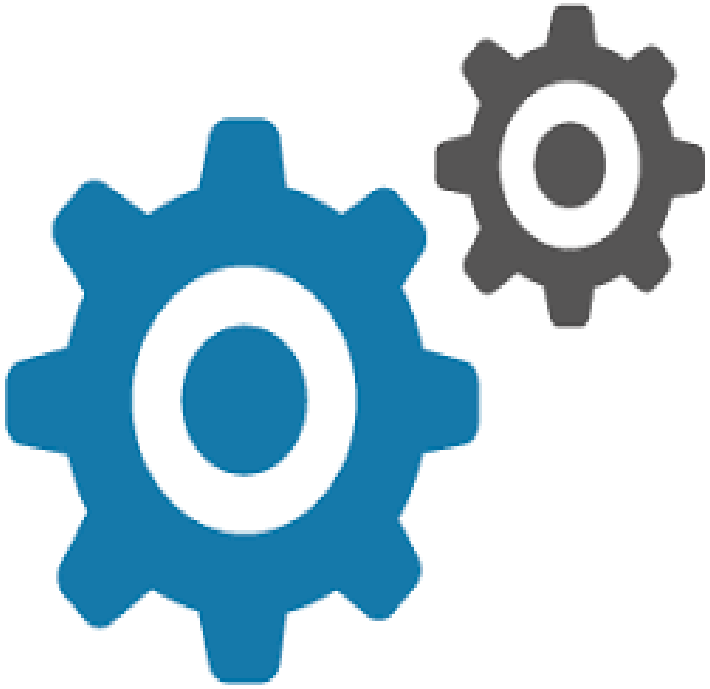
RQ3: Identifying Build Failure Cause

- ***Readme File***
- ***Build Log***

Build Failure Root Cause Revealed Distribution



RQ4: Automatic Resolution of Build Failures



- Build Command Extraction and Prediction
- Version Reverting
- Dummy File Generation

Build Command Extraction

Build commands in readme files / Wiki pages can be viewed as a type of entities and NLP Named Entity Recognition (NER) is a well-known task to identify a specific type of entities.

#Building
To build this project, first time you try to build you need to run this (requires Apache Ant 1.8 or higher and JDK 1.6):

ant -f updat_ dependencies.xml

which will setup the dependencies on intellij-core: is a part of command line compiler and contains only necessary APIs.
idea-full: is a full blown IntelliJ IDEA Community Edition to be used in former plugin module. Then, you need to run

ant -f build.xml

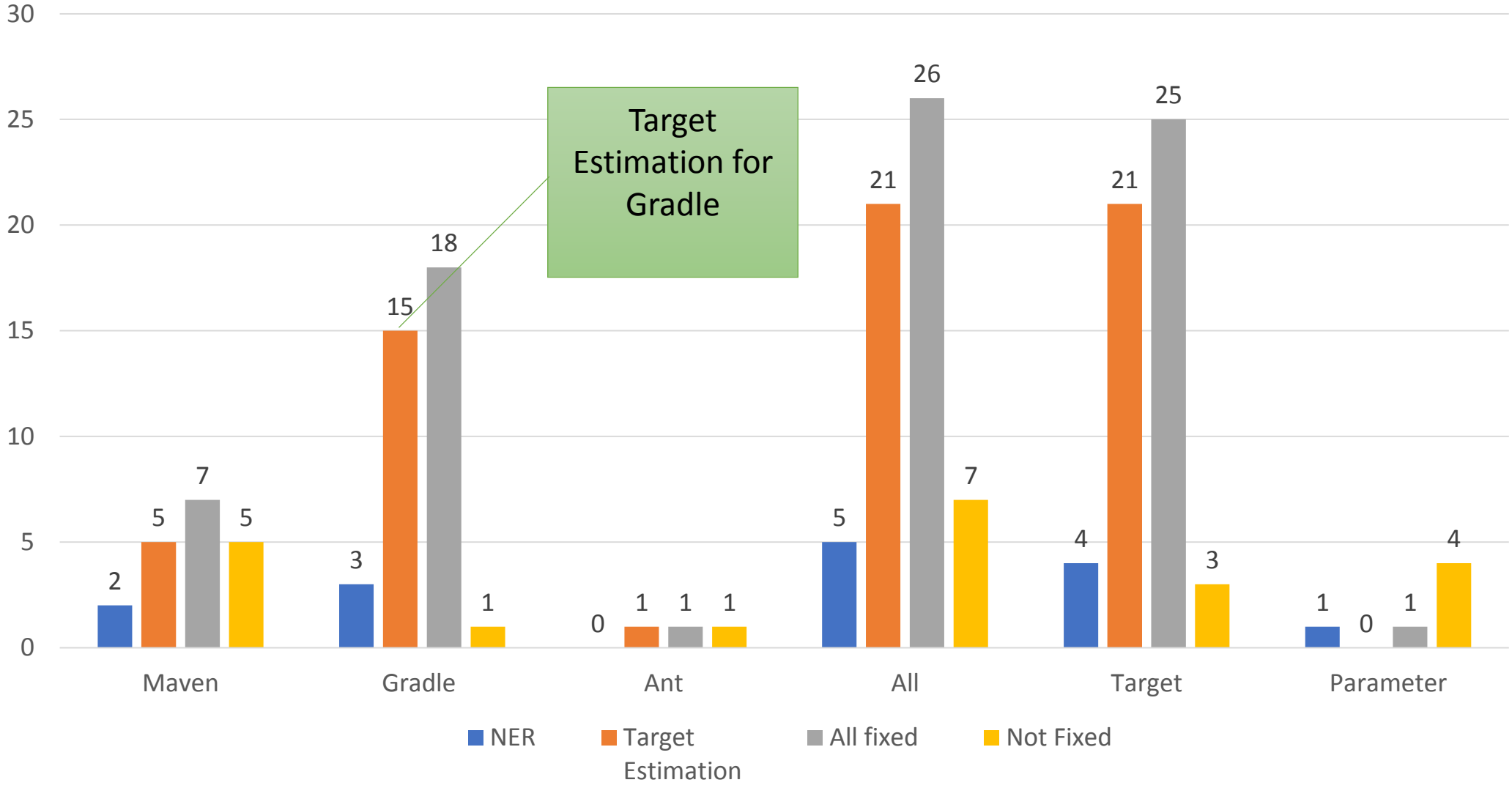
Build Command Prediction

- ***Readme Files are not there!!!***

- We can find which target is more like the correct building target by calculating the similarity between the target name and all the extracted commands in our training set of 857.

gradle tasks
Build tasks ---------- assemble - Assembles all variants of all applications and secondary packages. assembleAndroidTest - Assembles all the Test applications. assembleDebug - Assembles all Debug builds. assembleRelease - Assembles all Release builds. ….

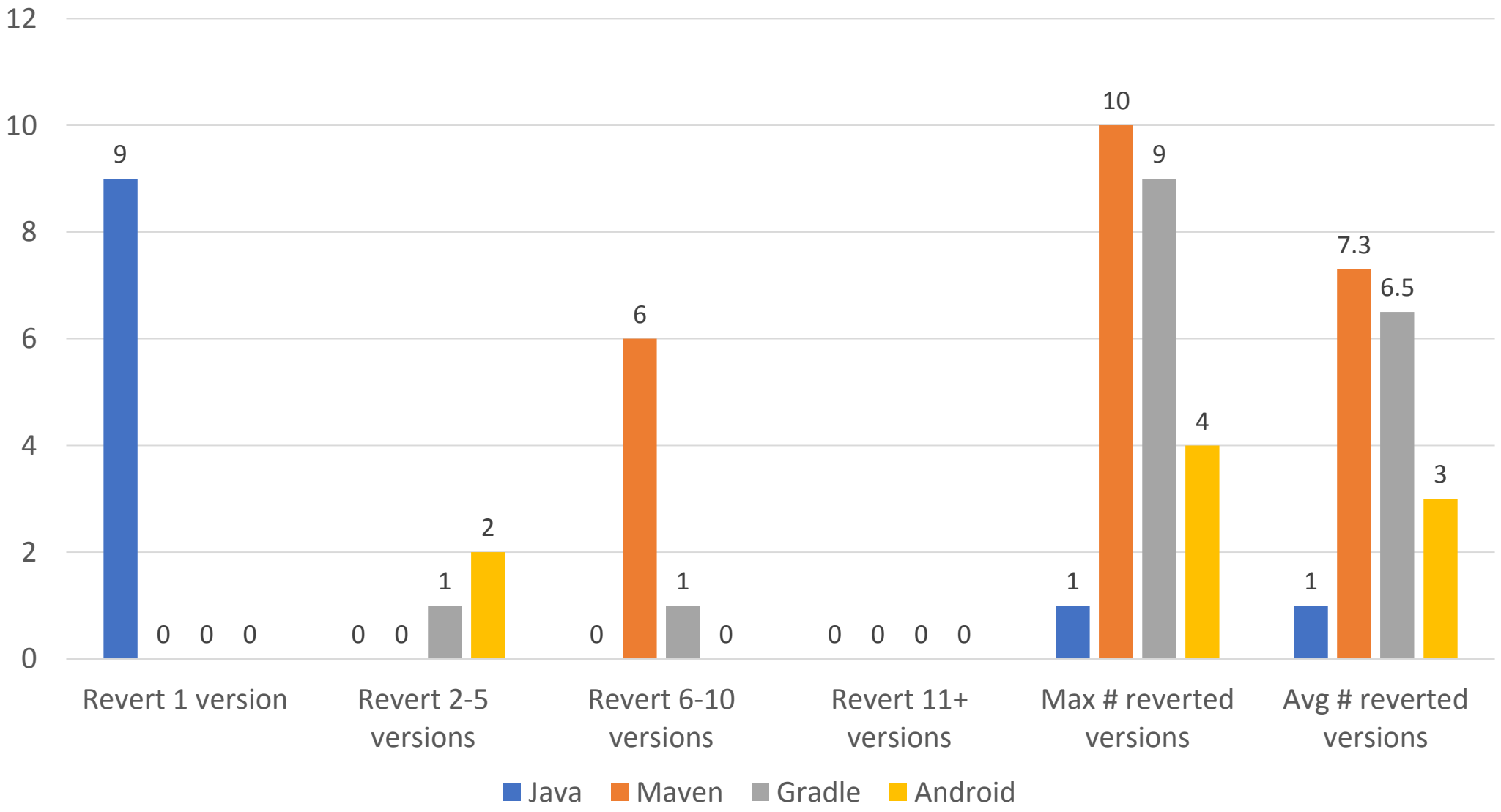
Resolved Build Failures By Extraction and Estimation



Version Reverting

- Many build failure happens due to incompatible SDK and build tools.
- Straightforward way to resolve SDK and build tools dependency is to revert the versions of SDK and build tools from the latest version.

Resolved Build Failures By Version Reverting



Dummy File Generation

- In many projects, a sample local file (e.g., local.property.example) is provided, and users can refer to it for what to be put into the local file.

- We find that, simply generating an empty local file will resolve **7 of the 13** require file setup build failures, and renaming the sample local file back will resolve **1 additional** build failures.

Build Failures Not Yet Analyzed

- Dependency Failure: Potential solution is to search for references to the Jar file in other projects' configuration files.

- Config Version Conflict failures: We need to perform in-depth analysis of config files and their dependencies.

Lesson Learned

- It is a necessity.
  - **Half of the top Java projects** cannot be straightforwardly built with default build commands.
- It is feasible.
  - Among the **86 projects** with build failures, **52 projects** can be built successfully with different approach such as build command extraction and estimation, version reverting etc.
- The challenges.
  - Our study has also identified several build failure categories whose automatic resolution can be difficult.


ThankYou
Merci
Salamat
Kop
Tack
Hvala
Gra
Xie
kun
Dank
Takk
Waad
Daw
Dakujem
krap
Dhanyavaadaalu
Dhanyavad
Khopjai
Kruthagnathalu
Arigatou
Dhonnobaad
Asante
Falemínderit
Grazie
Go
Dhanyavaad
al
or
Cam
Dankie
Spaas
Mul
Ačí
Shokriya
Ngiyabonga
Dziękuję
Shokrun
Todah
Mamnoon
Kasih
umesc
Diolch
Kiitos
Sheun
Shnorhakalutiun
Gamsahapnida
Te ekkür
Dekuju/Dekujeme
Gomapsupnida
Grazzi
raibh
Gracias
Nandree
Blagodariya
Fyrir
Terima
Enkosi
Danke
dank
Euxaristo
Kun
Shukriya
ederim
Hain
Dhan
daa