

圖 9/2 DevOps技能精進

# DEVOPS 2015

開發敏捷與維運高效的IT新典範

不只自動化而且更敏捷的Android開發工具  
Gradle

邱炫儒

CI engineer @ HTC

# 十五年前的持續整合



**“PROJECT CODE RUSH”**



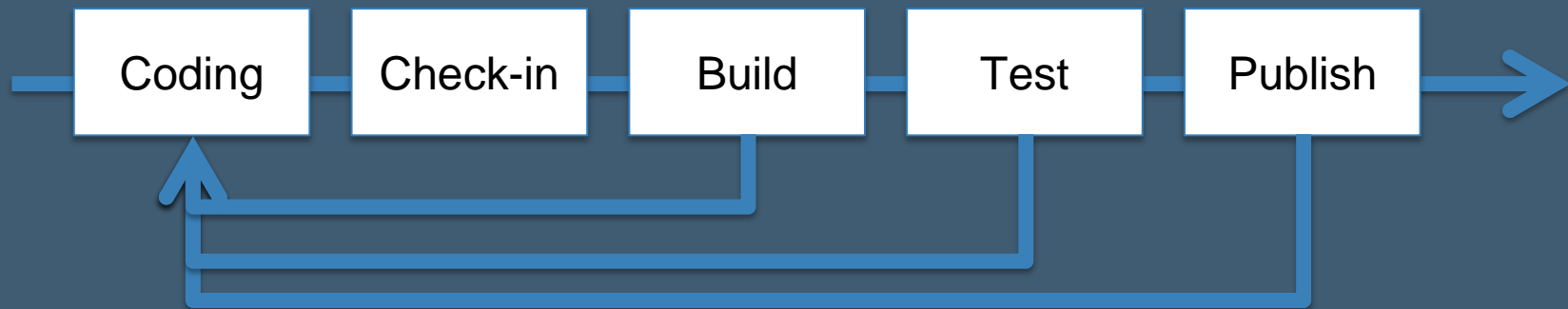
# How people do CI 15 years ago



# 解決問題的方法是...



# Automate everything and smoothly



可重複運行

可重製錯誤

減少人為/環境錯誤

流程透明

產生更有品質的程式碼

# Gradle - offering thoughtful conventions

From Command line to IDE to CI

Gradle build files are groovy scripts

Product Flavor

Powerful dependencies management

Convention 1

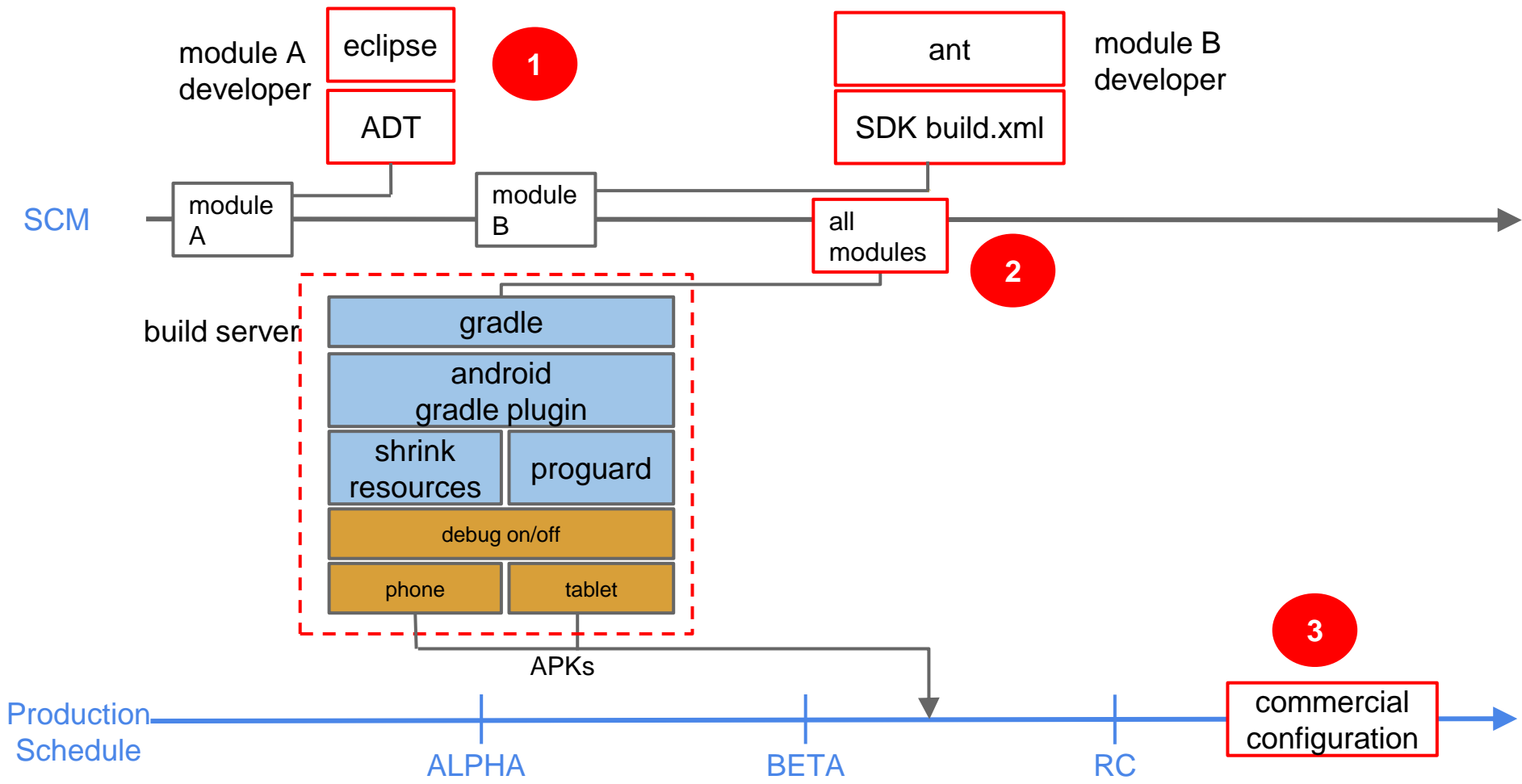
消除溝通的嫌隙

一句話惹毛**DevOps**：

我電腦沒這個問題呀！



# 溝通的嫌隙



# 減少溝通的嫌隙

## From **command line** to **IDE** to **continuous integration**

SCM

build server

module A  
developer

module B  
developer

all  
modules

gradle

android  
gradle plugin

shrink  
resources

proguard

debug on/off

phone

tablet

APKs

3

commercial  
configuration

Production  
Schedule

ALPHA

BETA

RC

# 後期出現的錯誤往往影響巨大

**591** 房屋交易  
.com.tw  
家，輕鬆自由找到

首頁 > 租屋 > 台中市 > 北區 > 獨立套房 > 10000-15000元 > 房屋詳細資訊 (R3401433)

套房出租 ★ 有乾屍分離 ★ 有浴缸 ★ 新豪宅 ★ 電梯套房



點擊查看大圖(共103張) >>



租金：14,000 元/月

(含管理費、清潔費、第四台、網路、水費、瓦斯費)

押金：二個月

坪數：14坪

樓層：3F/7F

用途：住家用

型態/類型：電梯大樓/獨立套房

車位：無

社區：新大樓

地址：台中市北區太原路二段

三分鐘學裝潢

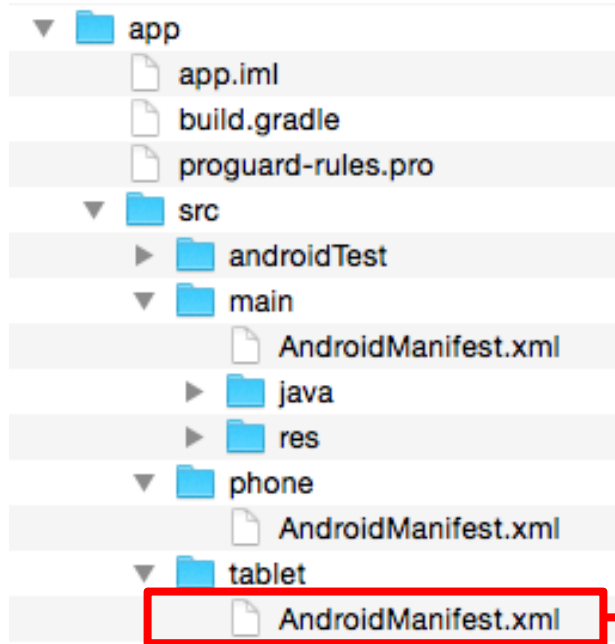
■ 裝潢攻略

將商業版本客製盡量提前到開發時期

## Convention 2

# **Product Flavor and Multiple APK**

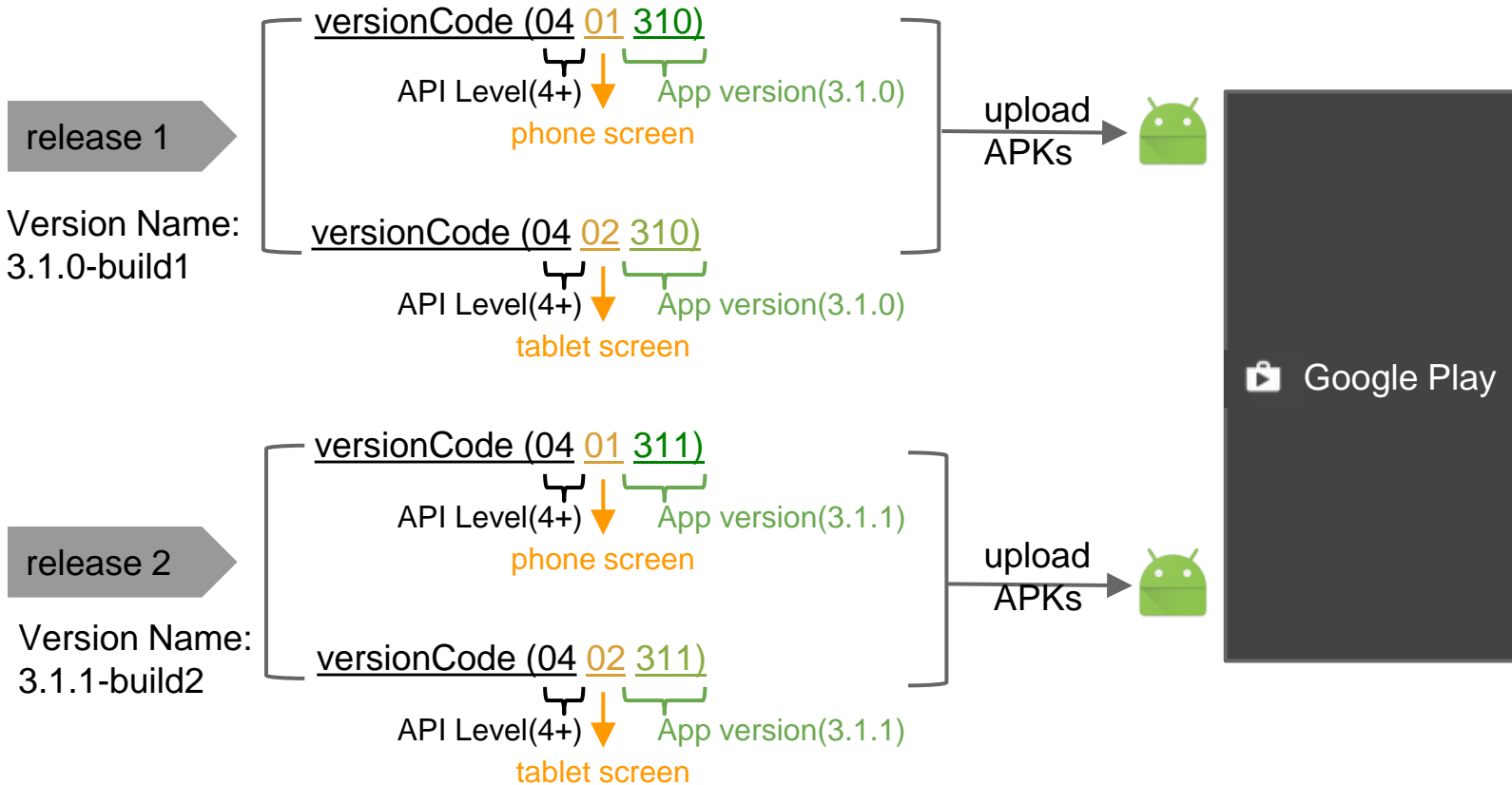
# Example on Google dev site: Design Multi-APK for tablet and phone



custom AndroidManifest.xml for tablet

```
<manifest ... >  
    <supports-screens android:smallScreens="false"  
        android:normalScreens="false"  
        android:largeScreens="true"  
        android:xlargeScreens="true"  
        android:requiresSmallestWidthDp="600"  
    />  
    ...  
</manifest>
```

# The versionName/versionCode schema for multiple APK



把處理邏輯都編寫至**gradle**設定檔中



# custom build logic and integrate with Jenkins

build.gradle:

```
android{
    defaultConfig {
        versionName computeVersionName()
    }
    productFlavors {
        phone{ versionCode computeVersionCode(1) }
        tablet { versionCode computeVersionCode(2) }
    }
}

def computeVersionCode(int flavor) {
    def version_code = ext.minSdkVersion * 100000 + flavor * 1000 + ext.versionMajor * 100 + ext.versionMinor*10 \
    + ext.versionIncremental
    return version_code
}

def computeVersionName(){
    def buildNumber = System.getenv("BUILD_NUMBER") ?: "dev"
    def version_name = ext.versionMajor+"."+ext.versionMinor+"."+ext.versionIncremental+"-"+buildNumber
    return version_name
}
```

為什麼要把 **Jenkins buildNumber** 寫進  
**version name**?

為什麼要把 **Jenkins buildNumber** 寫進  
**version name**?

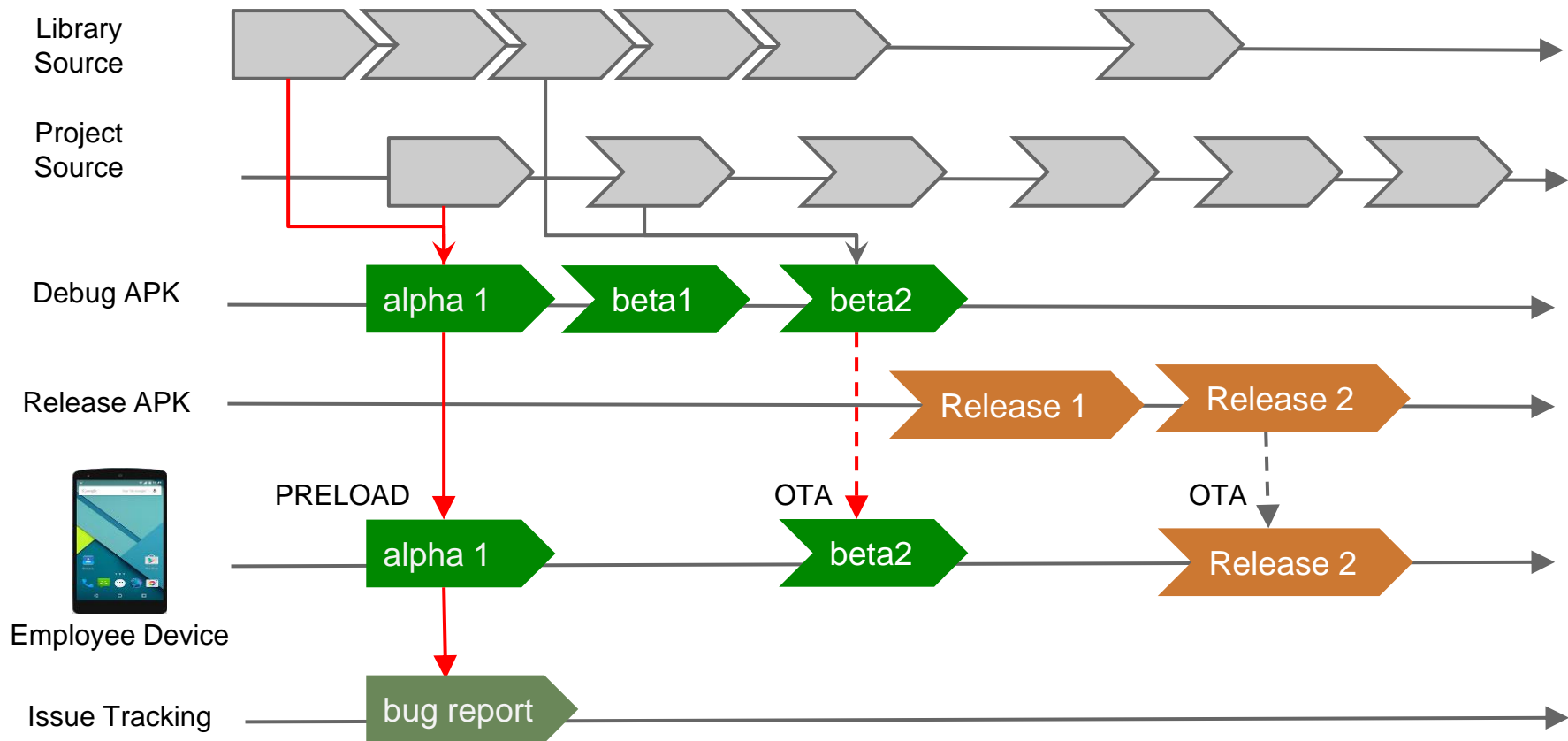
混在一起做撒尿牛丸？

為什麼要把 **Jenkins buildNumber** 寫進  
**version name**?

混在一起做 **Dogfooding**

# What is Dogfooding?

# Dogfooding 等於良好的版本管控策略



# 透過Jenkins追朔程式碼



versionName  
3.1.0-Build#2



## Build #2 (2013/4/3 下午 05:55:57)

**Changes**

- add log when loading readme url setting ([detail](#) / [githubweb](#))
- fix build break ([detail](#) / [githubweb](#))

**Started by an SCM change**

**Revision:** 21053a7d0a86a1fcaa6765f22b524ef96d37a4fd

- origin/master
- origin/HEAD

**add log when loading readme url setting**

master

iamsamchiu authored 6 days ago 1 parent 5000fe1 commit 2da58dc4f9ec

Showing 1 changed file with 2 additions and 0 deletions.

```
src/main/java/org/jenkinsci/plugins/readme/ReadMeRootAction.java
... @@ -71,6 +71,8 @@ public ReadMeDescriptor getDescriptor() {
71 71     public ReadMeDescriptor() {
72 72         // 在class的建構子呼叫load()可以從硬碟讀取之前透過save()的資料
73 73         load();
74 +         logger.info("load readme url:"+readMeURL);
75 +
76 76     }
77
78     @Override
```

程式碼和腳本**Script** 都寫好了  
哪些東西要放上**SCM**？



## Convention 3

有條不紊的程式碼管理

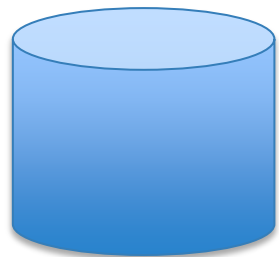
“By distributing your project source based on gradle ,anyone can work with it **without needing to install many annoying tools/dependencies beforehand**”

“Users of the build are guaranteed to use the **same process and the same dependencies version** that was designed to work with”

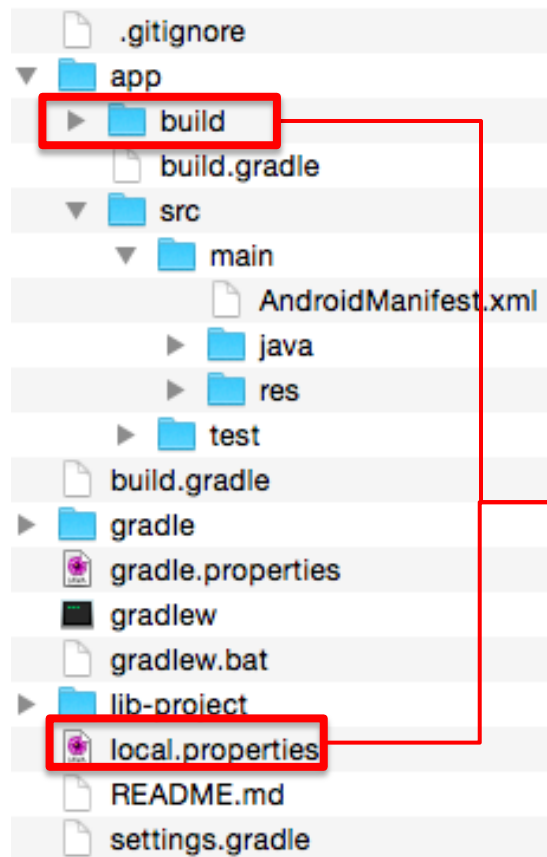
from <http://gradle.org/>

答案很明顯了  
全部都放上**SCM**?

# 建立可重複且可靠的流程

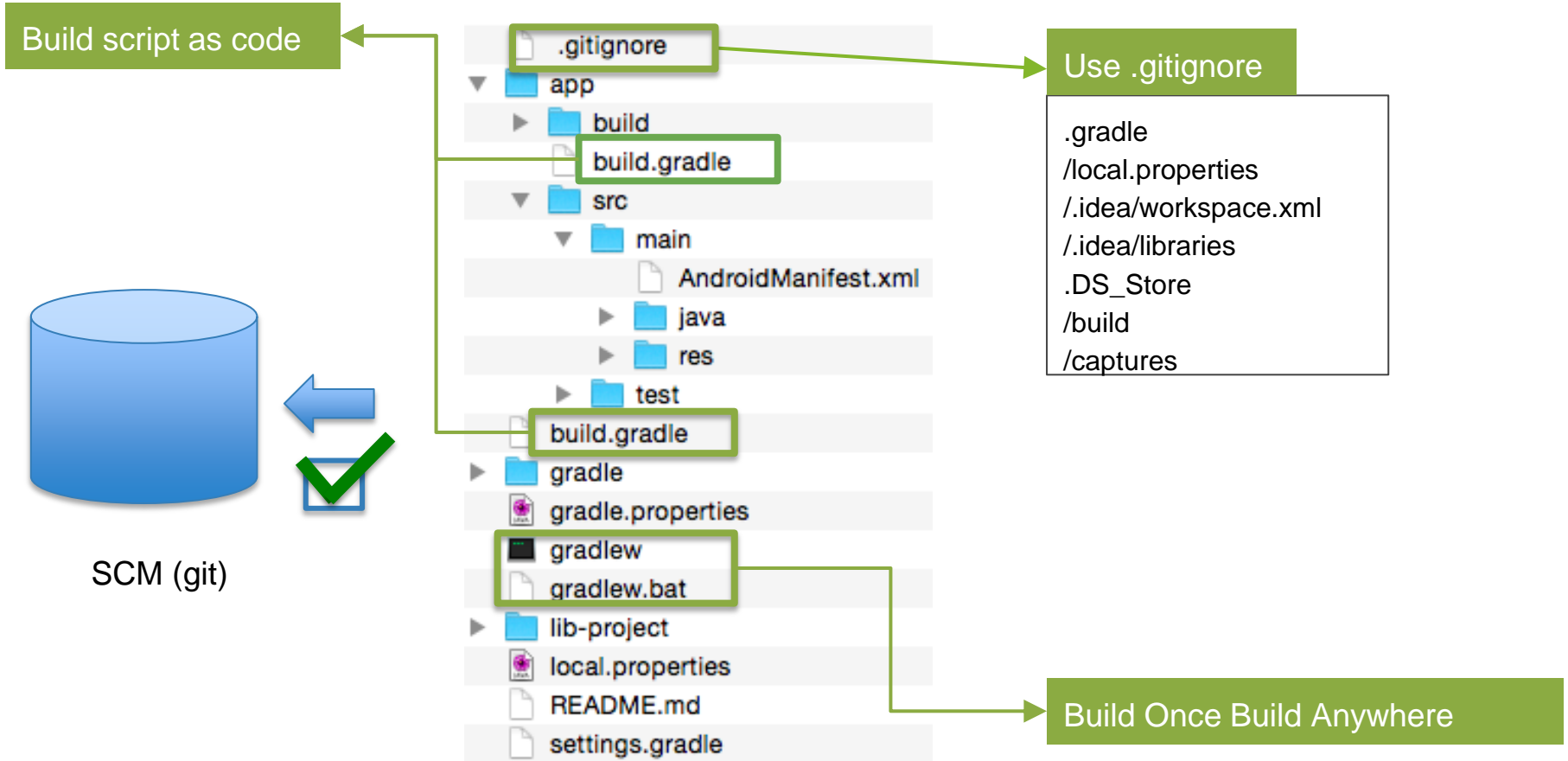


SCM (git)



Keep your workspace clean

# 建立可重複且可靠的流程



# gradle wrapper

```
#Sun Aug 09 19:57:07 CST 2015
```

```
distributionBase=GRADLE_USER_HOME
```

```
distributionPath=wrapper/dists
```

```
zipStoreBase=GRADLE_USER_HOME
```

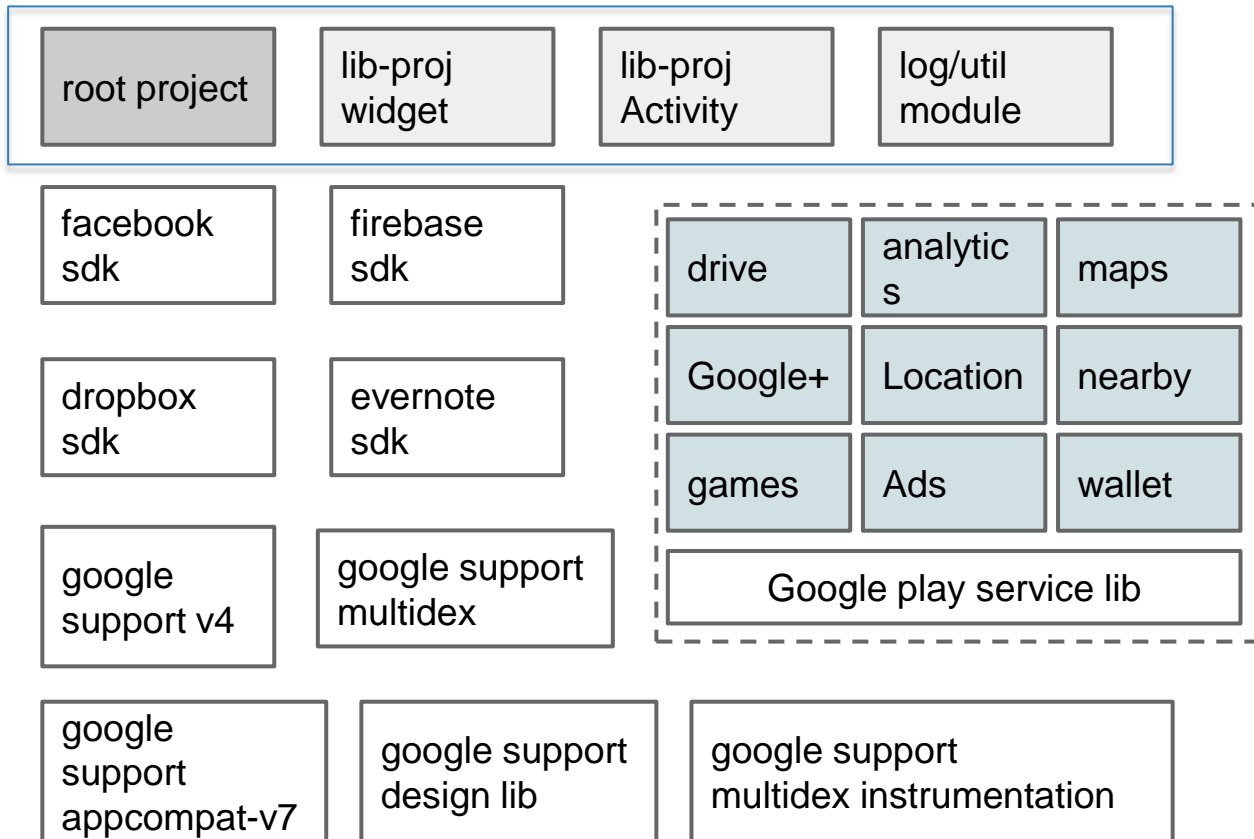
```
zipStorePath=wrapper/dists
```

```
distributionUrl=https\://services.gradle.org/distributions/gradle-2.5-all.zip
```

Convention 4

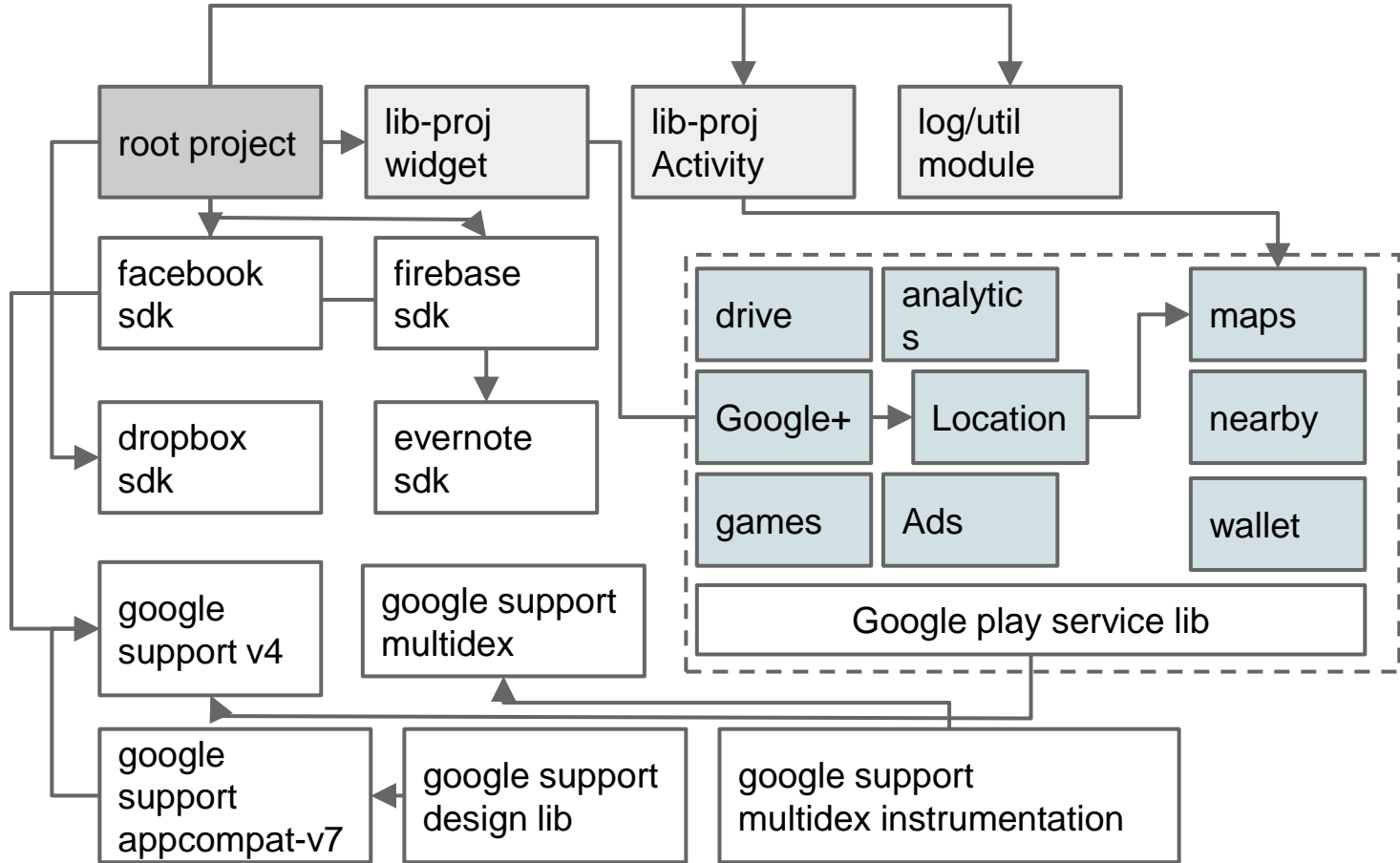
# **Dependencies Management**

# A large scale Android App

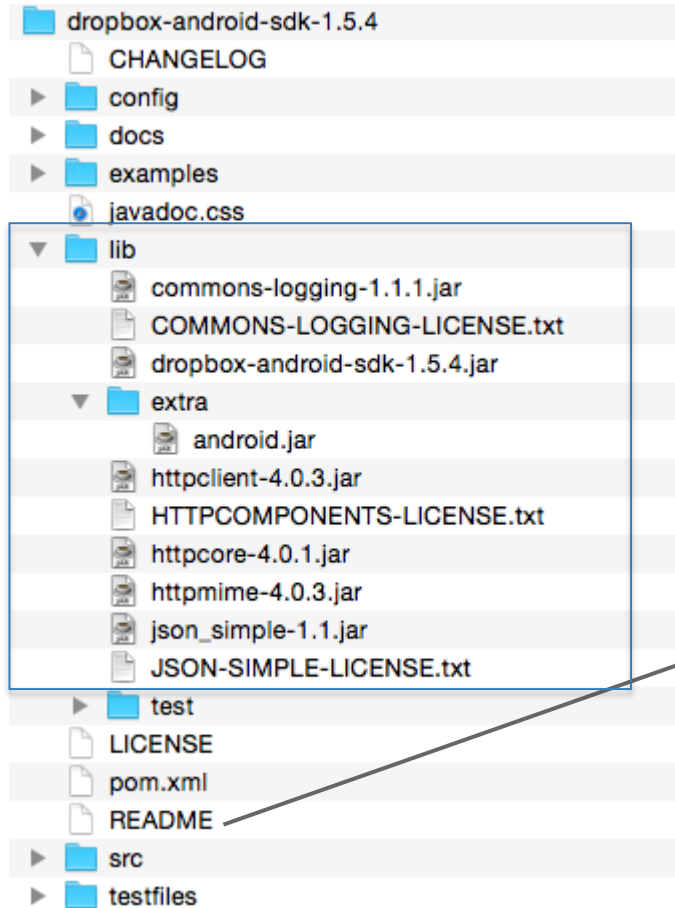




# The dependency nightmare



# Before Maven



1

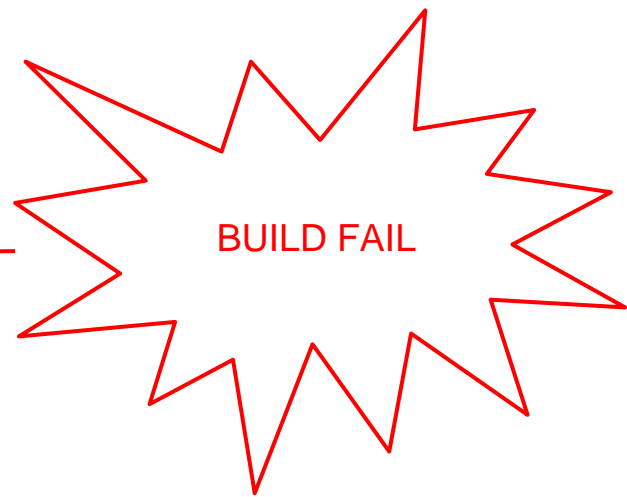
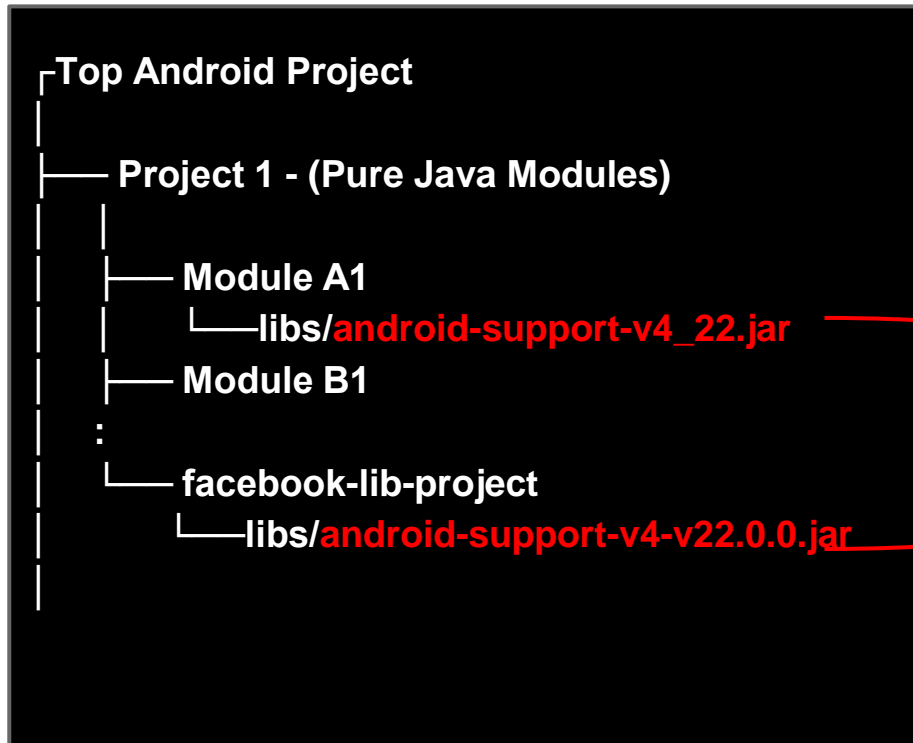
Download dropbox-android-sdk from dropbox website

2

Getting started with the Dropbox Android SDK:

1. Include everything under lib/ in your project/build.
2. You'll want to start off by creating an AndroidAuthSession with your consumer key and secret.
- 3...

# 到底用了哪一版library?



# After Maven

1. Use gradle

2.

```
dependencies {  
    compile 'com.facebook.android:facebook-android-sdk:4.1.0'  
}
```

```
dependencies {  
    compile 'com.google.android.gms:play-services-wearable:7.3.0'  
}
```

3. That's it.

# How about the depend on a tree of dependencies?

```
dependencies {  
  compile 'com.facebook.android:facebook-android-sdk:4.1.0'  
}
```

→ → android-support-v4:21.0.0

```
dependencies {  
  compile 'com.google.android.gms:play-services-wearable:7.3.0'  
}
```

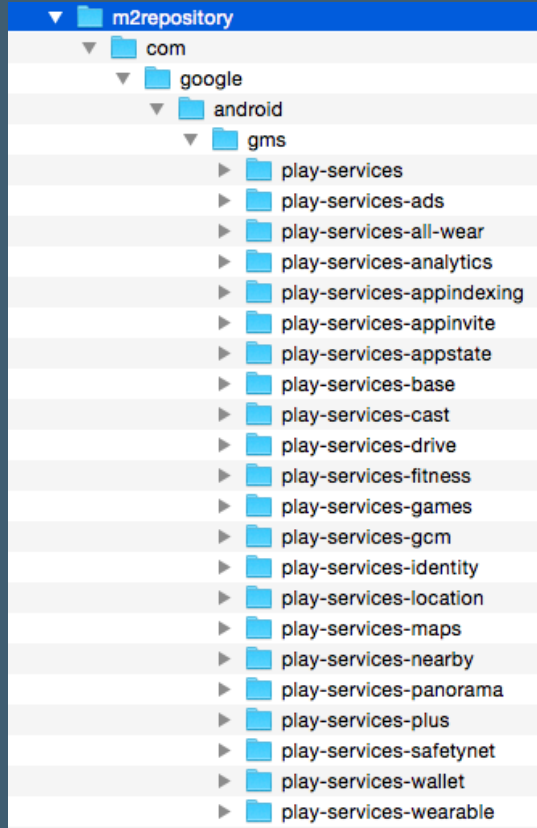
→ → android-support-v4:22.0.0

Gradle could be even better

## Transitive Dependencies on Gradle(可遞移性相依)

A -> B and B -> C  
hence A -> C

# google play service aar library hierarchy on maven



```
com.google.android.gms
play-services-ads          -> play-services-base
                           -> play-services-analytics
play-services-all-wear   -> play-services-ads
                           -> play-services-analytics
                           -> play-services-base
                           -> play-services-fitness
                           -> play-services-gcm
                           -> play-services-location   -> play-services-maps
                           -> play-services-wearable
play-services-analytics   -> play-services-base
play-services-appindexing -> play-services-base
play-services-appinvite   -> play-services-base
play-services-appstate    -> play-services-base
play-services-base        -> com.android.support:support-v4:22.0.0   ->support-annotations:22.0.0
play-services-cast        -> play-services-base
play-services-drive       -> play-services-base
play-services-fitness     -> play-services-base
                           -> play-services-location   -> play-services-maps
play-services-games       -> play-services-base
                           -> play-services-base
play-services-gcm         -> play-services-base
play-services-identity    -> play-services-base
play-services-location    -> play-services-base
                           -> play-services-maps
play-services-maps        -> play-services-base
play-services-nearby      -> play-services-base
play-services-panorama    -> play-services-base
play-services-plus        -> play-services-base
play-services-safetynet   -> play-services-base
play-services-wallet      -> play-services-base
                           -> play-services-identity
play-services-wearable    -> play-services-maps
                           -> play-services-base
```



# Tips of transitive dependencies

```
dependencies {  
  compile 'com.facebook.android:facebook-android-sdk:4.1.0'  
}
```



android-support-v4:21.0.0

```
dependencies {  
  compile 'com.google.android.gms:play-services-wearable@aar:7.3.0'  
}
```



android-support-v4:22.0.0

# As a library provider

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.google.android.gms</groupId>
  <artifactId>play-services-location</artifactId>
  <version>7.8.0</version>
  <packaging>aar</packaging>
  <dependencies>
    <dependency>
      <groupId>com.google.android.gms</groupId>
      <artifactId>play-services-maps</artifactId>
      <version>7.8.0</version>
      <scope>compile</scope>
      <type>aar</type>
    </dependency>
```

.....



Maven Repository

remember to provide your transitive dependencies in pom.xml

# Handle dependency conflict

Gradle offers the following conflict resolution strategies:

*Newest*: The newest version of the dependency is used. This is Gradle's **default** strategy, and is often an appropriate choice as long as **versions are backwards-compatible**.

*Fail*: **A version conflict results in a build failure**. This strategy requires all version conflicts to be resolved explicitly in the build script.

# Use 'Fail' resolution strategy

**build.gradle:**

```
configurations.all {
    resolutionStrategy {
        // fail eagerly on version conflict (includes transitive dependencies)
        // e.g. multiple different versions of the same dependency (group and name are equal)
        failOnVersionConflict()
    }
}
```

# Handle dependency conflict

## build.gradle:

```
dependencies {
    compile('org.hibernate:hibernate:3.1') {
        //in case of versions conflict '3.1' version of hibernate wins:
        force = true
        //disabling all transitive dependencies of this dependency
        transitive = false
    }
}
```

# dependency substitution

## 51.8.3.1. Substituting an external module dependency with a project dependency

One use case for dependency substitution is to **use a locally developed version of a module in place of one that is downloaded from an external repository**. This could be useful for testing a local, patched version of a dependency.

### build.gradle:

```
configurations.all {
    resolutionStrategy.dependencySubstitution {
        substitute module("org.utils:api") with project(":api")
        substitute module("org.utils:util:2.5") with project(":util")
    }
}
```

# Versioning Control



建置系統始終應該指定專案所需外部類別庫的確切版本:

```
dependencies {  
    compile 'com.facebook.android:facebook-android-sdk:4.1.0'  
}
```

開發時期指定專案所需外部函式庫的最新版本:

```
dependencies {  
    compile 'com.facebook.android:facebook-android-sdk:4.1.+'  
}
```

開發時期指定專案所需外部函式庫的SNAPSHOT版本:

```
dependencies {  
    compile 'com.facebook.android:facebook-android-sdk:4.1.0-SNAPSHOT'  
}
```

# 若違背架構原則就讓建置失敗(optional)

```
value=$( gradle dependencies | grep -ic "SNAPSHOT" )  
if [ $value -eq 0 ]  
then  
    echo "Didn't found SNAPSHOT usage"  
else  
    echo "Found SNAPSHOT usage"  
    exit 1  
fi
```

```
Internal use, do not manually configure ##  
support:appcompat-v7:22.2.1  
support:support-v4:22.2.1  
android.support:support-annotations:22.2.1  
com.google.android.gms:play-services:7.5.0  
com.google.android.gms:play-services-ads:7.5.0
```

```
+--- com.google.android.gms:play-services-base:7.5.0  
|    \--- com.android.support:support-v4:22.0.0 -> 22.2.1 (*)  
|    \--- com.google.android.gms:play-services-analytics:7.5.0  
|         \--- com.google.android.gms:play-services-base:7.5.0 (*)  
+--- com.google.android.gms:play-services-analytics:7.5.0 (*)  
+--- com.google.android.gms:play-services-appindexing:7.5.0  
|    \--- com.google.android.gms:play-services-base:7.5.0 (*)  
+--- com.google.android.gms:play-services-appinvite:7.5.0  
|    \--- com.google.android.gms:play-services-base:7.5.0 (*)  
+--- com.google.android.gms:play-services-appstate:7.5.0  
|    \--- com.google.android.gms:play-services-base:7.5.0 (*)
```



## Reference

- Android plugin for gradle:

<https://developer.android.com/tools/building/plugin-for-gradle.html>:

- Android tools project site, tips:

<http://tools.android.com/tech-docs/new-build-system/tips>

- Gradle dependency management:

[https://docs.gradle.org/current/userguide/dependency\\_management.html](https://docs.gradle.org/current/userguide/dependency_management.html)

- Google dev site, multiple apk:

<https://developer.android.com/google/play/publishing/multiple-apks.html>

- Project Code Rush:

<https://archive.org/details/CodeRush>

- Sample project on github:

<https://github.com/iamsamchiu/AndroidSampleForGradleUsage>

Q & A