Automate everything

Build

Quality

Delivery
Deliver fixes faster

First actual case of bug

Relay #70
(moth) in re

Andersen started.
closed down.
The goal

$ git commit
$ git tag beta
$ git push
Update fatigue
Update fatigue

My apps & games

INSTALLED  ALL  BETA

WhatsApp Messenger
WhatsApp Inc.
4.4 ★
UPDATE

Plaid
Nick Butcher
INSTALLED

yotube job search Switzerland
yoture AG
4.1 ★
INSTALLED

Rainy Days
Optimization for recent build system
Describe pipeline aware alternatives
Merge pull request #2438 from
jenkins-test-harness 2.13
Merge branch 'master' into Log
Adapted to changes in Log
@oleg-nenashev created a Base
Use LoggerRule.
Update changelog.html
Noting #2402
[JENKINS-35570] List of comments
Merge pull request #2420 from
Remove only @Option annotation
[JENKINS-35423] CLI comments
Noting JENKINS-36494
Merge pull request #2437 from
[JENKINS-36494] Clarify that
[FIXED JENKINS-36494] Switch
[FIXYB JENKINS-26438] Allow
updated changelog for release
[maven-release-plugin] prepare
jenkins-2.13 [maven-release
[FIXED JENKINS-34668] - set
[JENKINS-27530] Noting #2443
Merge pull request #2439 from
[FIXED JENKINS-27530] Jenkins
Merge pull request #2440 from
Another unnecessary call to do
Remove the concept of yanking
Lowercase i for consistency
That fix was never in a release.
Annotate DescriptorList method
updated changelog for release
app/build.gradle

android {

    defaultConfig {

        // User-visible display name; update for every release
        versionName '1.4'

        // Needs to be incremented for every release
        versionCode 4

        ...
    }

    ...
}

...
App traceability

```java
ext.versionMajor = 1; ext.versionMinor = 4; ext.versionPatch = 0;
ext.jenkinsBuildNumber = Integer.valueOf(System.env.BUILD_NUMBER ?: 0)

android {
  defaultConfig {
    versionName computeVersionName()
    versionCode computeVersionCode()
  }
}

// Returns name based on version values, e.g. ‘1.4.0’
def computeVersionName() {
  return "${versionMajor}.${versionMinor}.${versionPatch}"
}

// Returns auto-incrementing value, e.g. 140017, for Jenkins build #17
def computeVersionCode() {
  return (versionMajor * 100_000) + (versionMinor * 10_000) +
          (versionPatch * 1_000) + jenkinsBuildNumber
}
```
App traceability

android {
    signingConfigs {
        debug {
            // Override the local debug keystore, so that APKs built by
            // any developer, or by Jenkins can be installed on any device
            storeFile file('../debug.keystore')
        }
    }

    buildTypes {
        internal {
            // For internal builds, use an app ID separate from release builds
            applicationIdSuffix '.dogfood

            // Include the Jenkins build number in the displayed version name
            versionNameSuffix " (build ${jenkinsBuildNumber})"

            // Sign with the common debug key
            signingConfig signingConfigs.debug
        }
    }
}
Prerequisites: general

Prerequisites

- Source code
- JDK
- Gradle

Jenkins

- Git plugin
  (SVN, Mercurial, Perforce, ...)
- Automated install
- Automated install
  (Gradle wrapper, Jenkins plugin)
Prerequisites: Android

Prerequisites
- SDK & build tools
- Compile platform
- Support libraries

Automated install
- Android Gradle Plugin 2.2+
- android-sdk-manager Gradle plugin
- Pre-prepared container
Prerequisites: Android Gradle Plugin

build.gradle

buildscript { 
    dependencies { 
        // Version 2.2 or newer can auto-install components 
        classpath 'com.android.tools.build:gradle:2.2.0'
    }
}

gradle.properties

# Enable SDK auto-installation (default is false) 
android.builder.sdkDownload=true
def withAndroidSdk(String sdkDir = '/tmp/android-sdk',
    Closure body) {

    // Create the SDK directory, and accept the licences
    // (see: d.android.com/r/studio-ui/export-licenses.html)
    writeFile file: "${sdkDir}/licenses/android-sdk-license",
        text: 'e6b7c2ab7fa2298c1...
        ...5d1a37fbf41ea526'

    // Run the given closure with this SDK directory
    withEnv(["ANDROID_HOME=${sdkDir}" ])
        { body() }
node {

  // Check out the source code
  git 'https://github.com/googlesamples/android-topeka' 

  // Build the app using the 'debug' build type, 
  // and allow SDK components to auto-install
  withAndroidSdk {
    sh './gradlew clean assembleDebug'
  }

  // Store the APK that was built
  archive '**/*-debug.apk'
}

Demo

Building an APK & basic checks
Testing
Test frameworks & tools

JUnit plugin
xUnit plugin
...

appium
Running unit tests

```javascript
node {
  // Check out the source code
  git 'https://github.com/googlesamples/android-topeka'

  // Build the app using the 'debug' build type, and allow SDK components to auto-install
  withAndroidSdk {
    sh './gradlew clean assembleDebug testDebugUnitTest'
  }

  // Analyse the JUnit test results
  junit '*/TEST-*.xml'

  // Store the APK that was built
  archive '*/*-debug.apk'
}
```
Supported devices
3572
Matrix jobs

<table>
<thead>
<tr>
<th>Configuration Matrix</th>
<th>2.3.3</th>
<th>3.0</th>
<th>4.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_GB</td>
<td>ldpi</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xhdpi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>de_DE</td>
<td>ldpi</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xhdpi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nl_NL</td>
<td>ldpi</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xhdpi</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Demo

Unit & UI testing
Delivery
Deploying an APK to users

**Basic**
- Upload to web server with "Publish over..." plugins
- Users download and install APK manually

**Third-party solutions**
- Plugins available for app test/distribution services
  - e.g. HockeyApp, Crashlytics Beta

**Google Play alpha/beta testing**
- Publisher API was launched in mid-2014
  - Upload directly from Jenkins
Google Play Android Publisher Plugin

com.example.myapp

Alpha → Beta → Live
app/build.gradle

android {
    signingConfigs {
        release {
            storeFile file('..\release.keystore')
            keyAlias 'android'

            // Passwords will be injected by Jenkins during build
            storePassword System.env.RELEASE_KEYSTORE_PASSWORD
            keyPassword System.env.RELEASE_KEY_ALIAS_PASSWORD
        }
    }
}
Deploying from `git tag` to Google Play

Tag Git repo at any point with `alpha/<whatever>`

- Release notes may be attached as tag message

Jenkins job, triggered only by new `alpha/*` tags

- Tag message is exported to the environment

Inject signing keystore password into environment

Build and sign the app

Upload the APK to Google Play

- Tag message can be used as “Recent Changes” text
node {
    // Check out source from latest alpha tag
    checkout([$class: 'GitSCM',
        userRemoteConfigs: [[url: 'https://github.com/...',
            refspec: '+refs/tags/alpha/*:refs/remotes/origin/tags/alpha/*'],
            branches: [[name: '*/tags/alpha/*']]])

    // Create a credentials binding for the signing key password
    signingKeyPw = [$class: 'StringBinding',
        credentialsId: 'my-app-signing-key-password',
        variable: 'RELEASE_KEYSTORE_PASSWORD']

    // Use the signing key to build the app release config
    withCredentials([$signingKeyPw]) {
        withAndroidSdk {
            sh './gradlew clean assembleRelease'
        }
    }
}
Deploying to Google Play

node {

    // Upload APKs to Google Play using the given credential
    androidApkUpload googleCredentialsId: 'Google Play',

    // Upload all APKs found in the workspace
    apkFilesPattern: '**//*.apk',

    // Publish to alpha users only
    trackName: 'alpha',

    // Set the ‘recent changes’ text
    recentChangeList: [
        [language: 'en-GB', text: "Hey, ${BUILD_NUMBER}"],
        [language: 'de-DE', text: "Hallo, ${BUILD_NUMBER}"],
    ]
}

Demo

Git push to Google Play
Continuously building and delivering

Build
Commits being built in a clean environment
APKs are archived and available for every commit

Quality
Every APK can be traced back to a Jenkins build
Commits being tested, on multiple device configs

Delivery
Releases can be made at any time simply by tagging
APK signing happens in a secure environment
Thanks!

chris@orr.me.uk
github.com/orrcc
twitter.com/orrcc