

Android Release Process

Team Members

- [Sweta Shinde](#)

This is a work in **WORK IN PROGRESS** - [Sweta Shinde](#) or garett.petersen@autogravity.com to document

Current Coverage

Platform	Source Code	Automated Build Software	Build Failure Mechanism	Unit Test Coverage	Integrated Tests	Functional Tests/Automation Software	Automation Source Code	Automation Failure Mechanism	Release Manager Role	Confluence
Android	Git=Android		N/A	Android=	N/A	N/A	ios-turncar automation tests		• Sweta Shinde	

Branch Strategy Trunk Based Development

- There is only one branch, there are no other branches, so there is no merging.
- No merging equals no merge conflicts.

Code Compile

1. Create a branch
 - a. Pull request temporary feature branch
2. Dependency Analysis - Developer checks for dependencies
 - a. Microservices
 - b. Feature Flag On/Off
 - c. Library dependency analysis Tools Example: NPM, [OSSIndex](#), [Bundler Audit](#)
3. Code analysis ([Peer Review](#)) **APPROVED** trunk/master
 - a. Compile + Test + Build: Merge to trunk/master Jenkins triggers a build Build is Tagged
 - i. Tests: Unit Tests Integration Tests Acceptance Testing: E2E Cypress Tests **PASSES**
 - ii. If Test **FAILS**
 - b. Event Notification
 - i. Slack to notify events/status
 - ii. This should go up on Screen/Board if developers are co-located and email notification should be sent out

Deploy Acceptance Release Candidate

- Mobile iOS Develop Passed [Docker](#) Staging
1. Deploy **release candidate** to Staging
 2. Archive to executable Version has to updated by this point
 - a. Unit Tests Integration Tests Acceptance Testing: Mobile Manual (Feature-based testing) **PASSES**
 - b. Promote Bumped Version Release.YYYYMMDD.[SequenceNumber], HotFix.YYYYMMDD.[SequenceNumber] [Semantic](#)
 - i. The sequence number is the production build sequence number + 1
 - c. Event Notification
 - i. Slack to notify events/status
 - ii. This should go up on Screen/Board if developers are co-located and email notification should be sent out

Production Check List

1. Dependency Analysis
 - a. Microservices
 - i. [Release Manager](#) checks for dependencies
 - b. Feature Flag On/Off
 - c. Library dependency analysis Tools Example: NPM, [OSSIndex](#), [Bundler Audit](#)

2. INFRA ticket
 - a. INFRA Ticket Must be Cleared
 - i. Creating INFRA Tickets
 1. This can happen anytime from the first commit preferably on 'Staging' branch aka 'Release Branch' but should be done when at this point
 - a. This needs to happen in **unison with the commit where the app version** is updated
 - b. There is code in the app that says if the version does not match in DB (supported versions table - utms.version_support) Update Fails
 - i. Suggestion cleaner versioning to help bypass?
 - ii. QA & Stakeholder Must Approve
 1. INFRA ticket to deploy to production is created
 2. Creating INFRA tickets are required before any production environment change/release (script, new build, service restart)
 3. The INFRA ticket should contain a reference/description to all the tickets/features included in the release
 4. QA team managers should approve the INFRA tickets then an official approval should be provided by a director level or higher
 5. The team agreed to post the INFRA ticket to the QA slack channel to provide visibility then ask for confirmation from the QA team

Submit to Google Play Store

Hotfix and Unplanned Releases

1. Branch off `master` Verify Fix Cherry-Pick back to Release/Production
2. Code analysis (Peer Review) **APPROVED**
3. Jenkins triggers a build
 - a. Tests: Unit Tests Integration Tests Acceptance Testing: E2E Cypress Tests Manual/QA Verify Fix **PASSES**
 - b. Event Notification
 - i. Slack to notify events/status
 - ii. This should go up on Screen/Board if developers are co-located and email notification should be sent out
4. Deploy fix to Staging Docker
1. Tests: Unit Tests Integration Tests Acceptance Testing: E2E Cypress Tests Manual/QA Verify Fix **PASSES**
 - Event Notification
 - a. Slack to notify events/status
 - b. This should go up on Screen/Board if developers are co-located and email notification should be sent out
2. Deploy to Production Cherry-Pick to Release
3. Promote Bumped Version Release.YYYYMMDD.[SequenceNumber], HotFix.YYYYMMDD.[SequenceNumber] [Semantic](#)
 - a. The sequence number is the production build sequence number + 1
4. Dependency Analysis
 - a. Microservices
 - i. Check for dependencies
 - b. Feature Flag On/Off
 - c. Library dependency analysis
5. Mobile iOS Stakeholders Approval Apple Store Production
6. Notification: Release Notes