

HWSW - MOBILFEJLESZTŐI meetup

Mobil CI/CD a felhőben és pár
Android tipp

Bazsonyi Tamás

SENIOR SOLUTIONS ENGINEER



Bemutakozás



Eddigi tapasztalatok:

- Fullstack fejlesztő
- QA Engineer (manuális és automatizált tesztelés, mobil szoftverfejlesztés főleg új feature implementálás, CI/CD pipeline kiépítés/karbantartás)
- CI/CD felhasználók támogatása Support/Sales oldalról



Leggyakoribb kihívások

- Rugalmatlan CI szerver, ami könnyen elromlik
- Sok időt felemésztő tesztelési folyamatok (release sign-off)
- 2 havonta release
- Bug-ok és crash-ek az éles app-ban
- Negatív értékelések

És ne feledjük...

A mobil alkalmazásoknál nehéz jóvátenni a hibákat!



**Hogy küzdhetjük le
ezeket a kihívásokat?**

Mobile DevOps

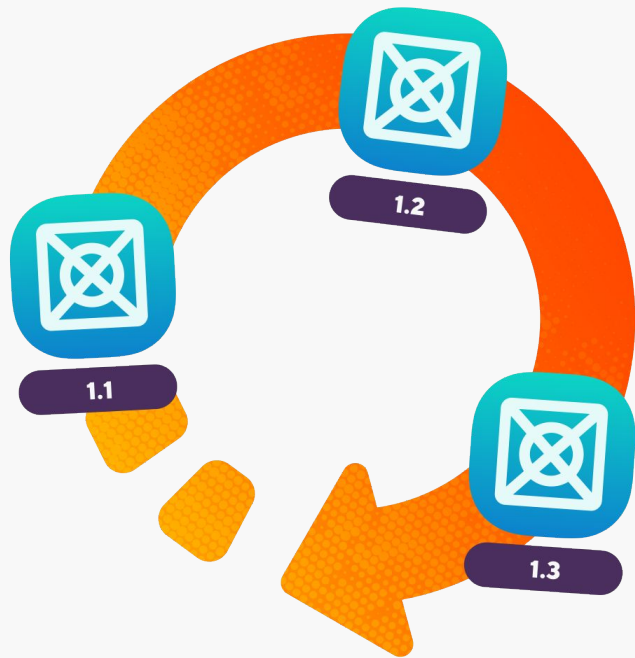


Mobile DevOps

Mobil CI/CD a **build és release folyamatok automatizálására**, hogy elérjük a **gyors és sűrű release ütemezést**

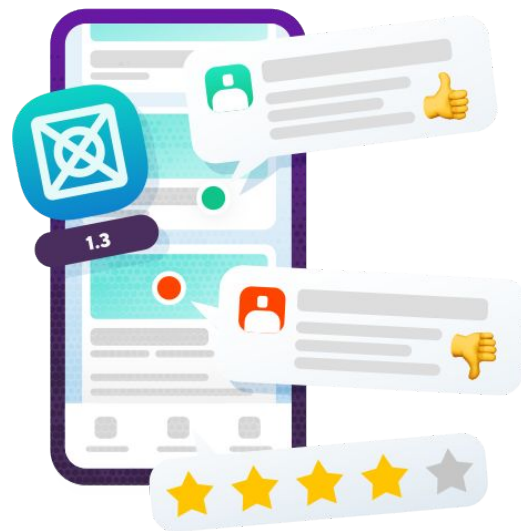
A Mobile DevOps kulcsfontosságú elemei:

- Folytonos kommunikáció és tervezés
- Continuous Integration, Delivery, Deployment
- Folytonos tesztelés és monitorozás



Mobile DevOps as a Service

- Source Code management
- Artifacts management
- CI/CD pipeline-ok
- Unit, Integration és UI tesztek
- Automatizált App Deployment
- Infrastruktúra





People

Culture
Mindset
Commitment



Process

Agile/Scrum
RFC
Code Review
Incident Management
Feature Flags



Tools

Test Automation
Reporting
Releasing
Monitoring



**Mobile
DevOps**

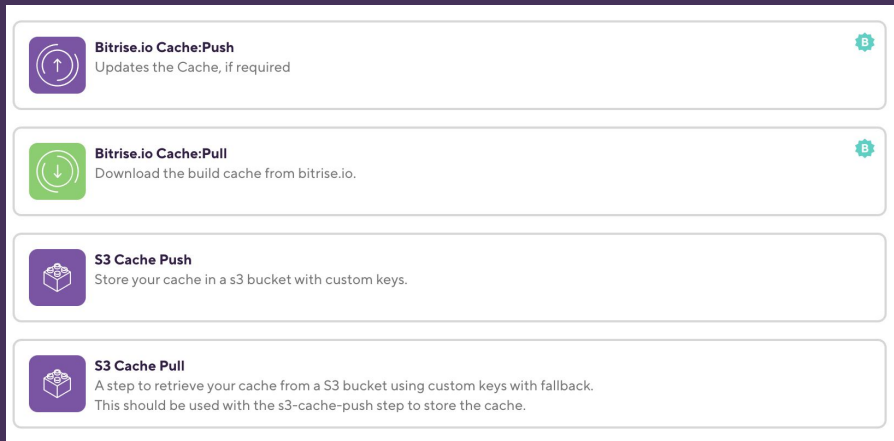
Continuous Integration
Continuous Delivery
Continuous Deployment

A Mobil DevOps előnyei

- A sikeres együttműködés alapja az érdekelt felek, product manager-ek és fejlesztők között
- Felgyorsítja a Release-hez jutás folyamatát
- Biztosítja a kód/alkalmazás minőségét
- Felgyorsítja a hibák felfedezésének és javításának folyamatát
- Növeli a felhasználók elégedettségét
- Növeli a cég dolgozói részéről az együttműködést

Cache használat

- Egy mobil CI/CD pipeline-nál minden új build egy új virtuális gépen (VM) fut. Emiatt minden alkalommal újra kell telepíteni a használt tool-okat
- Az alkalmazás által használt dependency-k letöltése minden alkalommal rengeteg időt emészt fel feleslegesen. A cache ezek hatékony tárolására és újrahasználatára szolgál.



The screenshot displays a vertical list of four pipeline steps in a Bitrise interface. Each step is represented by a colored square icon with a circular arrow, followed by its name and a brief description. The steps are: 1. 'Bitrise.io Cache:Push' (purple icon) with the description 'Updates the Cache, if required'. 2. 'Bitrise.io Cache:Pull' (green icon) with the description 'Download the build cache from bitrise.io.'. 3. 'S3 Cache Push' (purple icon) with the description 'Store your cache in a s3 bucket with custom keys.'. 4. 'S3 Cache Pull' (purple icon) with the description 'A step to retrieve your cache from a S3 bucket using custom keys with fallback. This should be used with the s3-cache-push step to store the cache.'. Each step also features a small blue 'B' icon in the top right corner.

Bitrise.io Cache:Push
Updates the Cache, if required

Bitrise.io Cache:Pull
Download the build cache from bitrise.io.

S3 Cache Push
Store your cache in a s3 bucket with custom keys.

S3 Cache Pull
A step to retrieve your cache from a S3 bucket using custom keys with fallback.
This should be used with the s3-cache-push step to store the cache.

UI Automatizált Tesztelés

- A Teszt Automatizálás egy kulcsfontosságú eleme a Mobil DevOps-nak
- A **Tesztelés** egy **teljes** csapatot érintő folyamat
- Teszt Keretrendszerek és Tool-ok
- Konfiguráció és eszközök

Q Enter what you are sea

Android

Access control

Artifact info

Build

Code sign

Dependency

Deploy

Installer

Notification

Test

Utility

Test



Android Detekt

This step runs detekt gradle task on your project.



Android Lint

Runs Lint on your Android project source files and detects potential syntax errors to keep your code error free.



Android Unit Test

This step runs your Android project's unit tests.



App Center Launch Test Android

Run App Center launch test for android



App-Ray Mobile Security

Mobile application security testing with App-Ray's engine via API



App Center upload and schedule tests

This step uploads and schedules tests on Microsoft App Center.



Automatizált Deployment

- A tesztelőknek való elküldés vagy az alkalmazás üzletbe történő frissítés feltöltés automatizálható elemei többek között: code signing, profile provisioning, verziózás, release alkalmazások elkészítése, meta adatok vagy képernyőképek összeállítása.
- Az APK **digitális aláírása (sign)** és a **Play Store-ba feltöltése** könnyedén scriptekkel is megoldható

The screenshot displays the Bitrise Workflow Editor interface. At the top, the breadcrumb navigation shows 'Dashboard / Awesome Org / reactive-detox-bitrise / Workflow Editor'. Below this is a menu bar with tabs: 'Workflows', 'Code Signing', 'Secrets', 'Env Vars', 'Triggers', 'Stacks & Machines', and 'bitrise.yml'. On the right of the menu bar are 'Discard' and 'Saved' buttons. The main area is divided into two panels. The left panel, titled 'WORKFLOW', shows a list of steps for the 'Android-CI-CD' workflow, including 'Gradle Runner', 'Install missing Android SD...', 'Android Build', 'Android Sign', 'Change Android versionCo...', 'Google Play Deploy', 'Send a Slack message', and 'Bitrise.io Cache:Push'. The right panel shows the configuration for the 'Google Play Deploy' step. It includes fields for 'Expansion file Path', 'Track' (set to 'beta'), 'User Fraction', 'Status', 'Name of the release', and 'Update Priority'. Each field has a 'CHANGE' button next to it.

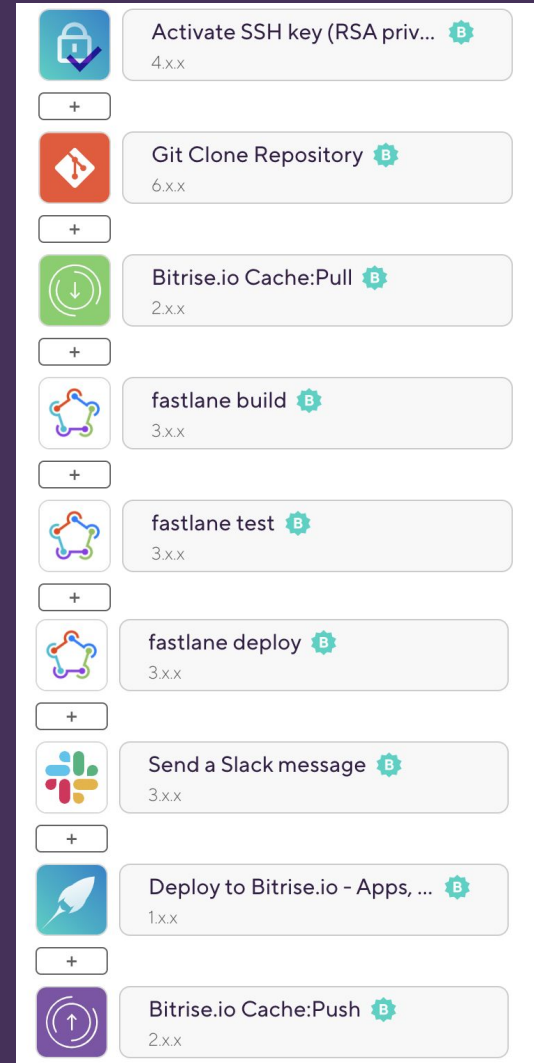
Folyamatok Automatizálása



fastlane

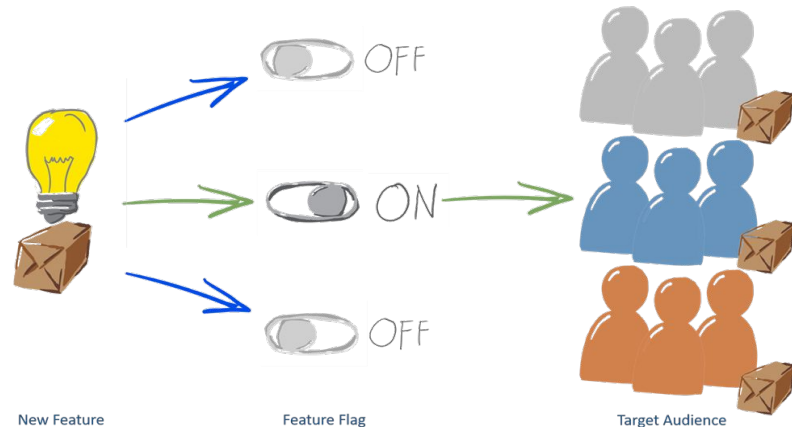
A fejlesztési és release folyamatok automatizálása

- Fastlane - Egy nyílt forráskódú tool aminek a célja az Android és iOS folyamatok automatizálása
- fastlane
 - Automatizált Képernyőképek készítése
 - Beta Deployment
 - App Store Deployment
 - Code Signing



Feature Flag-ek

- A Feature flag-ek segítenek bizonyos feltételek alapján elérhetővé tenni különböző funkciókat a felhasználók bizonyos csoportjának
- A Feature flag-ekkel könnyen lehet A/B tesztelést végrehajtani, csökkenteni a fejlesztések kockázatát és gyorsabban reagálni a felhasználók visszajelzéseire



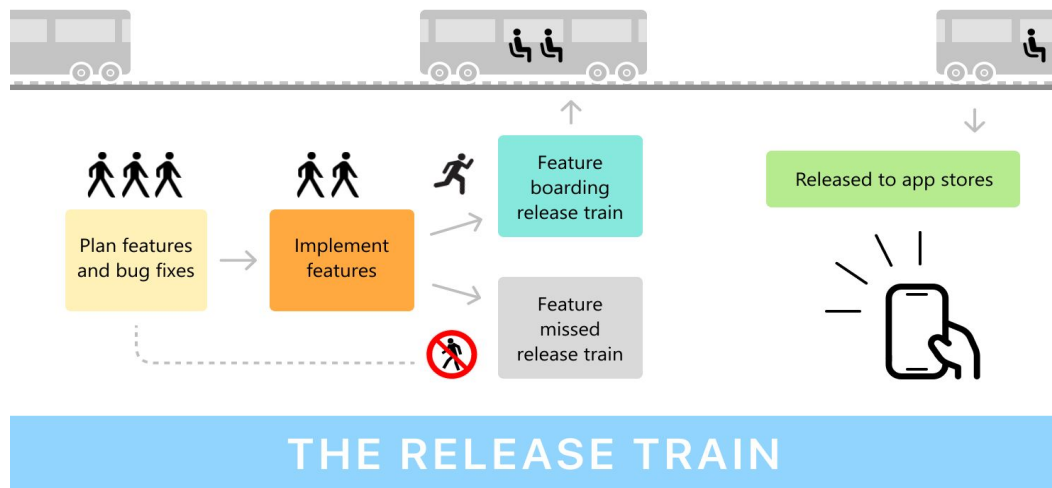
A Mobil Release Train

- Ha a CI folyamatok már összeálltak, akkor a **build train** segítségével felgyorsulnak a release folyamatok
- Jó módja a folyamatok vizualizációjának miközben biztosítja a hatékony kommunikációt
- Segít, hogy a release-ek **konzisztensebbek** legyenek valamint az elosztott csapatok összhangban maradjanak



Mobil Release Train

- A fejlesztési fázis során a csapat teszteli, jóváhagyja és merge-öli a változtatásokat a main branch-re (**miközben a master branch végig “zöld”**)
- Amikor a release tartalmát meghatározták és elkészült, akkor következik egy utolsó teszt fázis
- A csapattagoknak le kell ellenőrizni, hogy az új fix-ek/feature-ök ténylegesen működnek



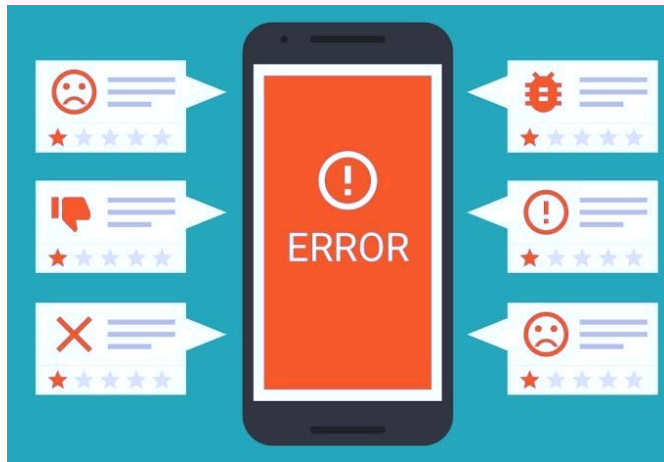
Útmutató az implementáláshoz

- Legyen egy előre meghatározott train beosztás
- Legyen egy kinevezett release manager
- Feature flag-ek bevezetése a magabiztosabb release-ekért és staged rollout-ok bevezetéséért
- A folyamaton folyamatosan iterálva fejleszteni kell

Monitoring (Teljesítmény és rendelkezésre állás)

A Mobil alkalmazások rengeteg okból **crash**-elhetnek. A különböző verziók **monitorozásával** sok esetben a felhasználók előtt fel lehet fedezni a különböző hibákat ezzel fokozva a felhasználóknak nyújtott élményt.

Ezalatt több különböző aspektusból kell figyelemmel követni az app-ot, **erőforrások használata, alkalmazás használat és stabilitás (crash-ek, hibák)**



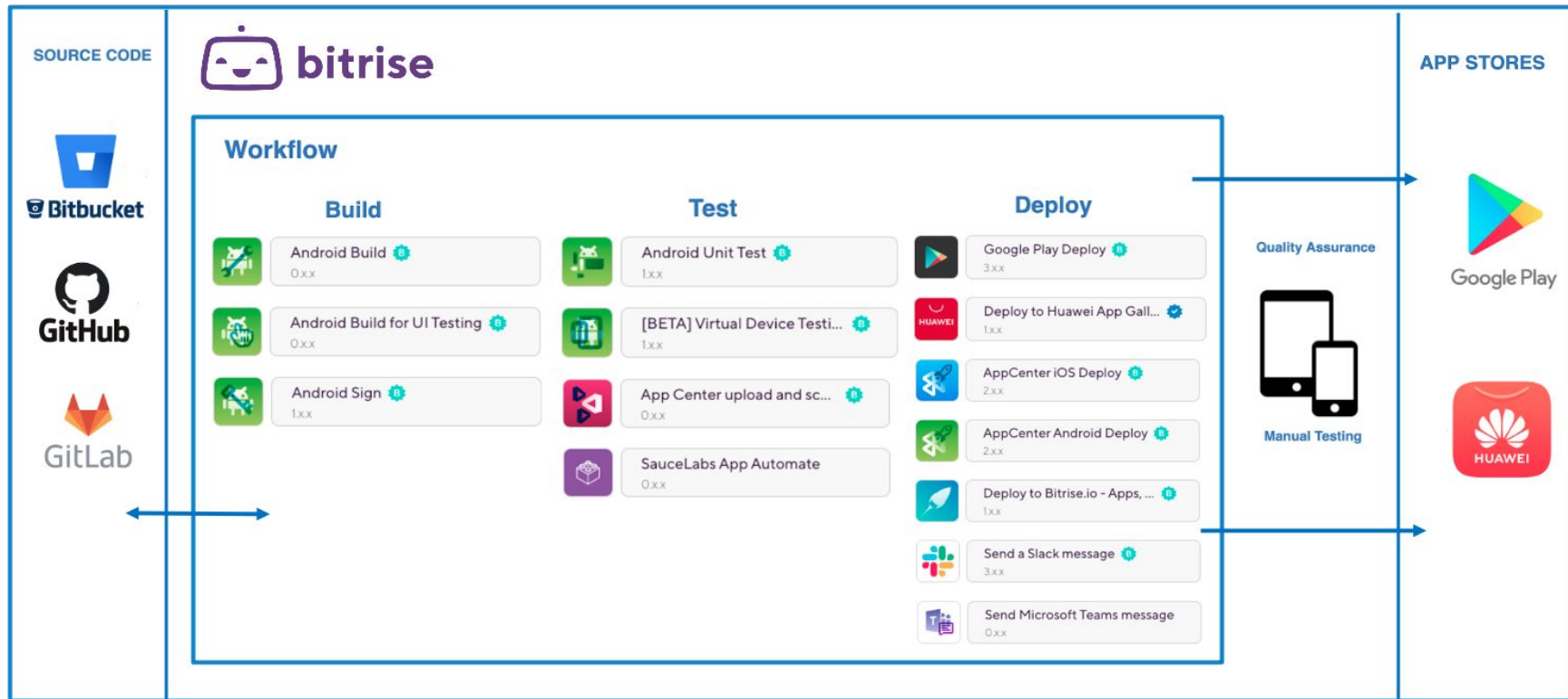
Android CI/CD Workflow

Continuous Integration

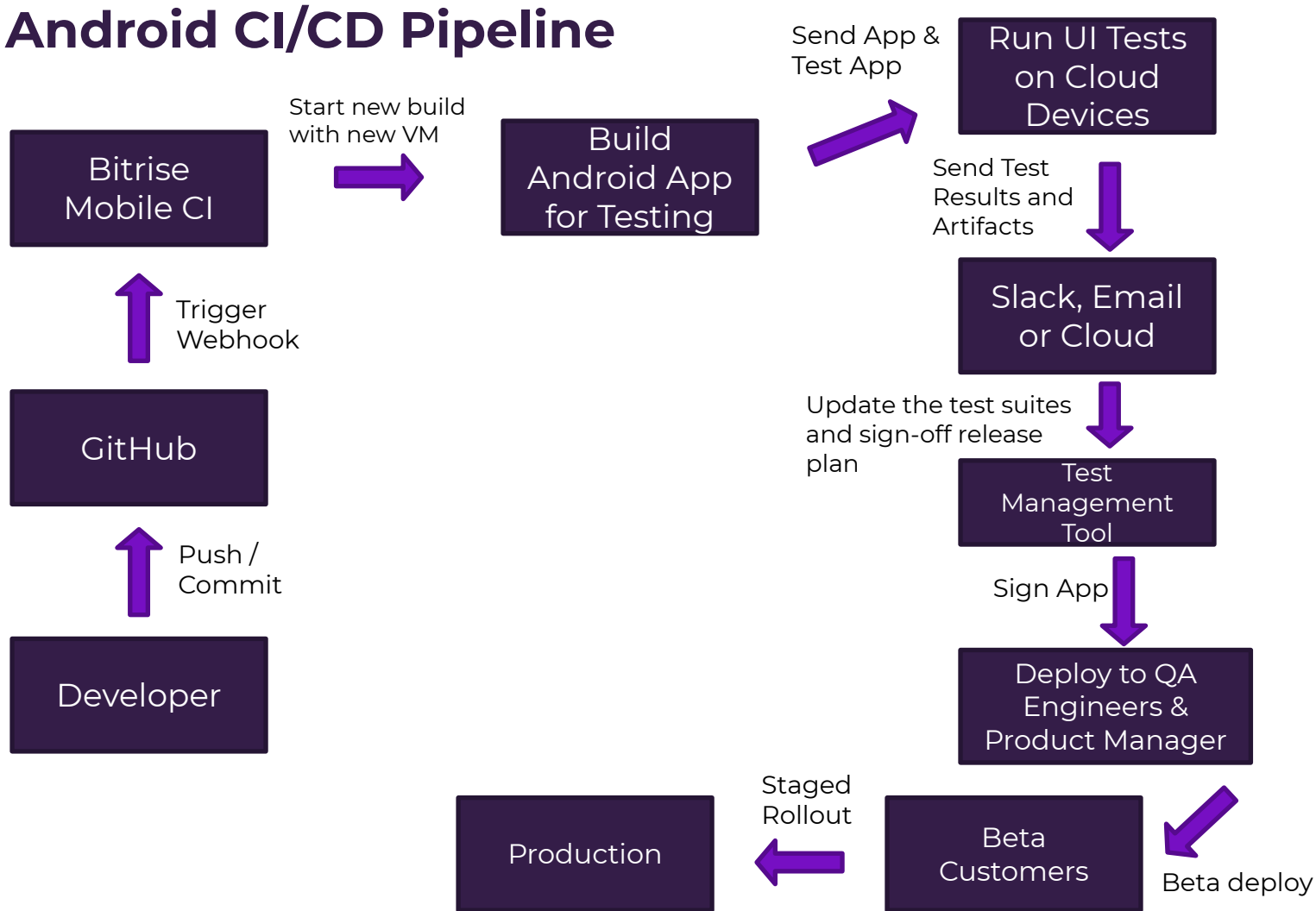
Continuous Delivery/Deployment



Release Lifecycle



Android CI/CD Pipeline



Áttérés felhőben host-olt & Mobil CI/CD-re

- A mobil app release-ek a megfelelő tool-ok nélkül egy elég komplex folyamat
- A mobil fókuszú CI/CD rendszerek az egyedi mobil specifikus kihívásokat tartják szem előtt és mobil specifikus feature-ökkel segítenek a build, teszt és deploy folyamatokban

Amikről Beszéltünk

- A Mobil DevOps **mindset, culture, folyamat** és **tool-ok**
- **A minőség** fontosabb, mint a mennyiség
- **Teszt Automatizálás** a Mobil DevOps kulcsfontosságú része
- **CI/CD** a koordinálási folyamat
- **A tesztelés** az egész csapat közös felelőssége
- **Az felhasználók** a **legfontosabbak**

Q&A Köszönöm!



@bitrise



bitrise-io



bitrise.io



bitrise



bitrise.io