



Mostoha Roland

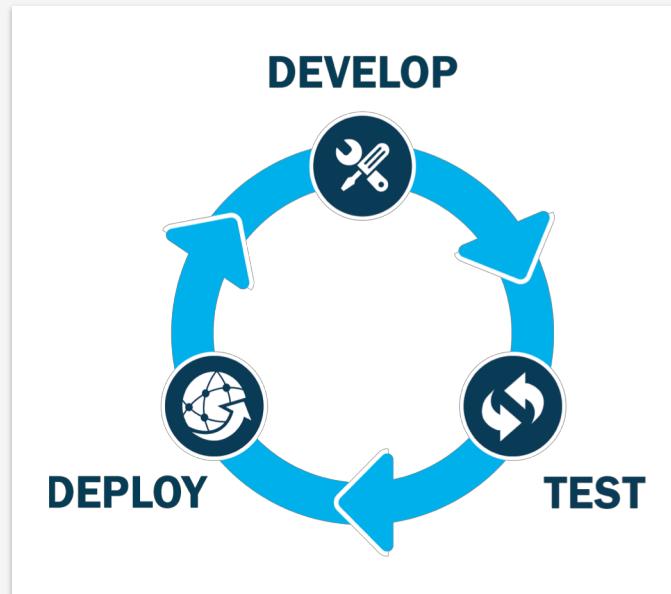
SquadLead @ AutSoft

Alkalmazás publikálás, Continuous Integration és
Continuous Delivery



Alkalmazás publikálás, CI és CD

Miért fontos ez a téma?



- minden ami fáj azt automatizáljuk!
 - Build és tesztek futtatása különböző konfigurációkkal
 - Alkalmazás kipróbálása különböző környezeteken
 - Aláírás
 - Release
 - Publikálás ...
- minden ami fontos azt kényszerítsük ki!
 - minden fejlesztési iteráció után legyen az alkalmazás
 - Futtatható
 - Telepíthető
 - Konzisztens
 - Zöld (a tesztek szempontjából)



Mit vihetünk hazá?

- Betelekintést nyerhetünk egy teljes alkalmazás kiadási folyamatába
- Kipróbálhatunk cloud alapú CI/CD és device farm megoldásokat
- Egyedülálló, értékes tudás a piacon
 - Mobilos körökben még nem terjedt el annyira
 - Sok cégnél még nincs kialakulva a DevOps kultúra
- Saját cégünkönél úttörők lehetünk 😊
- Állásinterjún is hatalmas plusz pont
- Az előadás végére minden tudásunk meglesz, hogy egy alkalmazás kiadásának teljes folyamatát átlássuk és automatizáljuk



Agenda

1

CI/CD, publikálás alapok

- Continuous Integration
- Statikus kód analízis
- Alkalmazás publikálás

2

The PIN Project CI/CD gyakorlat

Alkalmazás integrációja

- Alkalmazás azonosító
- Alkalmazás aláírás
- Continuous Integration

Alkalmazás publikálása

- Play Store publikálás
- Publisher plugin
- Continuous Delivery

Alkalmazás fejlesztési életciklus



- Milyen konfliktusokhoz vezethet ha több fejlesztő dolgozik egyszerre?
 - Nem tudjuk, hogy épp konzisztens állapotban vagyunk-e a többiekkel.
 - Nem tudjuk, hogy minden funkciót tartalmaz-e a release, amit kiadunk.
 - Nem tudjuk mi történik, ha mergeljük a kódunkat.
- Milyen konfliktusokhoz vezethet ha egy fejlesztő több helyről is dolgozik?
- Megoldás: **Folyamatos integráció és ellenőrzés**
 - Csökkentsük a „távolságot” a fejlesztők és a munkaállomások között
 - A lehető leghamarabb végezzük el az integrációt
 - Egy független bíráló ellenőrizze a fenti pontokat



Mi a Continuous Integration?

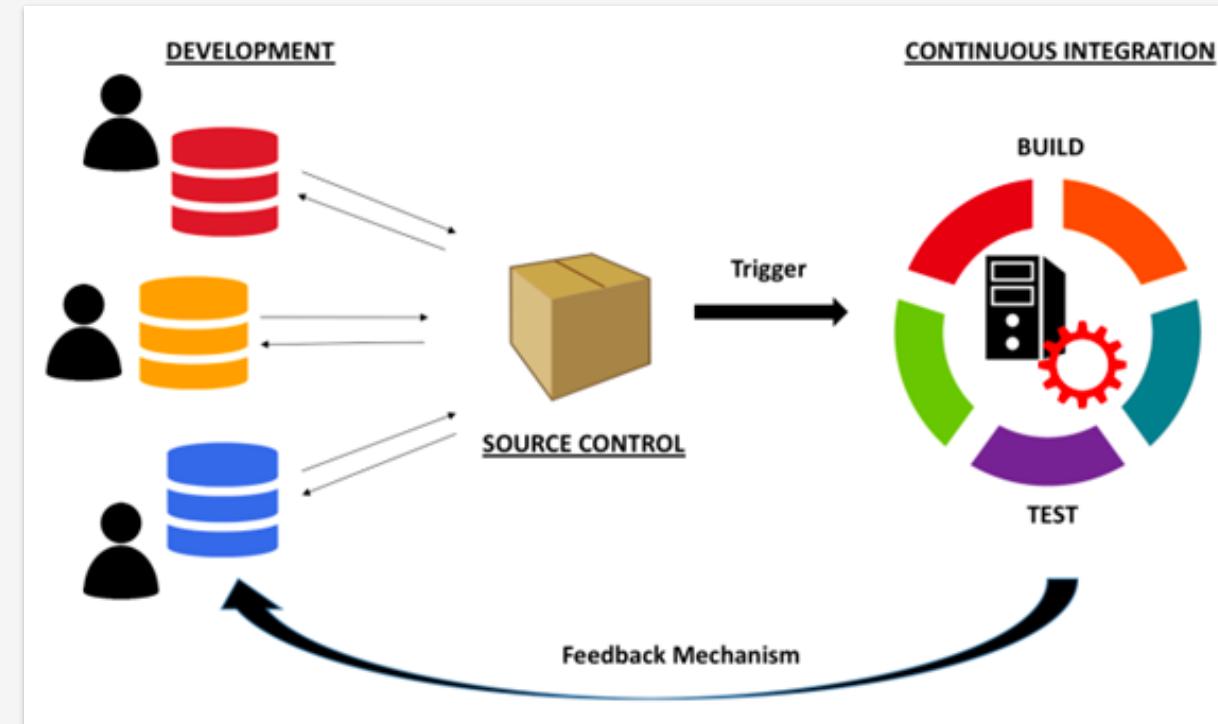
„Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.”

- Martin Fowler

<https://martinfowler.com/articles/continuousIntegration.html>



Continuous Integration visszajelzés



Continuous Integration feltételei

By Martin Fowler



1. Egy közös verziókezelő rendszer üzemeltetése – Git, SVN ...
2. Automatizáljuk a build folyamatot
 - Teljes fordítás egy parancssal
 - Javasolt trigger esemény: commit
3. Legyen a build ön-tesztelő
 - Automatizált tesztek
 - Kód analízis
 - Teljesítmény tesztek
4. minden nap legyen commit a fő branch-re
 - A kód legfrissebb állapota mindig integrálható
 - Minél több idő tellik el, annál nagyobb galibát okozhat egy integráció

Continuous Integration feltételei

By Martin Fowler

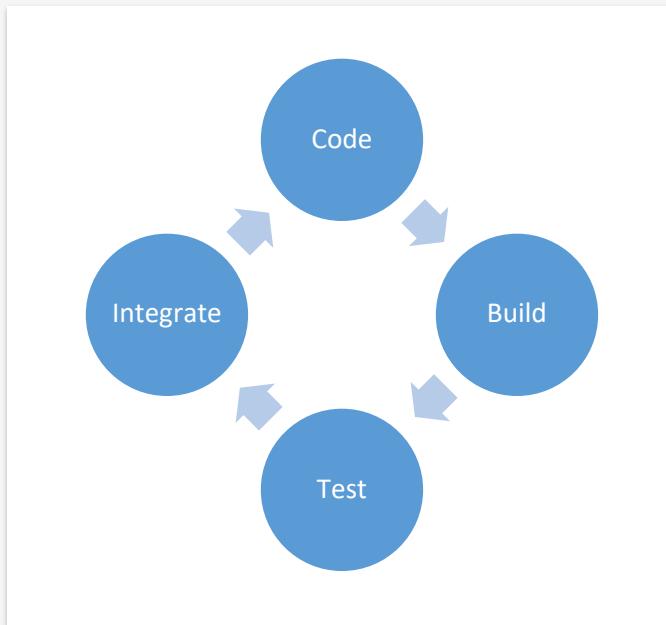


5. minden commit triggereljen egy buildet az integrációs szerveren
 - Független "bíráló"
 - Független környezeten
6. legyen gyors a build
 - minél gyorsabb visszajelzés az integráció eredményéről
7. a tesztelés az éles környezet "klón" környezetén történjen
 - Google Play Alpha/Beta
8. legyen egyszerű elérni a legfrissebb futtatható állomány
 - APK archiválása
9. a teljes build folyamat és a projekt állapota legyen transzparens mindenki számára
 - Riasztások, értesítések



CI/CD, publikálás alapok

Continuous Integration előnyei

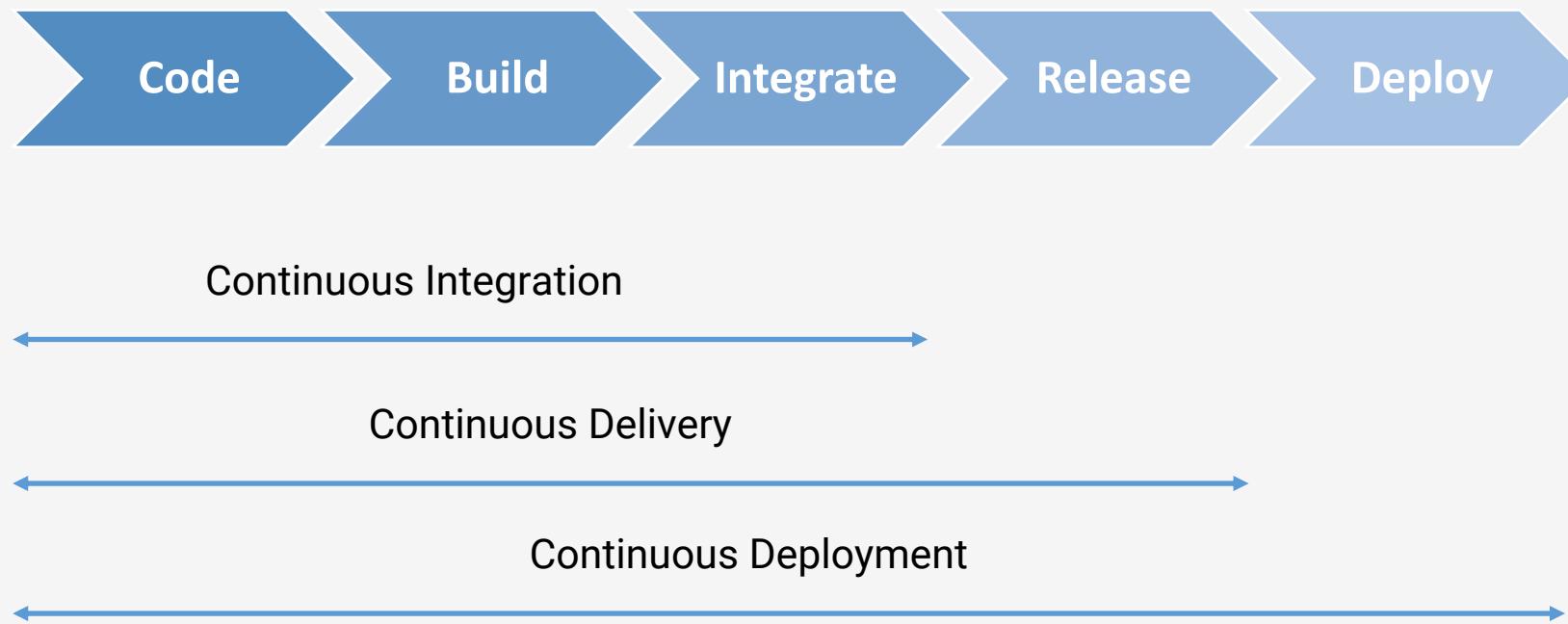


- Csökken az integráció és az integrációs hibák költsége
- Garantálja, hogy az életciklus teljes folyamata le fog futni
- Folyamatos visszajelzés az integráció eredményéről
- Az integrációs hibák azonnal észlelhetőek
- Az alkalmazás aktuális állapota teljesen transzparens
- Az alkalmazás mindenkor elérhető, naprakész és demozható
- Az alkalmazás mindenkor készen áll a publikálásra



Continuous Integration, Delivery, Deployment

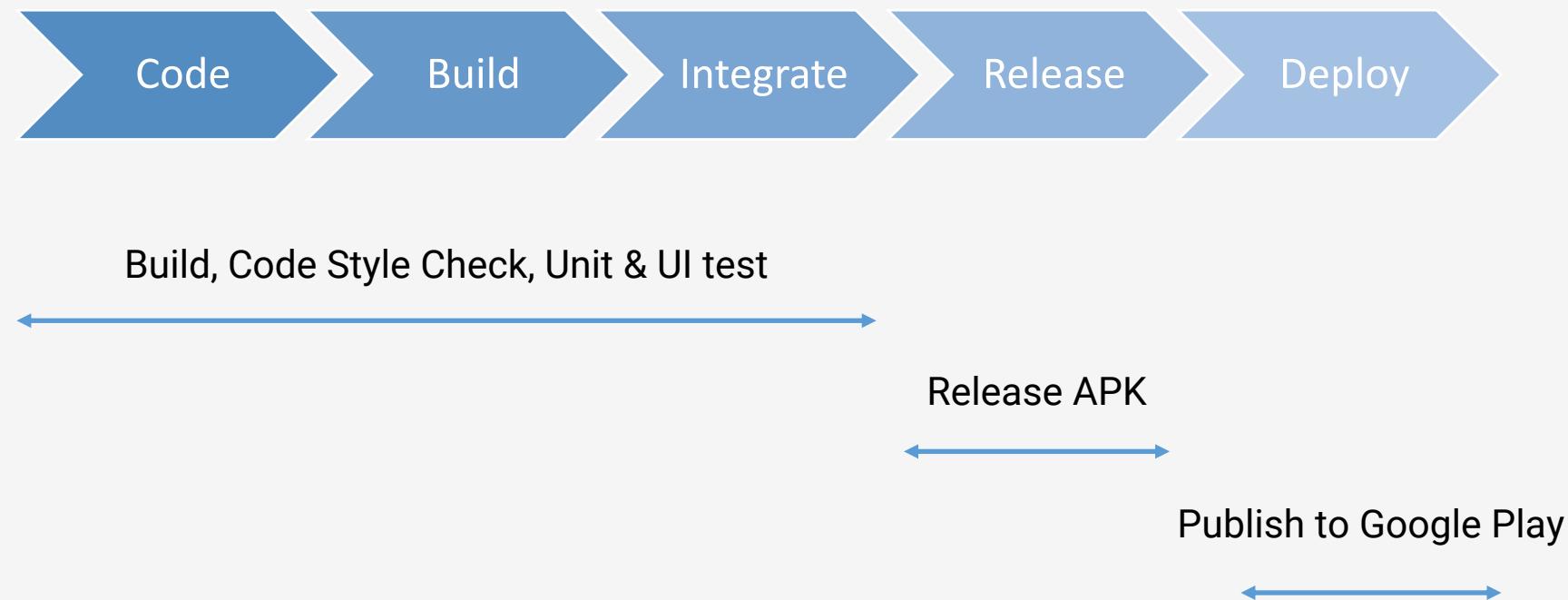
- Automatizált végrehajtás





Continuous Integration, Delivery, Deployment

- A gyakorlatban





CI/CD, publikálás alapok

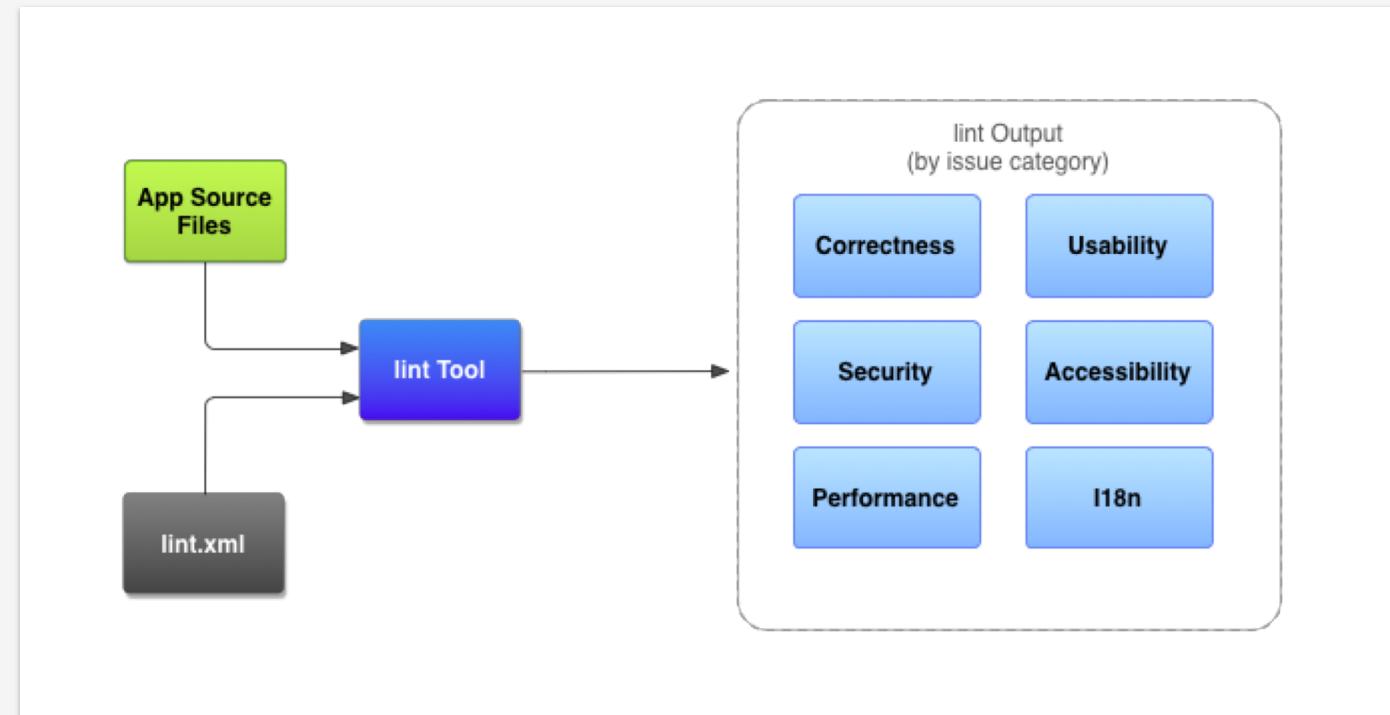
Statikus kód analízis eszközök



- Statikusan vizsgáljuk a kódminőséget, esetleges hibákat
- Android specifikus ellenőrzések
 - Android Lint
- Nyelv specifikus ellenőrzések
 - Kotlin – [detekt](#)
 - FindBugs
- Teszt specifikus ellenőrzések
 - Jacoco



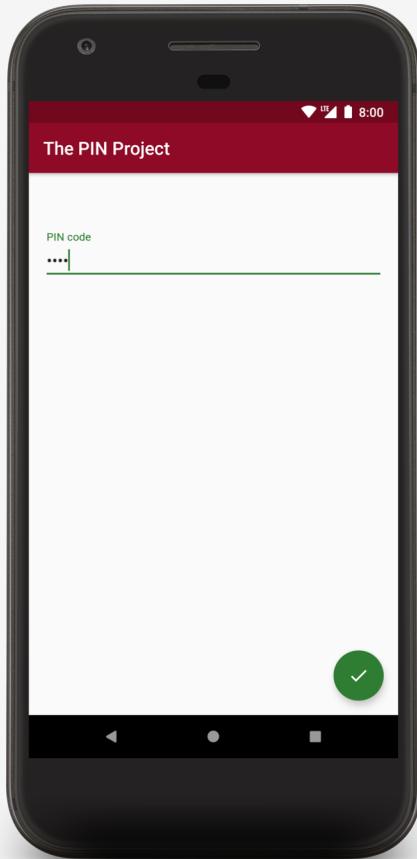
Android Lint





CI/CD, publikálás alapok

The PIN Project



Végezzünk el egy Lint ellenőrzést Android Studio segítségével!



<https://developer.android.com/studio/write/lint>



CI/CD, publikálás alapok

Google Play Store publikálás



- Google Play Developer account szükséges: 25 \$
- Az alkalmazás első változatát kézzel kell kiadni
- Extra funkciók
 - Statisztikák
 - Crash reportok
 - Pre Launch report
- Launch checklist



CI/CD, publikálás alapok

App publikálás követelmények



Google Play

- App releases
 - Alkalmazás verziók
- Store listing
 - Google Play Store megjelenés
- Content rating
 - Tartalmi besorolás
- Pricing & Distribution
 - Ingyenes/fizető
 - Támogatott nyelvek és országok



CI/CD, publikálás alapok

App releases

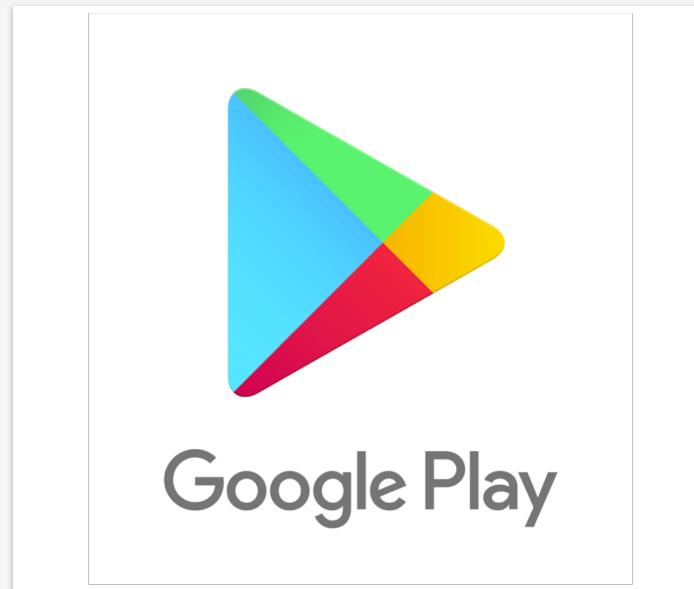


- APK feltöltés
- VersionCode ellenőrzés
- Track beállítás (Closed/Open/Production)
- Release Notes
 - Nyelvek szerint lebontva
- Tesztelői csoport hozzárendelése



CI/CD, publikálás alapok

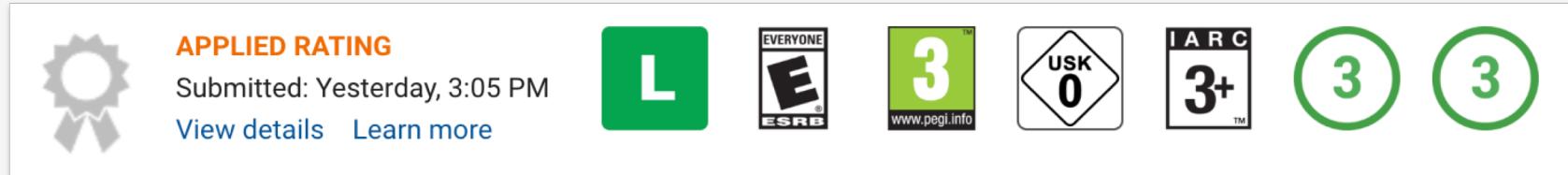
Store listing követelmények



- Cím, leírások
- Képernyőképek
- Alkalmazás típusa, kategóriája
- Fejlesztő elérhetősége
- Privacy Policy

Content rating követelmények

- Egy kérdéssort kell kitölteni, ebből meghatározza a besorolást





CI/CD, publikálás alapok

Privacy Policy követelmények



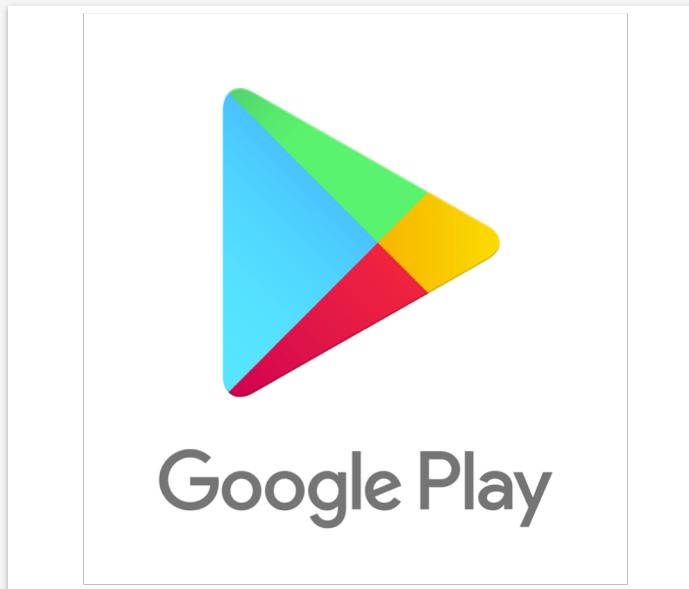
Google Play

- Ha az alkalmazás nem használ veszélyes engedélyeket (Pl. kamera, fiókok), akkor kihagyható
- Veszélyes engedélyek esetén kötelező egy publikált, online elérhető weblapon megosztani a GDPR-Ready Privacy Policy-t



CI/CD, publikálás alapok

Google Publish API v3



- Track (akár új track felvétele is lehetséges)
 - Internal Test
 - Closed Alpha
 - Open Beta
 - Production
- Draft releases
- Jobb Release notes támogatás



<https://android-developers.googleblog.com/2018/06/automating-your-app-releases-with.html>



Agenda

1

CI/CD, publikálás alapok

- Continuous Integration
- Statikus kód analízis
- App publikálás

2

The PIN Project CI/CD gyakorlat

Alkalmazás integrációja

- Alkalmazás azonosító
- Alkalmazás aláírás
- Continuous Integration

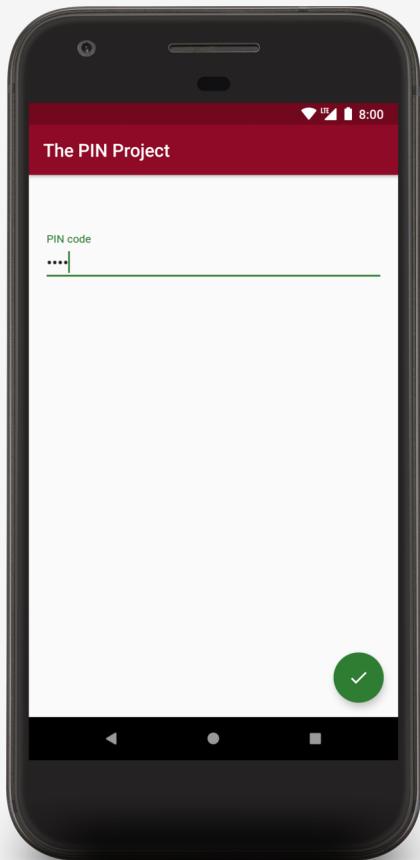
Alkalmazás publikálása

- Play Store publikálás
- Publisher plugin
- Continuous Delivery



CI/CD Gyakorlat

The PIN Project



A gyakorlat célja, hogy végigmenjünk egy teljes publikálási folyamaton és minden lépését automatizáljuk.

- Alkalmazás azonosító
- Alkalmazás áláírás
- Continuous Integration megvalósítása
- Continuous Delivery/Deployment megvalósítása
- Google Play Store publikálási folyamat vizsgálata



<https://github.com/RolandMostoha/the-pin-project-ci>



Alkalmazás azonosító

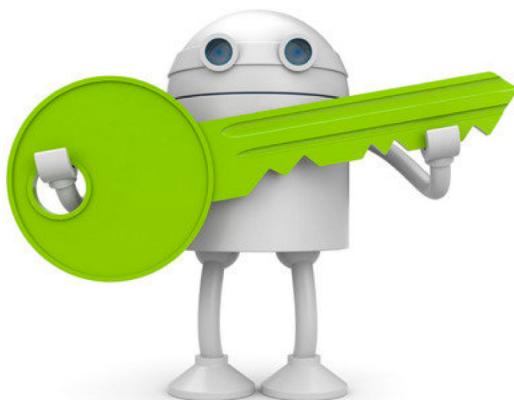
- Az alkalmazás egyértelmű azonosítására szolgál
- Google Play Store-ban egyedinek kell lennie
- 3rd party szolgáltatásokban egyedinek kell lennie (pl. Firebase)
- Adott esetben környezetenként is eltérő azonosítókat kell használni
 - Pl. Debug és Prod appok egyszerre telepíthetők legyenek
 - Pl. Firebase Database - Ne egy adatbázisba mentsünk Debug és Prod esetén
- **build.gradle**-ben közvetlenül megváltoztatható a **buildType** alatt

```
applicationIdSuffix ".debug"
```



CI/CD Gyakorlat

Alkalmazás aláírás

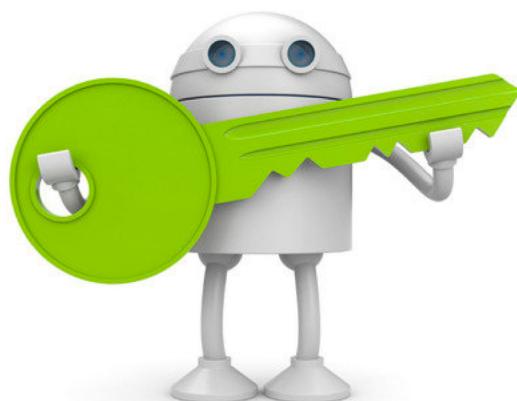


- Debug aláíró kulcs
 - Fejlesztési célokat szolgál
 - Kötött alias, password
 - Android SDK-ba alapból csomagolva van
 - Érdemes egy újat csinálni és megosztani
 - Fontosak a 3rd party szolgáltatások integrációjakor
- Release aláíró kulcs
 - Google Play Store publikálás



CI/CD Gyakorlat

Aláíró kulcs tárolása



- Verziókezelőben, pl. egy assets mappában
- Google Play App Signing
 - Lehetőség van a Google Play Store-ban tárolni
 - A Play minden (újra) aláírja az APK-t



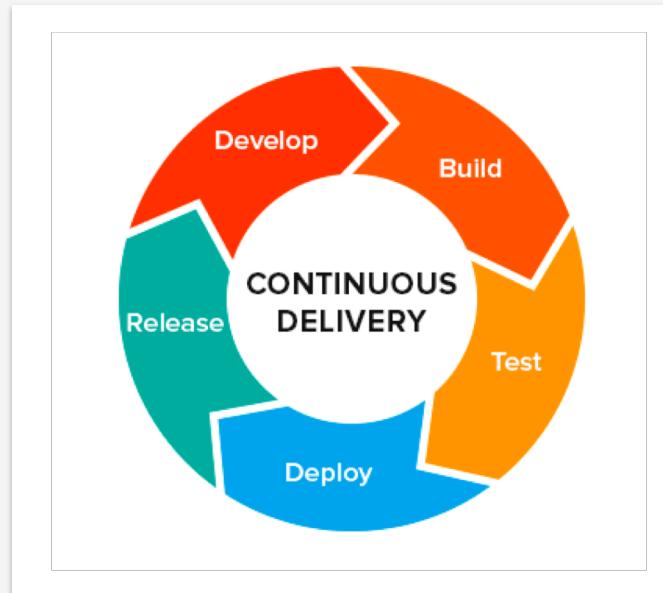
Continuous Integration eszközök

- Self-hosted – szoftver, saját gépen üzemeltett
 - [Jenkins](#)
 - [GitLab](#) – Git host szerver is
- Cloud alapú – szolgáltatás, a cég munkaállomásán üzemeltetett
 - [GitLab](#) - Git host szerver is
 - [CircleCI](#) – Könnyen testreszabható
 - [Bitrise](#) – Nagyon sok féle integráció



CI/CD Gyakorlat

Continuous Integration szempontok



- VCS támogtás (GitHub, Bitbucket...)
- Android SDK támogatás
- Docker konténer támogatás
- Könnyű legyen használni
- Ne kelljen menedzselni (cloud alapú)
- 3rd party integrációk
 - Slack
 - Firebase Test Lab



CI/CD Gyakorlat

CircleCI szolgáltatás

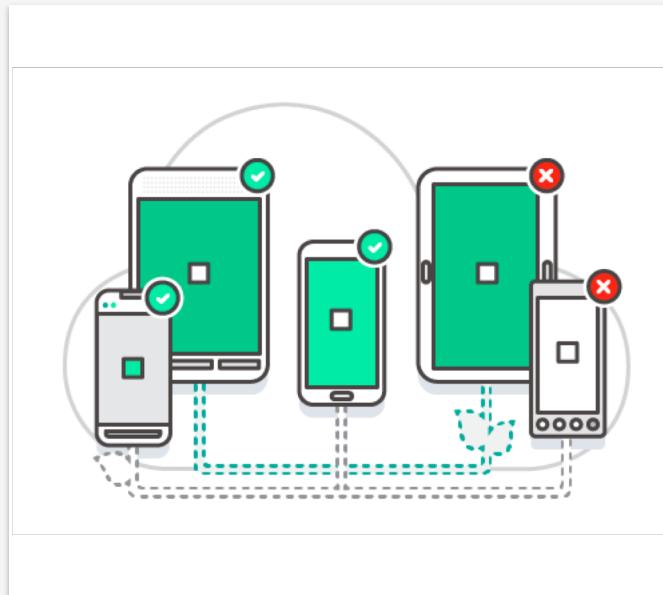


- Cloud alapú
- Docker konténer támogatás
- Default Androidos konténerek
- Ingyenes havi 1500 perc fordításig, egy szálon
- 3rd party integrációk
 - Firebase Test Lab



CI/CD Gyakorlat

Device Farm megoldások



- Távolról elérhető valós, virtuális eszközök
- Távolról futtathatóak instrumentation tesztek
- [Firebase Test Lab](#)
- [Amazon AWS Device Farm](#)
- [Samsung Remote Test Lab](#)



CI/CD Gyakorlat

The PIN Project



Végigmentünk egy teljes publikálási folyamatot és
 minden lépését automatizáltuk.

- ✓ Alkalmazás azonosító
- ✓ Alkalmazás álaírás
- ✓ Continuous Integration megvalósítása
- ✓ Continuous Delivery/Deployment megvalósítása
- ✓ Google Play Store publikálási folyamat vizsgálata



<https://github.com/RolandMostoha/the-pin-project-ci>

Köszönöm a figyelmet!

mostoha.roland@autsoft.hu

@RolandMostoha