

# 1. CI/CD

- 개요
    - Bitrise
  - pull request merge 완료 시 모든 내용 자동화
- 

## 개요

- 수정, 추가 개발하는 프로젝트의 개발 중인 내용이 아닌 시점의 자동 빌드 환경 구성이 필요하다.
  - 개발 중인 소스를 빌드하여 줄 수는 없으니 마지막 push 지점이나, Tagging 지점의 빌드가 필요한 경우.
- Webhook를 통해 release branch PR이 정상적으로 merge 되었을 경우 자동으로 빌드할 수 있다.
- 개발자가 App store에 잘 못된 apk (debugging 용으로)를 올릴 수도 있으므로, 실수를 방지하기 위함이다.
- CI/CD 도입.
  - Continuous Integration. (지속적인 통합)
  - Continuous Delivery (지속적인 서비스 제공)
  - Continuous Deployment (지속적인 배포)

### 소프트웨어 개발 파이프라인

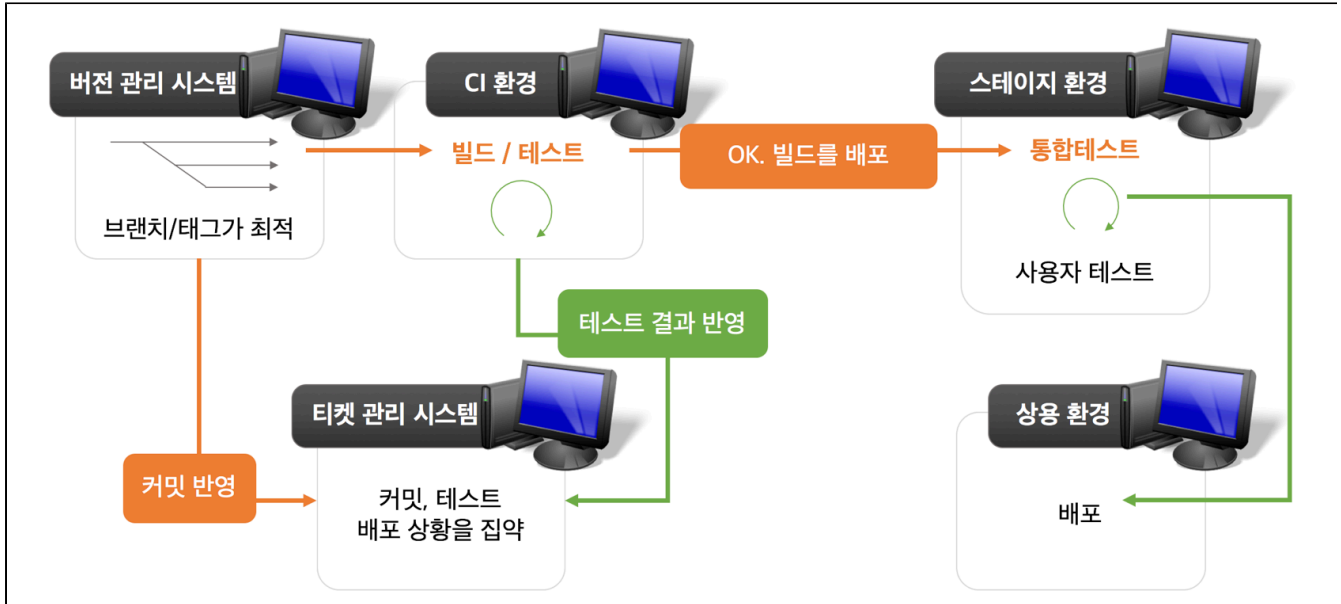


## CI 선택

- 제약 조건
  1. 적은 수의 팀원.
  2. 상시 변경되는 안드로이드 타겟 버전 관리.
  3. 자동화 시스템 관리를 안하는 방향.
  4. 정해진 설정관리에서 벗어나지 않는 방향.
  5. 통합된 툴과 무료로 사용할 수 있는 방향.
  6. 설치 없이 web 기반에서 바로 사용할 수 있는 방향.
  7. 옵션으로 선택하는 방향인데 pull request merge 완료 시 이후 과정들을 자동화로 하고 싶음.  
- 7번은 추후에 진행해보기. (web hook)
- Jenkins
  - 사용하는 회사들이 많기 때문에 블로그에 정리가 되어있는 자료들이 많다.
  - 직접 설치와 사용에 맞게 script문을 적용해야한다.
  - 플러그인 설정, 프로젝트 설정등 많은 작업이 필요할 수 있다.
  - 서버나 정해져있는 안드로이드 버전 개발 시 용이할 수 있다.
- power builder
  - 무료 웹 CI/CD 플랫폼
- 기타...
- Travis
  - <https://travis-ci.org/>
- MoBiL
  - <http://tech.kakao.com/2016/04/21/mobil/>
- Bitrise
  - 모바일에 최적화 된 CI 시스템
  - 웹 기반으로 무료로 사용할 수 있는 build 수가 정해져있음.

## pull request merge 완료 시 모든 내용 자동화

- web hook 사용하면 됨.
  - 제약 조건
    - 해당 프로젝트의 repository가 적절한 branch 전략을 가지고 운영해야 좋음.
      - release/develop/feature
- CI/CD
- TDD
- Release
- Notifications (mail, slack, Jira ...)
- App store 등록.



- 버전 관리 시스템 : Github
- 티켓 관리 시스템 : Slack
- slack & bitrise
  - <https://blog.bitrise.io/slack-away>

### 참고 문헌

- <https://m.blog.naver.com/PostView.nhn?blogId=cranjathe&logNo=220834321673&proxyReferer=https%3A%2F%2Fwww.google.com%2F> (Jenkins Android 빌드, 패키징, 배포)
- <http://woowabros.github.io/experience/2018/06/26/bros-cicd.html> (CI/CD 설명)
- <https://academy.realm.io/kr/posts/continuous-delivery-for-android/> (realm 설명)