

# Introduction to CTIP

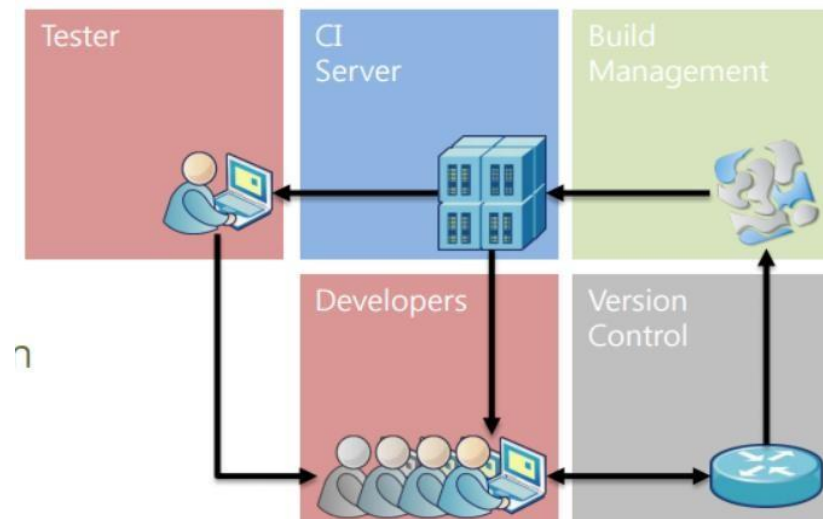
손준익

2018-03-04

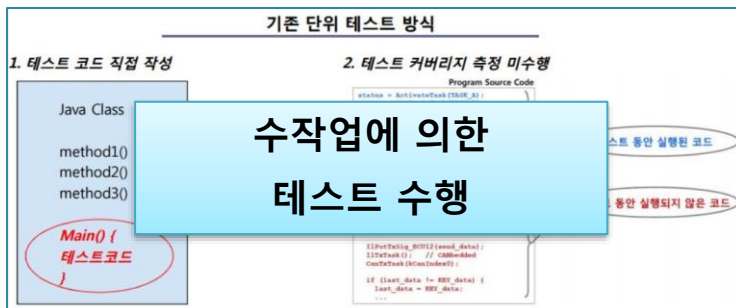
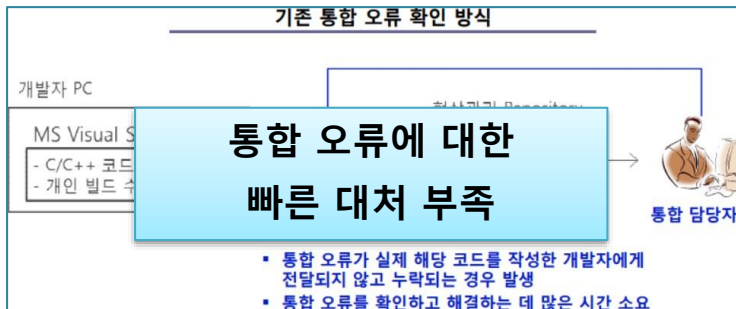
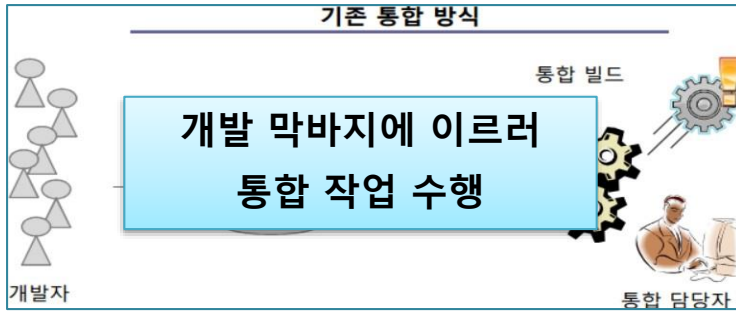
- CTIP
- CI & CTIP 추진배경
- CI
- CTIP 예제
- CTIP 장단점
- 2018 Software Verification

## Continuous Test & Integration Platform

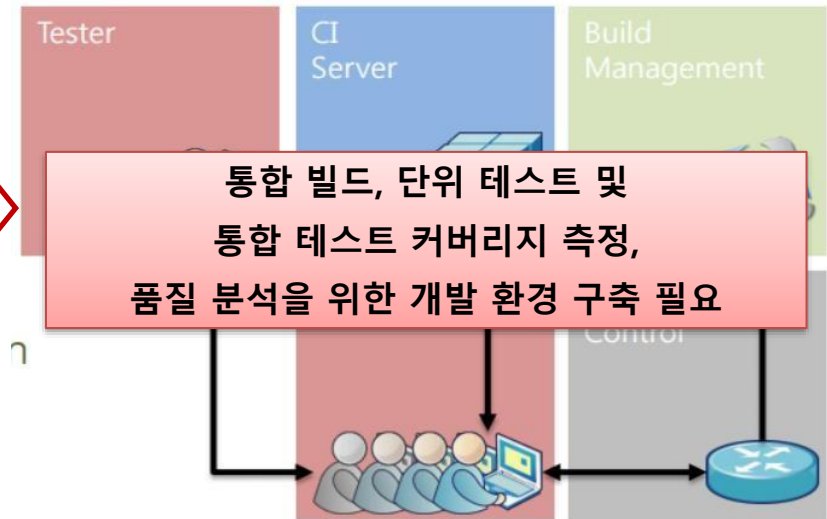
- **CI (Continuous Integration)**
  - 개발기간 동안 개별 개발된 모듈에 대한 빌드를 지속적으로 수행하기 위해 XP(eXtream Programming) 에서 도입한 지속적 빌드 자동화 개념
  
- **CTIP (Continuous Test & Integration Platform)**
  - **CI + Continuous Test**
  - 빌드 도구, 테스트 도구 및 기타 개발 시 유용한 도구들이 유기적으로 동작하도록 묶은 도구 집합(Tool Chain) 환경





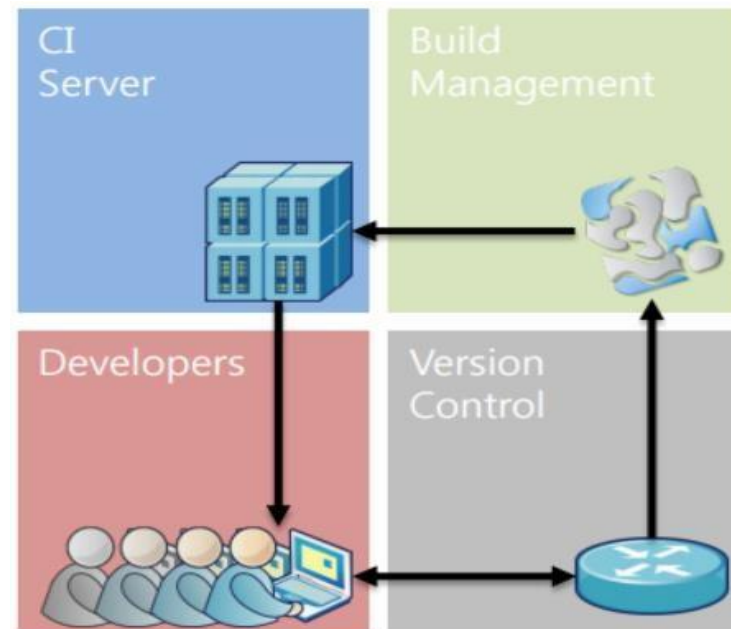


## CTIP

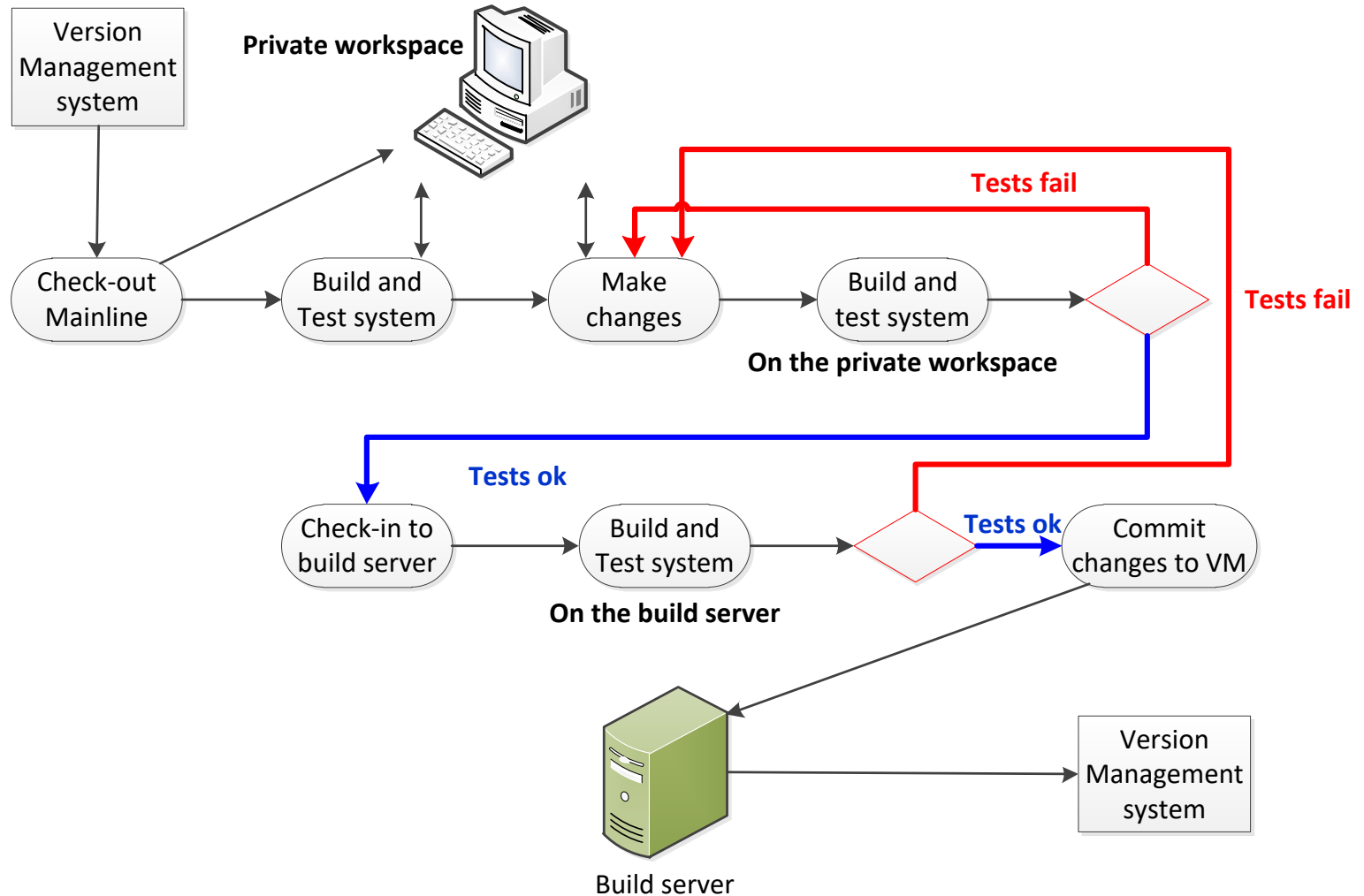


## Continuous Integration

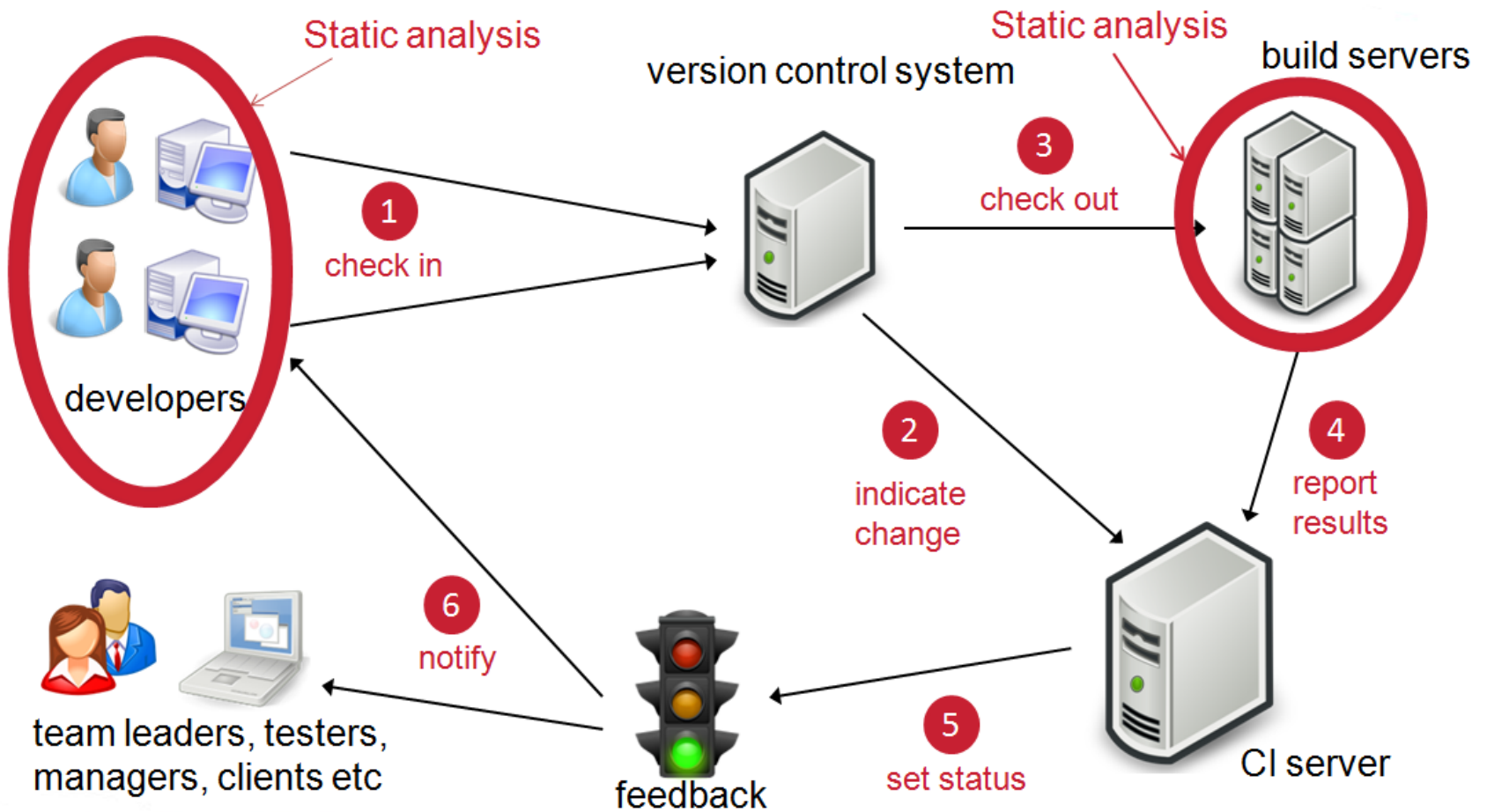
- 소프트웨어 개발에서 **Build**를 지속적으로 수행하는 것
- 지속적으로 개발된 Unit 코드에 대한 Integration 작업을 수행하여, 항상 최신의 상태를 유지
- Agile 개발 방법론/프로세스 중의 하나인 XP(eXtream Programming)의 Practice 중 하나
- 지속적인 통합(Continuous Integration)은 팀의 구성원들이 자신들의 작업한 내용을 자주 통합하는 개발 지침을 말한다. - 마틴 파울러 (Martin Fowler)



## Continuous Integration



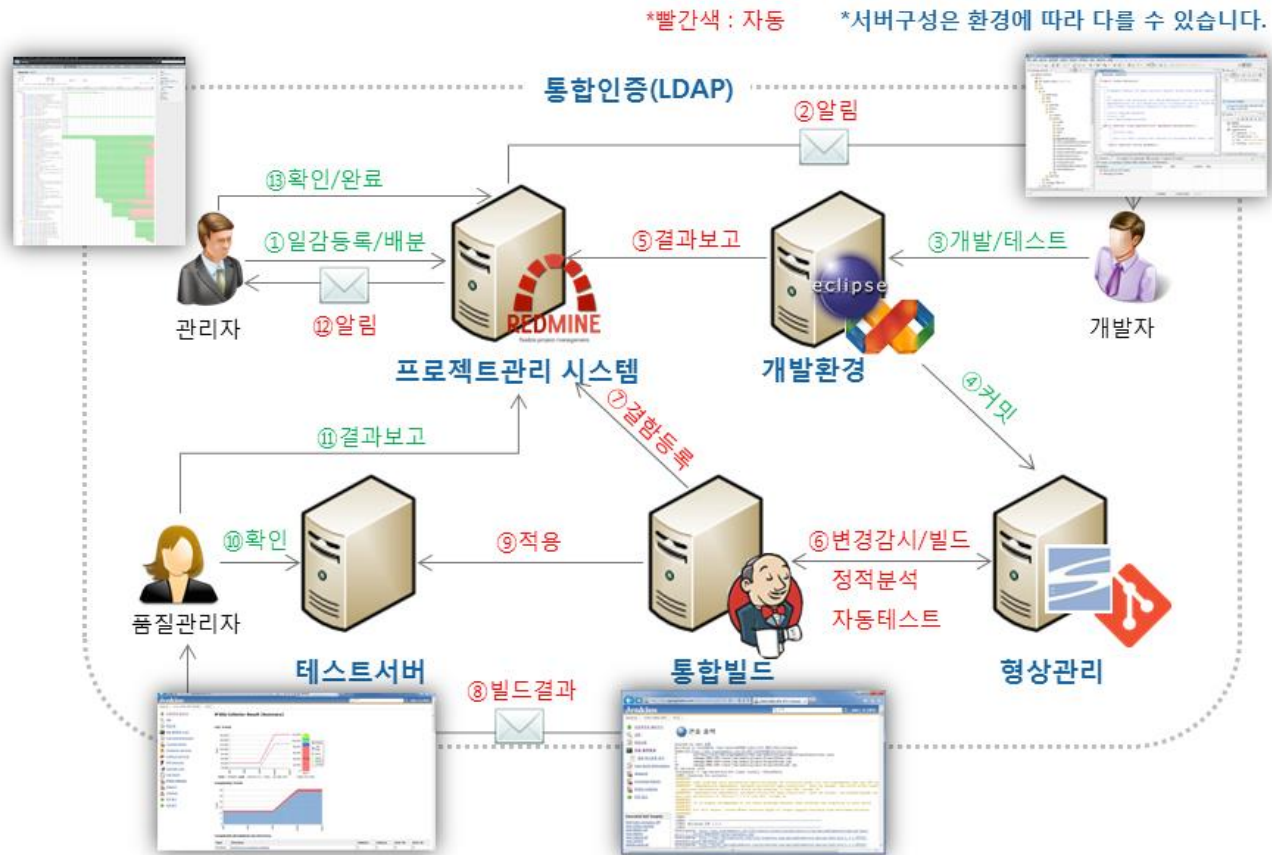
## CI Work Flow



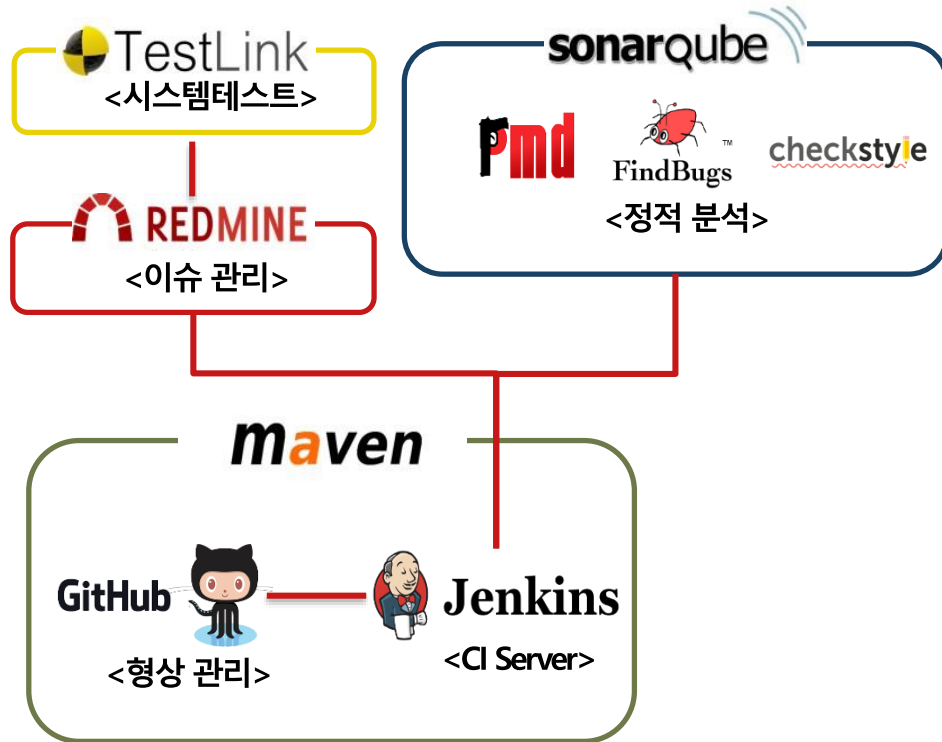
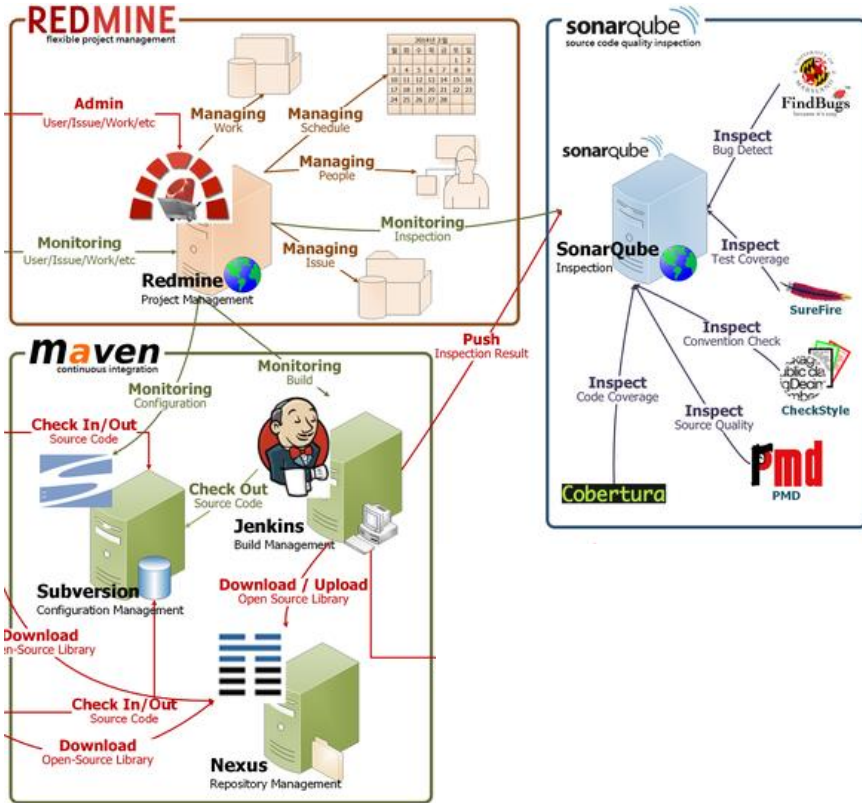


## 라이언즈 소프트 솔루션

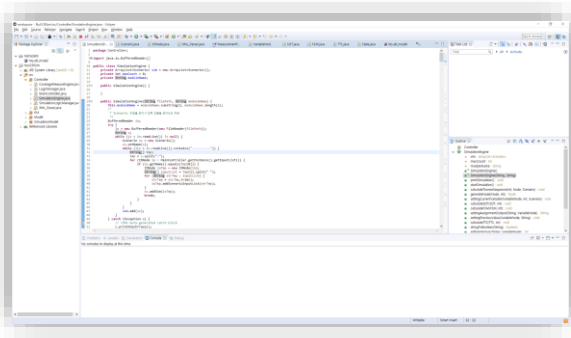
### - OPEN ALM



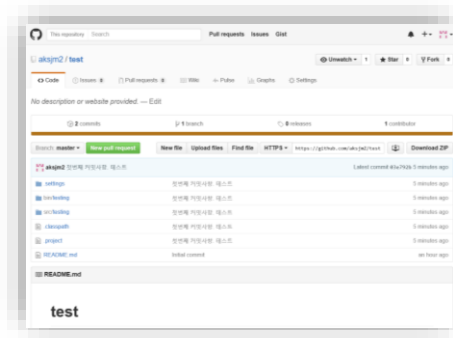
# CTIP 예제



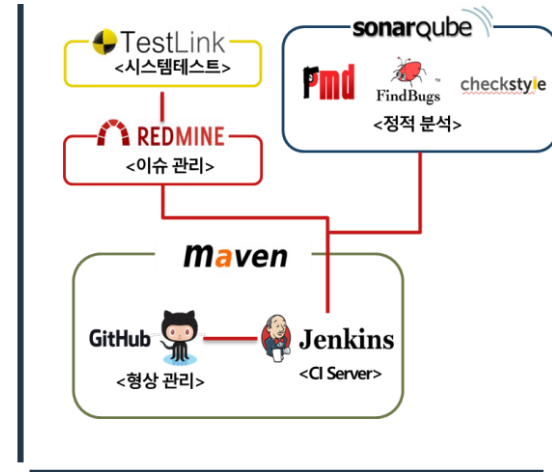
# CTIP 예제



코드 수정 후 푸시 - Eclipse



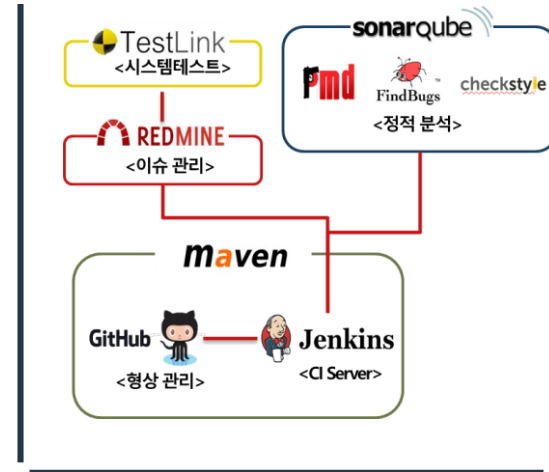
형상 관리 - Github



자동 빌드 및  
unit test 수행  
- maven

빌드 및 unit test 결과 확인 - Jenkins

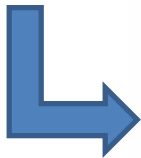
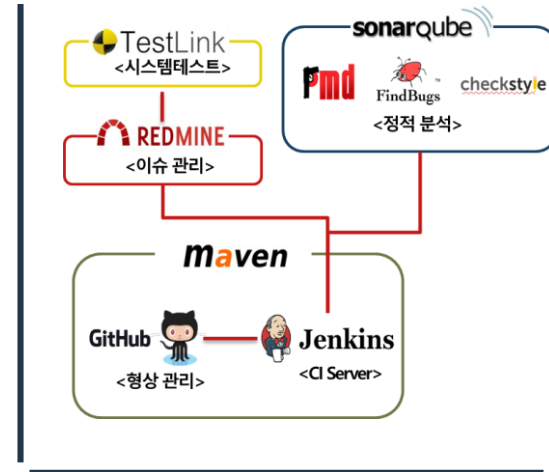
# CTIP 예제



## 자동 정적 분석 수행

- PMD, FindBugs and CheckStlye in SonarQube

## Unit Test의 Coverage 측정 - Cobertura in SonarQube




빌드 관련 결함 및 이슈 관리 - REDMINE

System Test 수행 - TestLink

- 통합 소요시간 감소
- 개발자간 코드 충돌 조기 발견
- 통합 후 오류 발생률 감소
- 지속적인 통합을 통해 문제의 조기 발견
- 수동적인 반복 작업 감소
- 빌드 결과의 배포 및 관련자에게 통보 (Feedback)
- 테스트 케이스 재사용으로 인한 개발 시간 단축
- 테스트 코드 커버리지 증가
- 품질 도구들을 통한 코드 품질 검토  
(테스트 및 정적 분석)
- 유지보수성 증가 (기능 변경, 결함 수정에 대한 공수 감소)
- 초기 환경 구축의 어려움
- Well-developed test-suite 이 필요  
(테스팅 품질 향상을 위해)

## 6 Team Presentation, 5 Team Practice

- TP#1 (03. 21) : Junit, Eclipse, 빌드환경
- TP#2 (04. 04) : Mantis, SVN, JFeature 및 CTIP
- TP#3 (04. 18) : 정적분석 도구, Testlink 및 시스템테스트 도구
- TP#4 (05. 28) : 1st System Testing
- TP#5 (06. 4) : 2nd System Testing & Static Analysis
- TP#6 (06. 11) : Final Presentation - English
  
- Team Practice #4 (05. 09) : 각 팀별로 SMA에게 CTIP 환경 전수
  
- **7개팀 (24명 – 3/3/3/3/4/4/4)**

## 3학년 "소프트웨어 모델링 및 분석" 수업과 연계

- 3학년 수업의 개발 결과물에 대해서 시스템 테스트 수행 후 결과를 Issue Tracking System (ex, Redmine, Mantis) 을 이용하여 공유합니다.
- 3학년 수업에서 단위 테스트 및 관련 분석을 수행할 수 있도록 CTIP 환경을 제공합니다.
- 테스트 결과에 대한 의견교환은 Issue Tracking System 을 사용합니다.
- 테스트를 위한 소스코드는 Configuration Management System (ex, Git, Svn)을 통해 공유합니다.
- NIPA SW 은행에서 제공하는 다양한 도구를 사용할 수 있습니다.
  
- NIPA SW : <https://www.swbank.kr/helper/tool/toolMain.do>
- 담당조교: 손준익  
E-mail: [jison.dslab@gmail.com](mailto:jison.dslab@gmail.com)  
메일 제목 양식 : [2018SV][TP#X]팀이름