



# CI/CD & Sonar Qube

- 일시 : 2020년 9월 17일(수)
- 멘토 : 삼성전자 무선사업부 SE그룹 황재용프로, 최민창 프로

## 데일리 업무

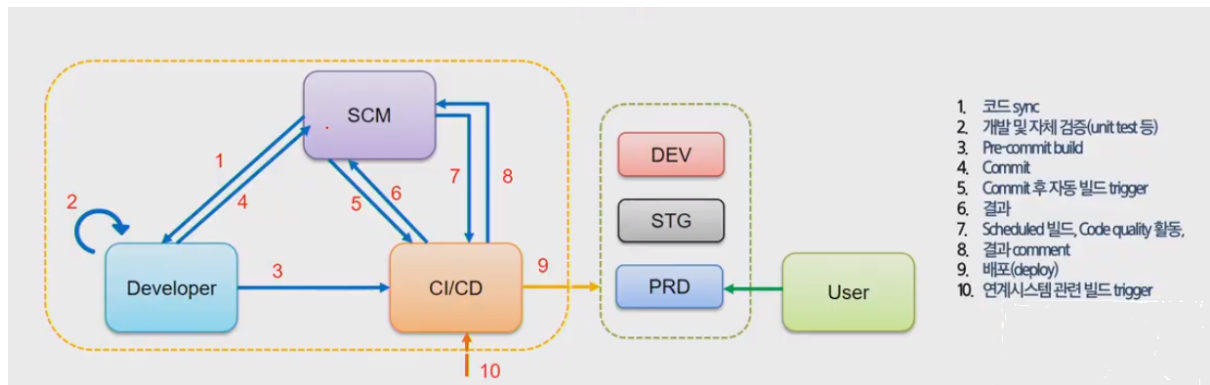
- 여러 사람들이 프로젝트를 진행하기 위한 정책, 프로세스 리딩
- 관련 툴 개발
- DevOps

## 1부

## CI/CD 활용 : 황재용프로

### 개발 파이프라인

- 개발자 : code sync, 개발, build, commit, review
- CI/CD : build, scheduled build, code quality



- Source Code Management tool(SCM)
- DEV와 STG(Stage)환경에 배포하여 테스트 수행
  - DEV - 테스트, STG - 최종 테스트

## CI/CD Tool

- Jenkins, circleci, gitlab, QuickBuild, ...

	Jenkins	QuickBuild
프로젝트 관리	각 Job에서 직접 관리	중앙 집중 Configuration(job) 관리
Agent 관리	Master에서 agent 정보 입력 후 사용	각 서버에 QB agent 설치 후 사용
확장	Plug-in	Plug-in
Plug-in Ecosystem	활발하다	활발하지 않다
Backup	Job, DB 등 간단하게 백업 가능	Configuration(job), DB 각각 백업

- 다양한 프로젝트에서 동일하게 버전업그레이드가 필요할 때 Jenkins의 경우는 각 Job마다 관리되므로 번거로움
- Agent 관리
  - Jenkins : ssh 명령어로 agent 정보 입력 후 사용 가능
  - QuickBuild : QB agent를 agent마다 설치하여야 함
- 백업
  - Jenkins : 폴더구조이므로 폴더 그대로 백업 가능
  - QuickBuild : 각각의 구조가 달라 백업방식이 달라져야 함

## CI/CD 활용 사례 및 에피소드

- 활용사례
  - 단순 업무 자동화 도구
  - 작은 규모 서비스
  - S/W Release system(대규모)
- 에피소드
  - Jenkins vs QuickBuild 사용
    - Jenkins의 경우 Job 관리자가 메일을 확인하지 못하여 버전 업그레이드를 수행하지 않는 경우가 발생하여 QuickBuild의 중앙화된 관리가 필요한 경우가 발생
  - 사용자 권한 관리 : Authentication & Authorization
    - Autentication  
신원을 확인하는 과정 ex)로그인
    - Authorization  
권한을 확인하는 과정 ex)관리자 계정인지 확인
  - 사내 proxy 환경  
proxy 환경설정을 하지 않으면 외부 네트워크에서 사내 네트워크로 데이터를 받지 못하는 경우가 발생할 수 있음

## 정리 : CI/CD

- 전반적으로 잘 사용되고 있는 Tool
- 대규모 서비스로 활용시 고려해야 할 부분이 존재
- 사내 proxy가 있을 때, 외부 접속 관련에 대해서는 항상 신경 써야 함

## SonarQube와 Code Quality : 최민창 프로

### SonarQube

코드의 품질을 관리하는 도구

- 다양한 분야에서 광범위하게 사용되고 있음

## 좋은 코드 VS 나쁜 코드

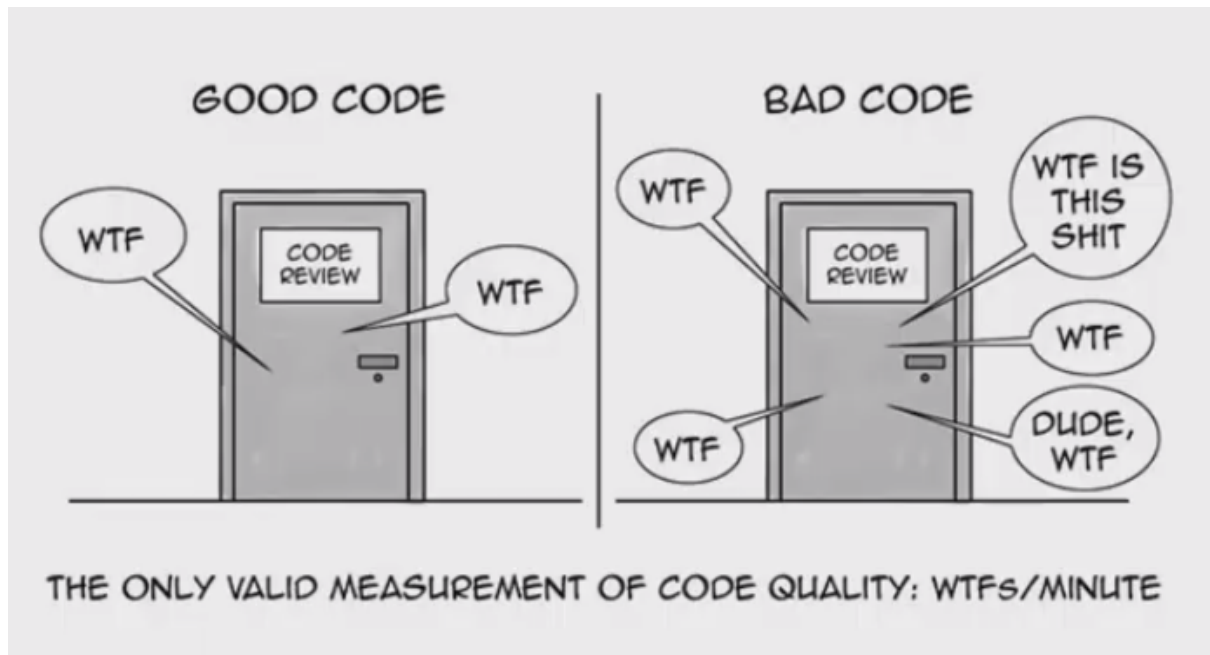
- 나쁜 코드 : 오래 지속되지 못함
- 좋은 코드 : 시간이 지나도 견고함
  - 유지보수가 용이
  - 변경 사항에 대해 기민하게 대응가능

## 좋은 코드?

1. 해야할 일을 해야한다.
2. 일관된 스타일을 유지해야 한다.
3. 이해하기 쉬워야 한다.
4. 문서화가 잘 되어있어야 한다.
5. 테스트 가능해야 한다.

## 코드 품질의 측정

욕을 적게 들수록 좋은 코드?



- ISO9126 품질 표준
  - 해야할 일을 해야한다 : Reliability(신뢰성)
  - 일관된 스타일을 유지해야 한다 : Coding Style
  - 이해하기 쉬워야 한다 : Readability(가독성)
  - 문서화가 잘 되어있어야 한다 : Documentation
  - 테스트 가능해야 한다 : Testability

## | Sonar Qube로 관리

## 코드 품질 지표

### 1. Reliability

- Unit Test
  - 어떠한 테스트 케이스들이 실패하는가? = 원하는 값을 얻지 못함
- Test Coverage
  - 테스트 수행시 해당 테스트가 포함하는 대상 범위

```

21 - int err = MPI_Init(&argc, &argv);
22 - if (err != MPI_SUCCESS) {
23     fprintf(stderr, "MPI_init failed: %d\n", err);
24 - exit(EXIT_FAILURE);
25 }
26
27 - err = MPI_Comm_rank(MPI_COMM_WORLD, &rank); /* Get id of this process */
28 if (err != MPI_SUCCESS) {
29     fprintf(stderr, "MPI_Comm_rank failed: %d", err);
30     MPI_Abort(MPI_COMM_WORLD, EXIT_FAILURE);
31     exit(EXIT_FAILURE);
32 }

```

- 초록색 : 테스트 됨
- 노란색 : 테스트가 완전하지 않음
- 빨간색 : 테스트 되지 못함
- Static Analysis(정적 분석)  
코드를 직접 실행시키지 않고 코드에 어떤 이슈가 있는지 검출. 정적 분석 도구 사용.
  - Sonar Cube는 이슈를 해결하기 위한 리소스를 시간으로 환산하여 출력

## 2. Coding Style

- 입력된 규칙에 위배된 코드를 검출

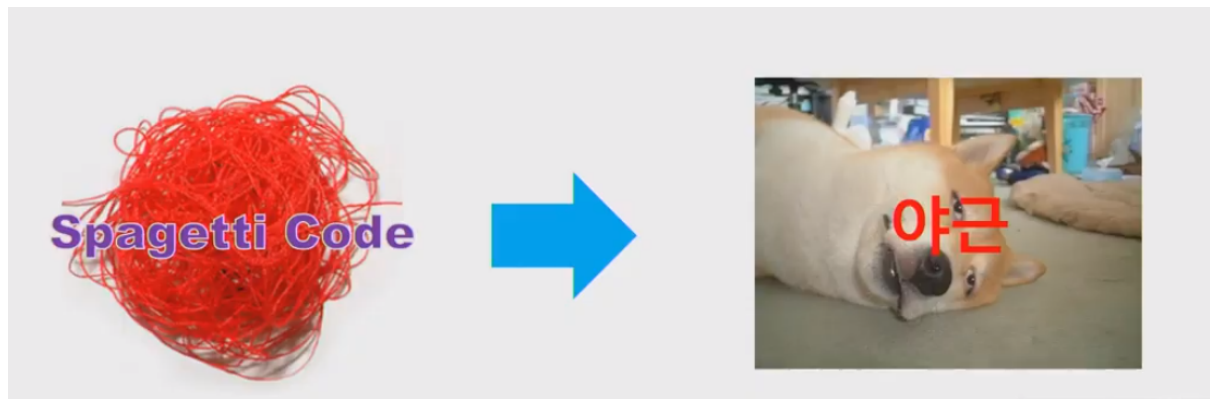
## 3. Redability

- Cyclomatic Complexity(순환 복잡도)  
작성한 코드가 어느정도의 분기를 가지고 있는가?  
⇒ 메소드나 파일내에서 순환복잡도가 20이 넘어가면 위험성 높음
- Cognitive Complexity(인지적 복잡도)  
표준적인 지표는 아니나 하위 분기가 더 높은 가중치를 가지는 것.

## 4. Testability

수치적으로 측정할 수는 없으며 Sonar Cube에서도 지원하지 않음

## 나쁜 코드의 결과



## Video

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4c696a51-a026-4578-856d-406db84eb5fc/CICDSonarCube\\_.mp4](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4c696a51-a026-4578-856d-406db84eb5fc/CICDSonarCube_.mp4)

## 2부

**문서화를 어떤식으로 작성하고 궁금합니다.**

### 무선사업부

- Confluence Wiki를 사용하여 작성(자체 구축)
- 회사 차원에서는 문서화를 독려
- SCM 툴을 사용하여 문서관리도 하는 중
- helix purpose

## 클린코드 책을 많이 참고하시는지?

- 기능구현을 우선하고 코드 리팩토링 수행
- 책을 참고하지는 않음

## 삼성 내에서 코드 작성에 관한 가이드라인이 존재하는지?

- 없음
- 코드 작성에 관하여 잘 알고 있는 아키텍터 양성에 노력하고 있음
- 이러한 인원을 기준으로 코드리뷰를 수행

## Sonar Qube에서 코딩 스타일 검사를 위한 커스터마이징 정도는?

- 자유도가 높진 않음
- 정해진 기준에서 수치조정 수준

## TDD 방법론으로 개발한다면 테스트 코드 작성하는 순서는?

- Red ⇒ Green ⇒ Refactoring
- 아무 구현내용이 없는 상태에서도 테스트 코드를 작성하기도 함

## 기간을 맞추기 위해 품질관리에 소홀해질 수 있는 부분도 발생할 수 있는데, 이럴 경우 Sonar Qube를 도입하는 게 맞을지?

- 프로젝트의 잠재적인 이슈의 총량과 새로운 기능을 도입하는 것에서 소요되는 리소스 간의 비중차를 기반으로 판단하여야 할 것으로 보임

## 테스트 코드를 작성해도 Coverage가 잘 올라가지 않은 경우가 있는데, 어떻게 하면 테스트 코드를 잘 짰다고 할 수 있을까요?



- 먼저, 테스트 커버리지가 높아야함
- 커버리지를 높이기 위해서는 테스트가 가능한 코드를 작성하는 것이 중요???????
- 100%는 거의 불가능
- 절대적인 수치는 없음. 케이스마다 다름

## **Jenkins는 배포 자동화정도로만 사용했는데, 신입들에게 어느 정도의 경험을 요구하는지?**

- 신입에게 실질적인 역할을 담당시키진 않음
- 기술 이해의 과정부터 시작
- 간단한 사용법만 알고가도 충분
- 본인들도 회사에 들어와서 툴을 사용해봄
- 소프트웨어 개발단계에서 CI/CD가 무엇인지 개념 파악정도만 하여도 충분

## **관련 툴이나 기술을 주로 어떤 방식으로 학습하고 적용하였는지?**

- 기초적인 사용부터 천천히 고도화해가며 적용하면서 학습
- 지표를 정하고 해당지표에 대해서 개선하며 이해도를 높임

## **CI/CD의 산출물을 대규모 인프라에서 어떤식으로 관리를 하는지?**

- Storage 서버에 트래픽이 몰리면 문제가 발생하므로 산출물을 관리하고 배포하는 툴을 따로 제작
- 분산된 서버에서 파일이 나누어져 있어 요청시 탐색하여 리턴시키는 툴을 작성

## **컴퓨터 공학 지식을 늘리기 위한 조언을 해주실 수 있는지?**

- OS를 하나 타겟팅하고 업데이트마다 어떠한 기능들이 추가되고 수정되었는지 확인하였음
- IT뉴스 또는 관심 사이트에서 단어를 익히는 것이 좋음

## Sonar Qube로 Vuejs의 html파일도 관리가 될지?

- Vuejs가 유명한 프레임워크다 보니 누군가가 plugin을 만들어 냈을 수도 있으나, 여러분이 직접 만드실 수도 있음

## 바라는 신입사원의 모습?

- 뭔가를 잘 알고 들어오신분 보다는, 열의가 있고 배우려는 자세를 가진 후배가 좋음
- 부정적인 모습보다는 긍정적인 자세를 가지고 열정을 보이는 동료가 좋음

## Video

### 2부

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/215e21a3-4c28-4c17-8a87-adc9d00b6a51/200917\\_.mp4](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/215e21a3-4c28-4c17-8a87-adc9d00b6a51/200917_.mp4)