4. Using Open Source SW

Hyunchan, Park

https://github.com/hyunchan-park/osscourse

Division of Computer Science and Engineering

Chonbuk National University

강의 일정

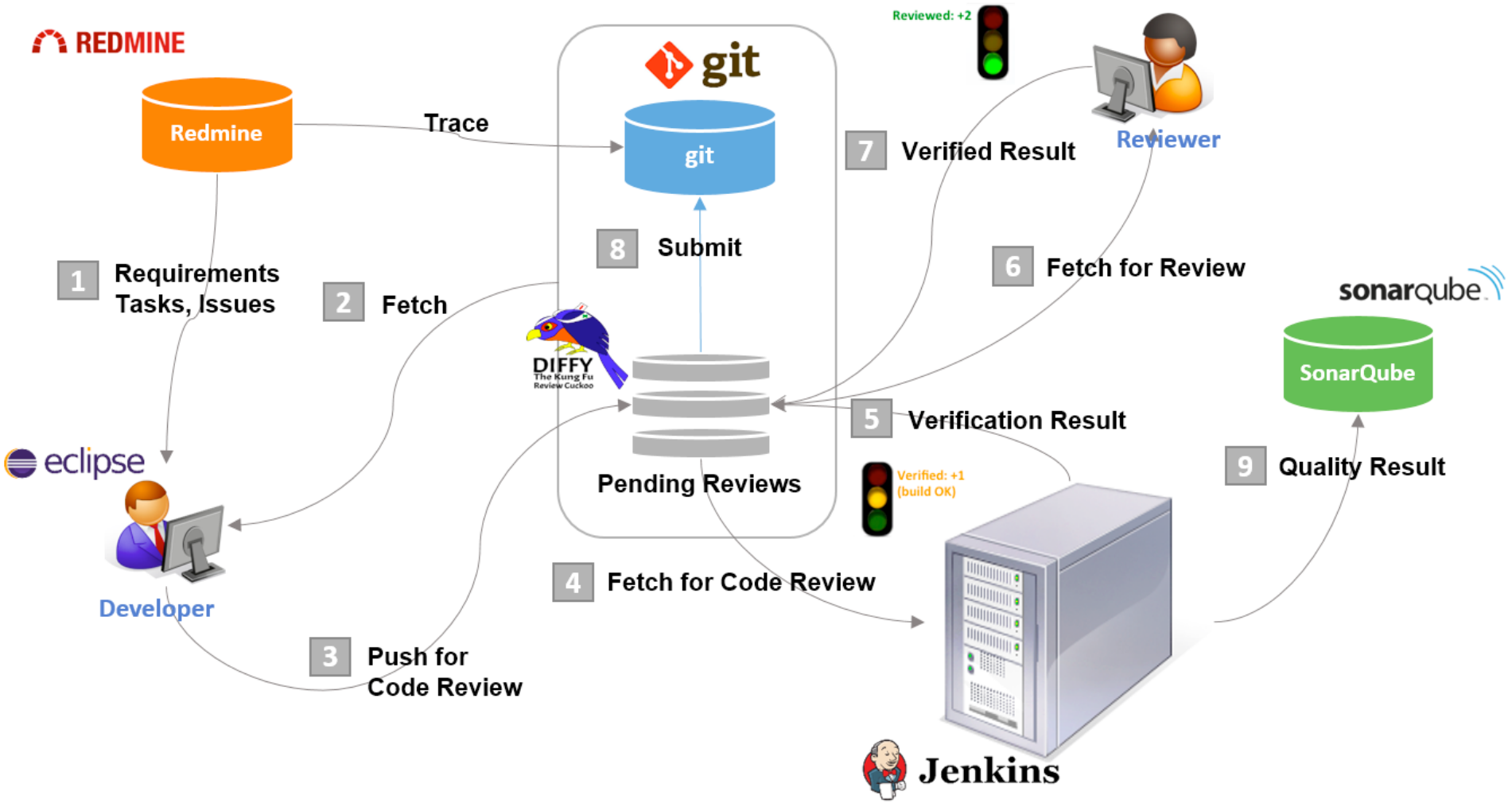
주차	월 (100분)	수 (50분)
1 (9/1)	Introduction	OSS 역사 1
2	OSS 역사 2 / OSS 개요 1	추석
3	OSS 개요 2 / OSS 라이선스 1	OSS 라이선스 2
4	OSS 활용 방법 / 버전 관리 도구 Git 1	버전 관리 도구 Git 2
5 (9/30)	버전 관리 도구 Git 3	버전 관리 도구 Git 4
6	GitHub 1	한글날
7	GitHub 2	개인 프로젝트 1: 프로젝트 선정 및 계획
8 (10/21)	중간고사	중간고사
9	코드 분석 1	코드 분석 2
10 (11/4)	코드 분석 3	코드 리뷰
11	개인 프로젝트 2: 대상 프로젝트 코드 분석 결과 발표 및 기능 구현 방향, 커뮤니티 참여 방안 발표 및 피드백	
12	클라우드 기반 개발 환경 구성 1	클라우드 기반 개발 환경 구성 2
13 (11/25)	CI/CD 도구 및 활용	문서화 도구
14	개별 프로젝트 진행	
15 (12/9)	개인 프로젝트 3: 최종 프로젝트 발표, 기능 구현 소개 및 시연, OSS 커뮤니티 활동 소개	

Agenda

- Development with OSS
- What can we do?

Development with OSS

Typical ALM with Open Source



* ALM (Application Lifecycle Management)

Typical ALM Workflow

1. 개발자는 Redmine에 작성되어 있는 요구사항, 업무, 이슈들을 이클립스의 작업 리스트에서 확인
2. 개발자는 자신의 업무와 관련 있는 소스 코드를 Git + Gerrit 으로부터 Fetch 받음
3. 개발자가 코딩과 Local Test를 마친 자신의 소스 코드를 리뷰 요청을 위해 Gerrit 에 Push 함
4. Jenkins는 빌드, 정적분석, 단위 테스트를 위해 Gerrit에 Push된 소스 코드를 Fetch 함
5. Jenkins는 확인 결과를 Gerrit에 등록
6. 리뷰어들은 개발자가 개발한 소스 코드의 리뷰를 수행
7. 리뷰어들은 검증 결과를 Gerrit에 등록
8. Gerrit은 코드 리뷰 결과와 빌드 검증 결과를 바탕으로 소스를 Git에 submit

OSS 용어

- Issue tracking
  - 프로젝트의 이슈를 생성, 해결하기까지의 과정을 추적하고 공유하는 시스템
  - 기능의 수정, 개선, 추가 기능 등 프로젝트에 대한 논의가 이슈로 진행됨
  - Issue ticketing 이라고도 하며, 이슈를 발행(ticket)한다고 표현함
  - 예) Jira, Redmine, GitHub
- Bug tracking
  - 버그의 보고, 수정 작업, 완료 보고 등 버그의 생성과 최종 해결까지의 과정을 추적하고 공유하는 시스템
  - Issue tracking 에 통합되는 경우도 많음
  - 예) Bugzilla, Redmine, Trac, Mantis
- Repository
  - 프로젝트의 소스 코드 보관소
  - 여러 사람이 동시에 작업하므로 소스 관리 (source control) 기능이 필수적. Version, branch control 이 주된 관리 기능.

OSS 용어

- Code review
  - 여러 개발자가 함께 코드를 상호 검토하여 코드의 질을 향상시키는 작업
  - Gerrit 과 같은 코드 리뷰 지원 도구가 유명함
- Quality Assurance
  - 코드 품질 관리. 일반적으로 정형화된 정적 테스트를 자동으로 수행하여 코드의 문법, 스타일, 보안 등의 오류를 탐지함
  - 예) SonarQube, PMD, FindBugs, CheckStyle, SCALe
- CI/CD
  - CI: continuous integration (지속적 통합)
    - 소스 코드의 빌드, 품질 관리를 위한 테스트를 자동으로 수행하여, QA (Quality Assurance)가 완료된 통합본을 생성하는 프로세스
  - CD: continuous delivery (지속적 배포)
    - CI 를 통과한 새로운 통합본을 자동으로 서비스 환경에 즉시 배포하는 것
  - Jenkins 가 가장 유명

OSS 용어

- Contributor
  - OSS 프로젝트에 기여 (contribution)한 사람
  - 기능 추가, 문서 작성 등 모든 형태의 기여가 가능
- Committer
  - OSS 프로젝트의 코드 관리자
  - 코드를 직접 수정하거나, contributor의 수정 요청 (PR)을 수락 or 거절
- Fork
  - 저장소를 복제하는 것. 기존 저장소에 영향을 주지 않고, 추가 수정 작업을 하기 위해 사용하는 동작.
- Pull Request (PR)
  - Fork 해서 수행한 변경 내용을 다시 기존 저장소에 적용하고자 요청하는 것

수업에서 실습할 내용

- OSS 관련
  - Git & GitHub
  - Community & Participation
- 개발 관련
  - Code analysis
  - Code review with GitHub
- 빠진 내용
  - Issue/Bug tracking
  - CI/CD, QA

What can we do?

프로젝트 진행 순서

1. OSS 프로젝트 선정
2. 개발 및 테스트 환경 구성
3. 코드 분석
4. 개발 (or others)
5. Pull request!
6. Merged! (hopefully…)

대상 프로젝트

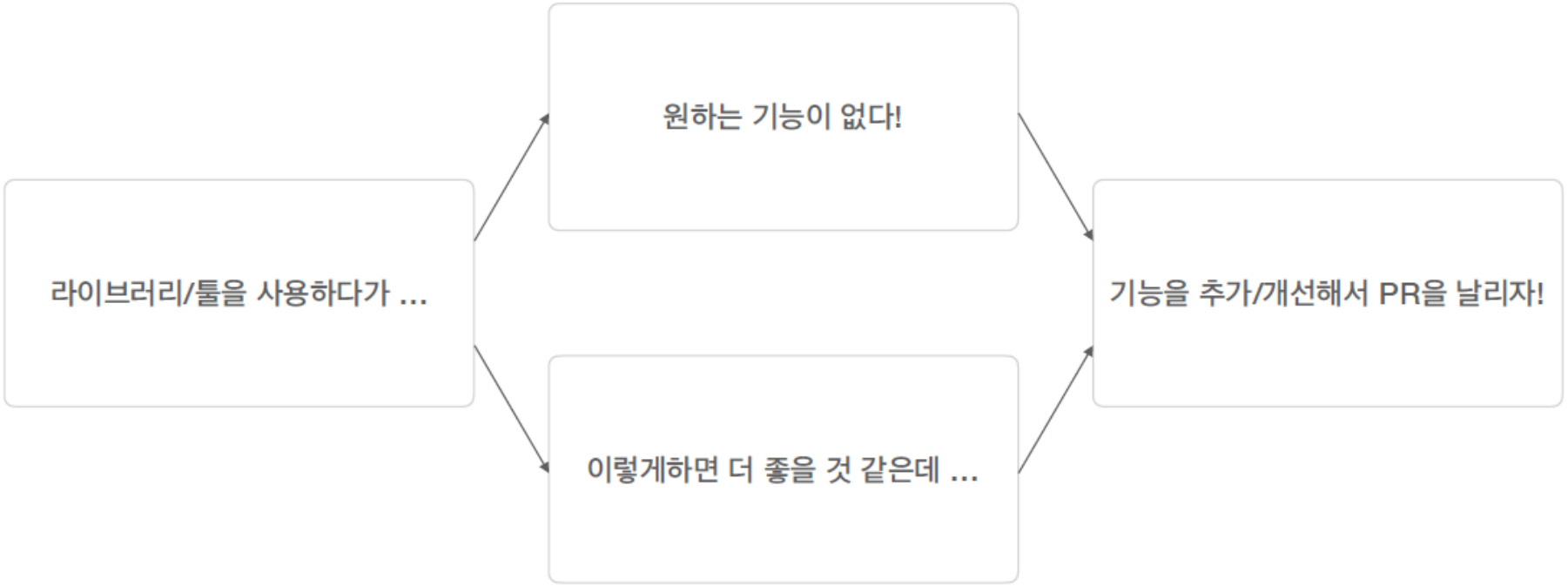
- GitHub 또는 타 Git repository 에 공개된 프로젝트
  - 약간이라도 경험이 있는 프로젝트가 좋음
  - 혹은 사용자가 많거나, 새롭고 유망한 분야의 프로젝트

- 공개되지 않은 프로젝트
  - 개별 미팅 후, 허가를 받고 진행하며, 학기 중 GitHub 공개를 전제함
  - 예) 학과 및 동아리 서비스 개발: Litmus, J-Cloud 등

What can we do for an OSS project?

기능 추가/개선: 난이도 A,B

기능 추가/개선

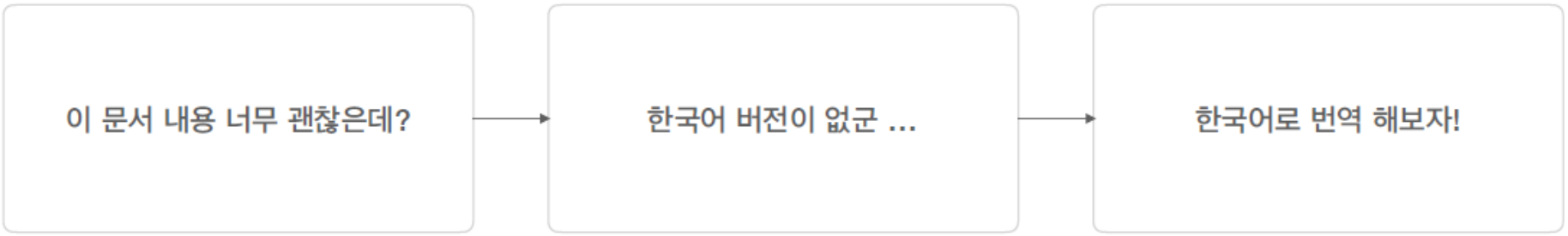


이슈 해결: 난이도 A, B

이슈 보고 및 해결

번역: 난이도 C

번역



(문서 번역의 **규칙**을 숙지하고, **이미** 진행중인지에 대한 여부도 이슈를 통해 판단할 수 있어야함)

첫 시작이 어렵다면 문서 수정/번역이나 작은 프로젝트에 대한 기여부터 시작하세요

소스코드 기여만 기여인 것은 아닙니다. 다양한 경로로 접근해보세요

프로젝트를 억지로 선택하진 마세요. 본인이 사용하고 있는 라이브러리/툴도 좋습니다

개발이 활발한 프로젝트와 커뮤니티에 참여하는게 장기적으론 좋을 것 같습니다

사실 무엇보다 중요한건 오픈소스 개발에 대한 흥미와 의지인 것 같습니다

또다른 방법: 새로운 프로젝트 구축

- GitHub 기반의 새로운 OSS 프로젝트 구축
- 소개 페이지, 기존 코드의 문서화, 이슈 정리 등
- 커뮤니티 운영 (GitHub page, messenger 등)
- Gerrit, Jenkins 를 활용한 code review, CI/CD 구축 및 운영

새로운 OSS 운영: 난이도 A

직접 운영

평소에 **하고 싶었던** 프로젝트를 시작해봅니다 (아주 쉽죠)

어떻게 보면 가장 **쉽지만** 한편으로는 가장 **어려울 수도** 있는 부분인 것 같습니다

(특히 처음이라면) **다른 개발자**들이 어떻게 **관심**을 갖게 할 것인가?

어떻게 운영하고, 어떻게 관리할 것인가?

새로운 OSS 운영: 난이도 A

직접 운영

다른 개발자들이 어떻게 **관심**을 갖게 할 것인가? 어떻게 관리할 것인가?

README

코드 퀄리티

커밋 관리

프로젝트 관리

프로젝트를 **공개** 저장소에 올리게 되면 신경써야 할 것들이 많아지게 됩니다

새로운 OSS 운영: 난이도 A

직접 운영

개인 과제 #2

- OSS 프로젝트 3개 정리 후, 보고서 작성
  - GitHub 등 검색
  - 프로젝트의 개요, 현황, SW의 동작 환경, 개발 환경 등 조사
- A4 4장 이내
  - 제목, 학번, 이름 간단히 기재
  - 워드 기본 서식 그대로 사용
  - 각 프로젝트 별 1장 분량 정리
- 기한: 10/6 (일) 23:59
  - 지각 감점: 5%p / day
  - 3주 내 제출해야 함
- 제출: LMS 시스템
  - “과제 2”