Software Verification

Team Presentation#3

201311301 이재규
201411261 김건우
201411264 김민범
201511248 김소영

Advertising Management
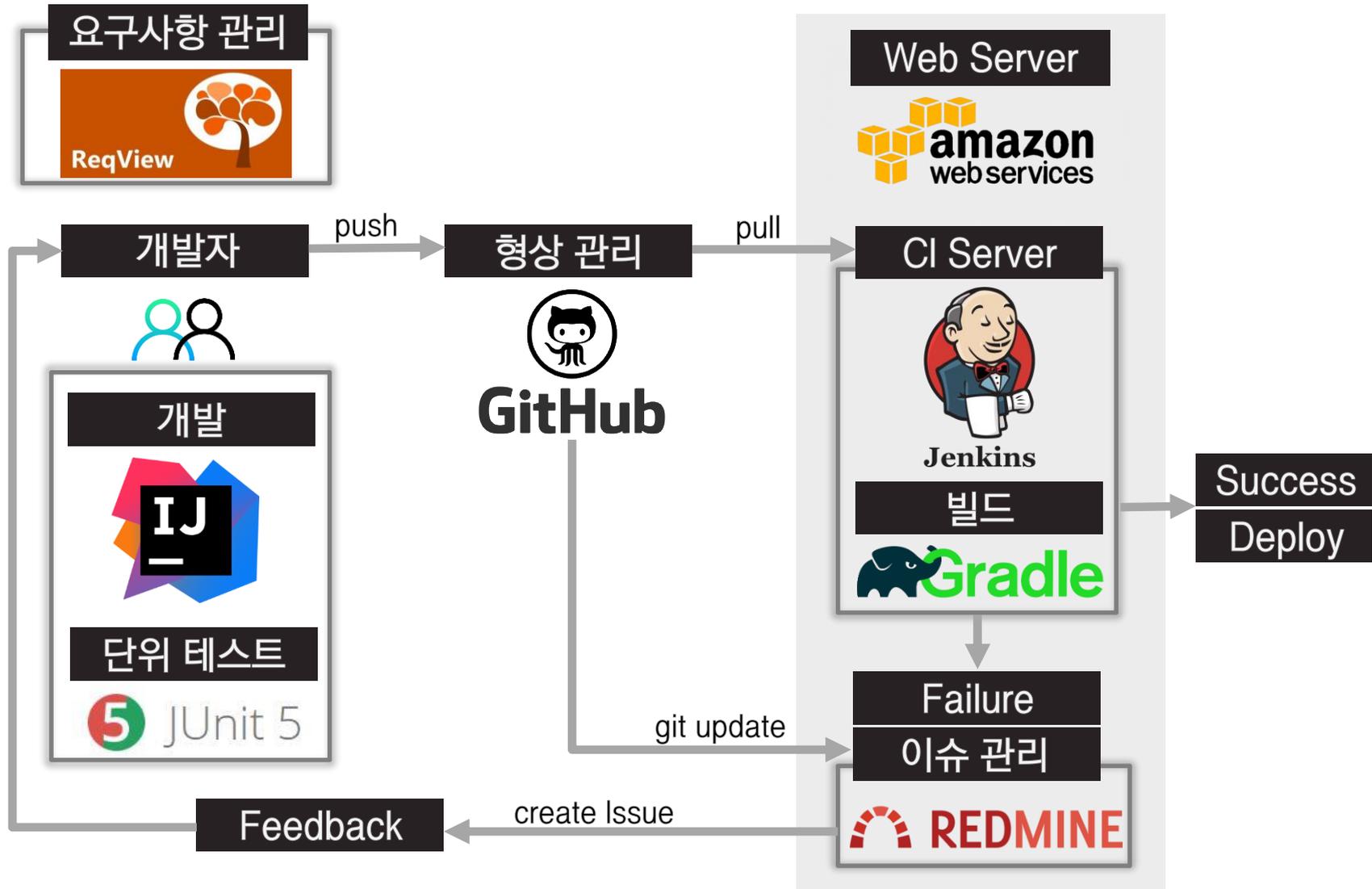
INDEX

CTIP / 2차 구축 현황

Continuous Test & Integration Platform

요구사항 관리
ReqView

Web Server
amazon web services

개발자 — push → 형상 관리 — pull → CI Server

GitHub

Jenkins

개발
IJ
단위 테스트
JUnit 5

빌드
Gradle

Success
Deploy

git update

Failure
이슈 관리
REDMINE

Feedback ← create Issue

Intro

+ **정적 분석 도구**

+ **테스트 관리**

Sonarqube

정적 분석 도구

01
02
03
04

>> 정의
실제 실행 없이 컴퓨터 소프트웨어를 분석하는 것

Might be false...

Automated Tool

>> 정의
실제 실행 없이 컴퓨터 소프트웨어를 분석하는 것
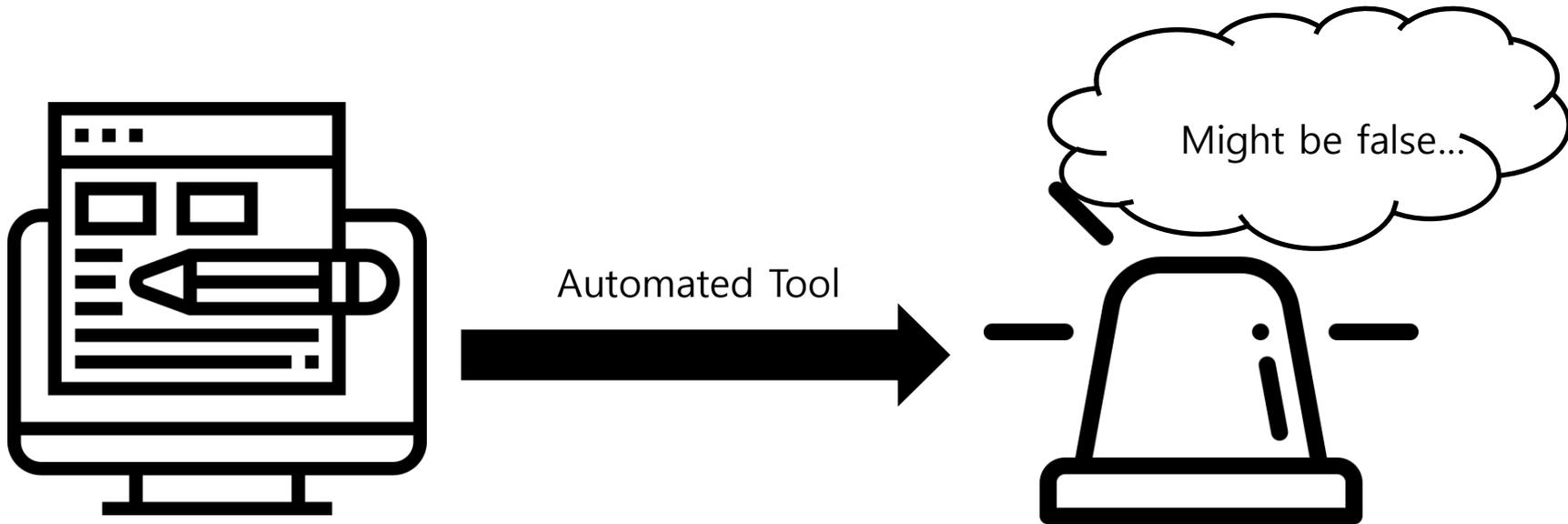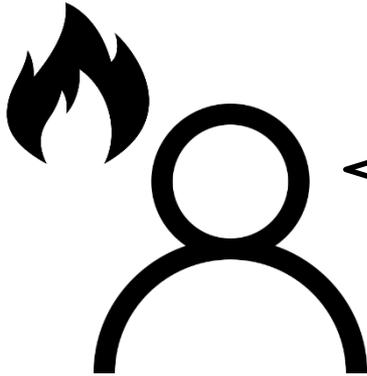
A를 입력했을 때 a가 나오나 확인하고....
B를 입력했을 때 b가 나오나 확인하고....
.
.
Z를 입력했을 때 z가 나오나 확인하고....
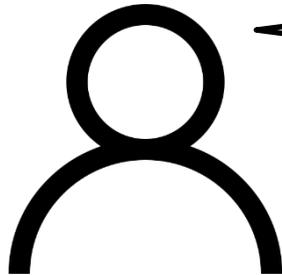
동적 분석

코드 단에서 임의의 대문자 알파벳을 입력하면
소문자가 출력되는지 자동화된 툴을 통해 분석

정적 분석

➤➤ What is SonarQube?

- 프로젝트 품질 관리를 위한 여러가지 모니터링 툴 제공 플랫폼
- continuous inspection을 통한 새로운 이슈 추적 기능 제공
- Powerful path sensitive dataflow 엔진을 통한 tricky 이슈 추적
- 25 가지 이상의 언어 제공
- Jenkins 등 CI 툴들 간의 연동

➤➤ 장점

- 오픈소스 프로젝트이므로 라이선스 비용없이 사용 가능
- 파워풀하고 심플한 Web Monitoring UI
- 테이블과 차트를 통해 직관적으로 개선 척도 확인 가능
- 품질 개선을 위한 정보를 프로젝트 단위부터 파일 단위까지 제공

sonarQube

Benefits of Using SonarQube For Code Reviews

>> Overview

Sonar Scanner

aws

>> **실행 예제**

실행 예제

›› 실행 예제

sonar-project.properties - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
# must be unique in a given SonarQube instance
sonar.projectKey=my:project
# this is the name displayed in the SonarQube UI
sonar.projectName=My project
sonar.projectVersion=1.0


sonar.host.url=http://ec2-13-125-142-52.ap-northeast-2.compute.amazonaws.com:9000


sonar.login=861496b                                        534
# Path is relative to the sonar-project.properties file. Replace "₩" by "/" on Windows.
# Since SonarQube 4.2, this property is optional if sonar.modules is set.
# If not set, SonarQube starts looking for source code from the directory containing
# the sonar-project.properties file.
sonar.sources=./src
sonar.java.binaries=target/classes
# Encoding of the source code. Default is default system encoding
#sonar.sourceEncoding=UTF-8
```

>> 실행 예제

01

02

03

04

≫ 실행 예제

01
02
03
04

⟫ 실행 예제

실행 예제

>> PMD

- 코딩에서 흔히 하는 실수들을 잡아주는 plugin (미사용 변수, 불필
  요한 객체 선언, 빈 블록, duplicated code 등)
- 위반 사항에 대한 report파일 수정이 용이함.
- 한 번의 클릭으로 여러 규칙에 대한 수정이 용이함.

>> CheckStyle

- 여러 개발자 간 다를 수 있는 coding convention을 통일
- 규칙 변경이 용이.
- 기업에서 사용하는 공개된 convention 적용도 가능.

>> Findbugs **→** Spotbugs

- Bytecode에서 버그 패턴을 찾아줌.
- 에러타입에 따라 100여개로 분류되며 위험도에 따라 rank 가능
- Jenkins plugin은 제공되지 않음

01

02

03

04

≫ 실행 예제

실행 예제

실행 예제

실행 예제

실행 예제

01

02

03

04

>> 실행 예제

TestLink

시스템 테스트

》》　다음과 같은 테스트 프로세스가 존재함

- 테스트 케이스가 어떻게 요구 사항에 mapping 되는지
- 테스트 케이스의 시나리오가 어떻게 되는지
- 테스트 케이스의 expected output은 무엇인지
- 테스트 결과가 어떻게 나오는지
- 테스트 계획은 어떻게 되는지
- ...

테스트 프로세스나 테스트 케이스 전체를 관리할 수 있는 도구가 필요

>> 한 번에 총체적으로 조직/관리하기 위한 테스트 활동의 묶음

SW 개발 단계	테스트 레벨
모듈 설계	← → 컴포넌트(단위) 테스트
아키텍쳐 설계	← → 통합 테스트
시스템 설계	← → 시스템 테스트
요구사항 분석	← → 인수 테스트

각 테스트 레벨은
SW개발 단계에 대응

각 테스트 레벨은 서로 독립적

테스트 주체,
테스트 대상,
테스트 목표,
테스트 산출물이 다름

→ **서로 다른 테스트 계획/전략 필요**

어떤 구성요소들이 **시스템 전체로서 상호작용**하는지 확인하기 위해

주변 장치들을 포함하여 전체적으로 **통합된 애플리케이션을 테스트**하는 것

Input   System Testing   Output

Who ? 독립적인 테스트 팀이

When ? 단위 테스트와 통합 테스트가 완료되어 기능상 문제가 없는 상태에

How ? 실 사용 환경과 동일한 모의 시스템을 구성하여 Blackbox 기법으로 수행

01

02

03

04

▶▶ 시스템 테스트 도구

- 오픈소스
- Web 기반의 테스트 관리
- **jenkins와 연동 가능**
- 다양한 버그 트래킹 시스템과 연동 가능 ➔ **redmine과 연동 가능**
- 테스트의 우선순위 부여 기능 및 사용자의 활동영역 정의
- 트리 구조를 이용한 테스트 케이스 작성 기능
- 다양한 형식의 보고서 제공 및 E-mail 보고서 전송 기능

TestLink 용어

Test Project

테스트를 수행하는 프로젝트 자체

테스트 팀 단위로 분할하는 것이 관리에 용이

Test Specification

테스트를 진행하고자 하는 Test Case들의 집합

Test Suite과 Test Case로 분류

Test Suite

테스트의 대상이 되는 컴포넌트나 시스템에

사용되는 여러 Test Case들의 집합

Test Case

Test Specification의 소분류

버전 관리 가능

Test Plan
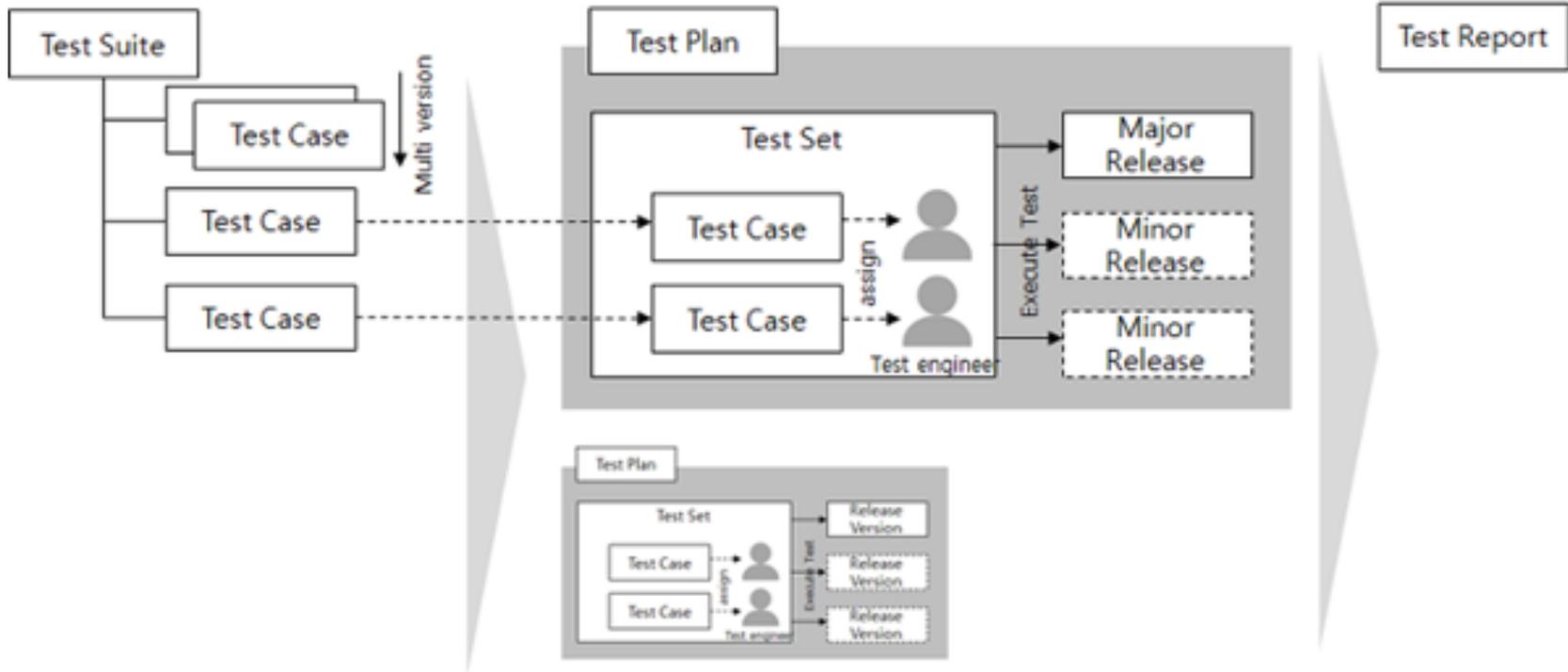
실제로 진행하는 테스트

Test Suite/Test Case를 모아놓은 것

Test Execution

Test Plan에 따라서 할당된 테스트를 수행

테스트 결과(Pass/Fail) 여부 체크

Test Report
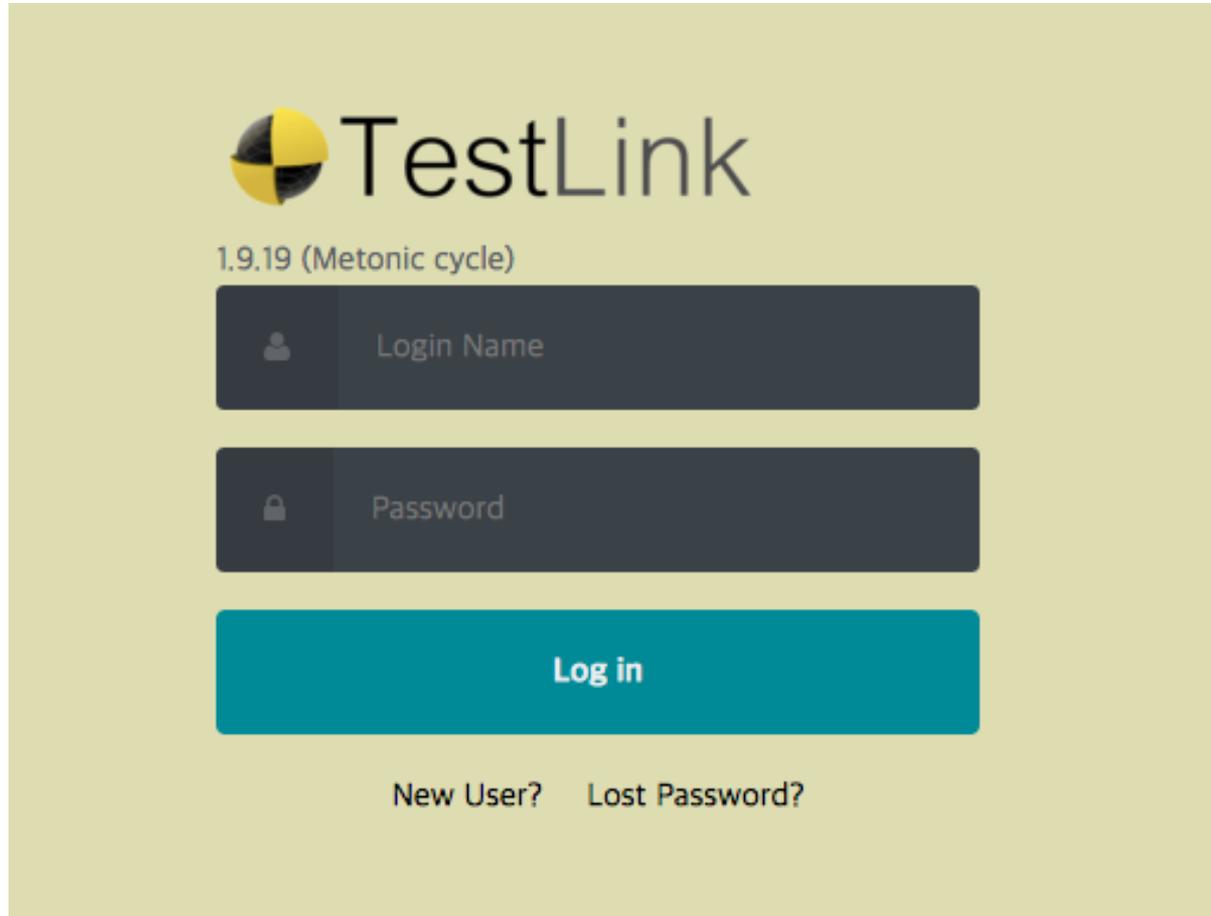
테스트 결과 리포팅

테스트 성공, 실패 여부 리포팅

≫ TestLink의 테스트 프로세스

조대협 (http://bcho.tistory.com)

01

02

03

04

➤➤ 시작 화면

01

03

04

➤➤ 프로젝트 생성

>> 생성된 프로젝트에서 사용할 수 있는 기능

01

02

03

04

>> Test Specification에서 Test Suite/Test Case 생성

TestLink 실행

TestLink

01

02

03

04

≫ Test Specification에서 Test Suite 생성

36 / 52
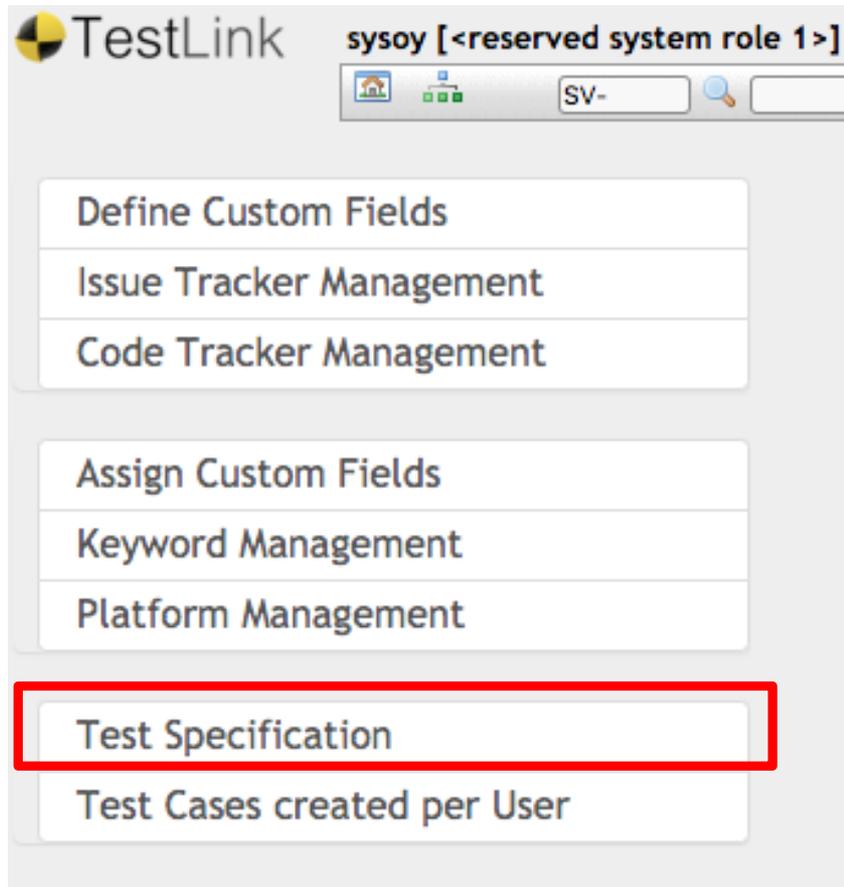
>> Test Specification에서 Test Suite 생성

01

02

03

04

➤➤ Test Specification에서 Test Suite 생성 확인

▶▶ Test Suite에서 Test Case 생성

TestLink 실행

》》 Test Suite에서 Test Case 생성

01
02
03
04

>> Test Specification에서 Test Case 생성 확인

≫ Test Case 에서 실제 Test Step 생성

≫ Test Case 생성 확인

Test Case

SV-1 : TestCase 생성 테스트 - Version 1

Summary

어떤 내용을 테스트하는지 기술

Preconditions

테스트 전제조건 입력

#	Step actions	Expected Results	Execution		
1	Create Step 버튼을 누른다.	Step 생성 화면이 나온다.	Manual	⊗	⊕
2	Step actions 에 내용을 입력한 뒤 저장한다.	Step 에 내용이 반영된다.	Manual	⊗	⊕

01

02

03

04

➤➤ Test Plan 생성

Test Project [SV:Software_Verification ⬍]

Test Plan Management

Test Plan Management - Test Project Software_Verification

| Name | Test Plan 테스트 생성 |

| Description | ◇ Source ⤺ ⤻ ☰ ☰ |

Normal ▾ **B** *I* U S

Test Plan 테스트 생성

body p

| Active | ☑ |

| Public | ☑ |

Create Cancel

⟫ Test Plan Builds/Releases

Current Test Plan: ❓

Test Plan 테스트 생성 ▼   OK

Test Plan Management

Builds / Releases

Milestones

Access Execute Tests RO

Test Cases Assigned to Me

Test Reports and Metrics

Metrics Dashboard

Add / Remove Platforms

Add / Remove Test Cases

Assign Test Case Execution

Set Urgent Tests

Update Linked Test Case Versions

Show Test Cases Newest Versions

Build management - Test Plan : Test Plan 테스트 생성

Create a new Build

Title	version 1.0
Description	Source ⟳ ⟲ ☰ ☰
	Normal ▼ **B** *I* U S̶
	첫번째 builds/release
	body p
Active	☑
Open	☑
Release date	02/05/2019 📅 ❌

›› Test Plan에 Test Case mapping

01

03

04

>> Test Plan에 Test Case mapping

01

02

03

04

>> Test Engineer한테 Test Case 할당

>> 그 이후 Test Execution, Report 기능을 사용

TestLink는 **테스트 팀 관리**와 **테스트 케이스 관리** 기능을 제공해

시스템 테스트를 편리하게 해주는 도구이지만,

테스트에 필요한 코드 작성을 지원하지 않기 때문에

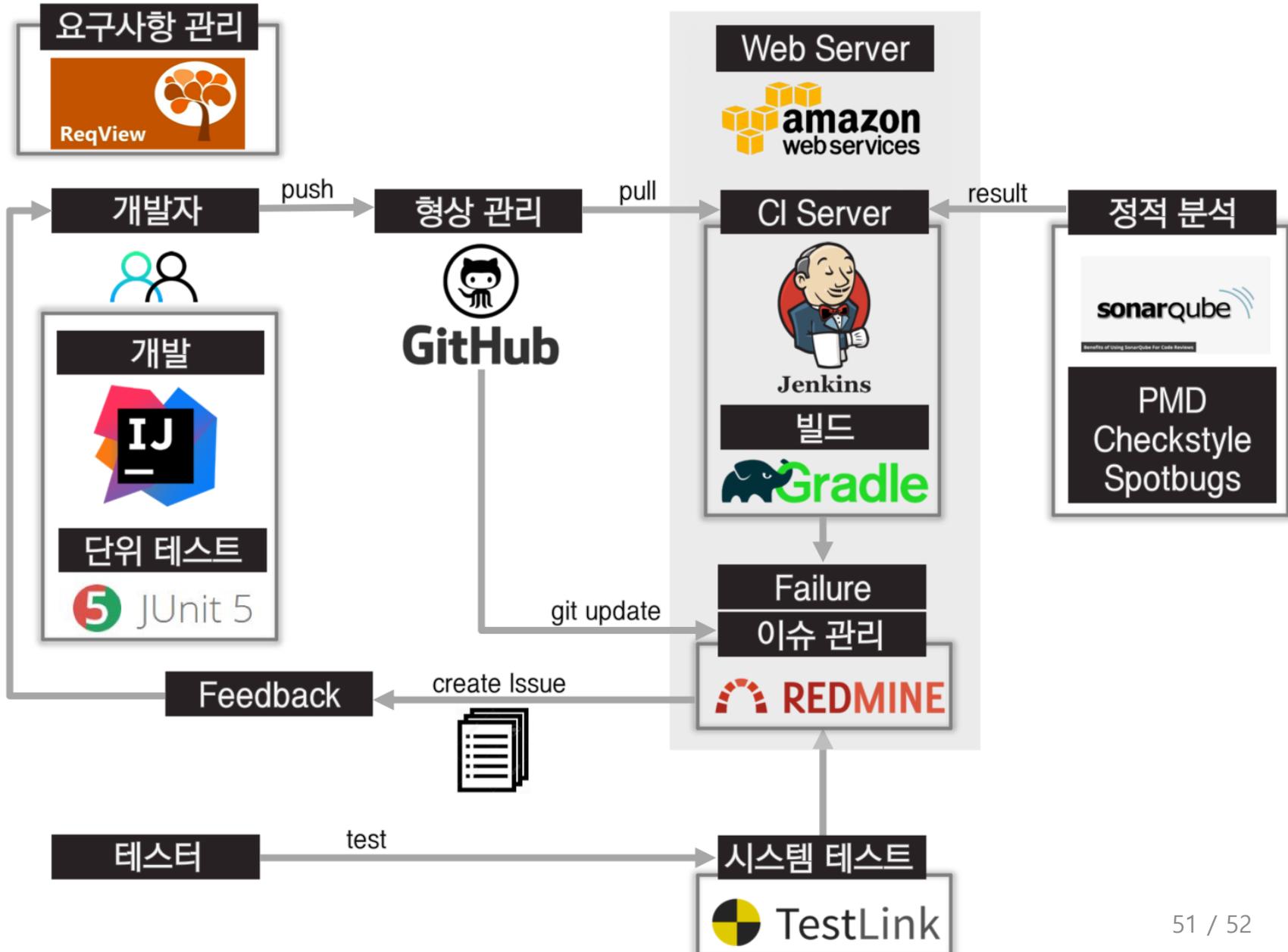✓ **어떻게 테스트 팀을 구성**하고

✓ **어떻게 테스트 프로세스를 정의**하는지에 따라

테스트 품질이 결정된다.

Summary

THANK YOU