

V&V프로젝트 발표 3

CTIP 환경 구축(2)

발표자: 이규은

공민정, 이규은, 김태형, 최지현

- 1 정적 분석 도구
- 2 시스템 테스트 도구
- 3 Overall

INDEX

01

정적 분석이란?

1. 개념

- 실제 실행 없이 소스코드의 의미 분석-> 컴퓨터 소프트웨어의 결함을 찾아내는 분석
- 자동화된 툴을 사용한 분석하거나 코드 검토
- 소프트웨어의 코드 보안 약점을 점검 -> 완성된 소프트웨어의 발생 가능한 취약점 예방

⇒ 실행되지 않거나 규칙에 맞지 않는 코드 검출

2. 특징




- 소프트웨어 실행할 필요 없음
- 실행 과정이나 결과보다는 실행 전 구현에 초점
- 개발초기에 보안 약점 파악 가능 -> 수정비용 절감 가능
- 컴포넌트 간 발생할 수 있는 통합된 보안 약점 발견은 제한적
- 설계, 구조 관점의 보안 약점은 발견 불가
- 동적 분석기법과 상호 보완적 기능 수행





01

정적 분석 도구 - SonarQube

- SonarQube의 주요 기능
 1. 복잡도 확인
 2. 코드 중복 검출
 3. 잠재적 버그 검출
 4. 단위 테스트
 5. 커버리지
- SonarQube의 특징
 1. 웹 서버, 검색 서버, 연산 서버로 구성됨
 2. DB는 SonarQube 자체 설정과 각 프로젝트의 정보가 저장됨
 3. 플러그인 추가 가능
 4. SonarQube scanner로 코드 분석



-  **Multi-Language**
27 Programming Languages
-  **Detect Tricky Issues**
Powerful code analyzers
-  **Security Analysis**
Code Security, for everyone

-  **Enhance Your Workflow**
CI/CD integration, PR decoration
-  **Built-in Methodology**
Focus on New Code
-  **Release Quality Code**
Clear go/no-go indicators
-  **Elevate Your Game**
Sharpen your developer skills



PMD

- 미 사용 변수, 비어 있는 코드 블록, 불필요한 오브젝트 생성과 같은 Defect를 유발할 수 있는 코드 검사
- 중복된 코드를 검출
- 다양한 언어 지원 (Java, C/C++, C#, PHP, Ruby...)
- 단점: 복제된 코드를 찾는 속도 느림

<PMD 점검 기준>

표준 코드 기준	<ul style="list-style-type: none"> - 명명 규칙: 클래스, 메소드, 파라미터, 변수 이름 - 클래스 및 메소드 길이 - 주석 및 JAVADocs 존재 및 서식
코드 안티 패턴	<ul style="list-style-type: none"> - 비어 있는 try/catch/finally/switch 블록 - 사용되지 않은 지역 변수, 파라미터, private 메소드 - 너무 복잡한 표현(불필요한 if 문, 무한 루프가능한 for문) - 높은 복잡성 지표를 가진 클래스
CPD(Cut and Paste Detector)	<ul style="list-style-type: none"> - 의심스러운 코드 복사를 찾는 도구 - 최소 크기의 코드 블록에 의해 매개변수화 가능



CheckStyle

- 개발된 코드가 얼마나 코딩 룰을 잘 따르고 있는지 분석
- 목적: 자바 소스파일을 읽어서 소스 코드 표준에 위반되는 것들 검색
- 기본적으로 내장된 코딩 룰: Google's style, Unix Style
- 장점
 - 조직만의 코딩 룰을 정의하고 작성된 코드들이 이를 따르는지 점검 가능
 - => 작업 시, 여러 사람과 코딩 스타일 맞출 수 있음.
 - 개발 초기 잠재적 결함 발견 가능
- 단점: 실제 버그를 찾을 수 없음



FindBugs

- 대표적인 Java 정적 분석 도구
- 목적: 잠재적 버그 찾기(자바 소스 파일이 아닌 바이트 코드 이용)
- Java 프로그램에서 100여개의 잠재적인 에러 타입 찾음.
- 잠재적 에러: scariest, scary, troubling, concern 구분해서 점수(rank)가 주어짐
- 장점
 - 실제 결함을 잘 찾음.
 - 정확성이 높음
 - 바이트 코드 읽음 -> 속도 빠름
- 단점: 컴파일된 클래스 파일에서 바이트 코드를 읽어 사용함-> 빌드 과정 필수

01

정적 분석 도구 - 설치

- Sonarqube 설치 과정

```
VnV_admin@raspberrypi:~ $ wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.3.1.34397.zip
```

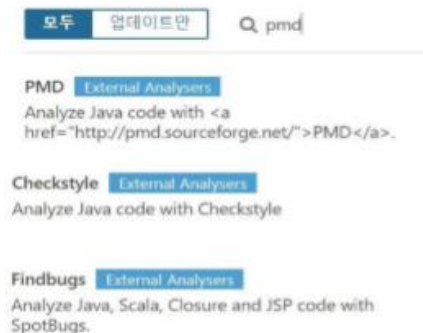
Unzip후 원하는 디렉토리로 이동 → Database와 사용자 생성 (postgresql로 접속)

sonarqube/conf/sonar.properties에서 항목 수정 → 실행 및 서비스 등록!

```
# User credentials.
# Permissions to create tables, indices
# The schema must be created first.
#sonar.jdbc.username=sonar
#sonar.jdbc.password=sonar
```

```
sonar.web.javaAdditionalOpts=-server
# Binding IP address. For servers with multiple
# address will be used for listening on the
# By default, ports will be used on all IP
#sonar.web.host=0.0.0.0
# Web context. When set, it must start with
# The default value is root context (empty)
#sonar.web.context=
# TCP port for incoming HTTP connections.
sonar.web.port=9001
```

- PMD,CS,Findbugs: Sonarqube의 Marketplace에서 플러그인 설치



* 시스템 테스트란?

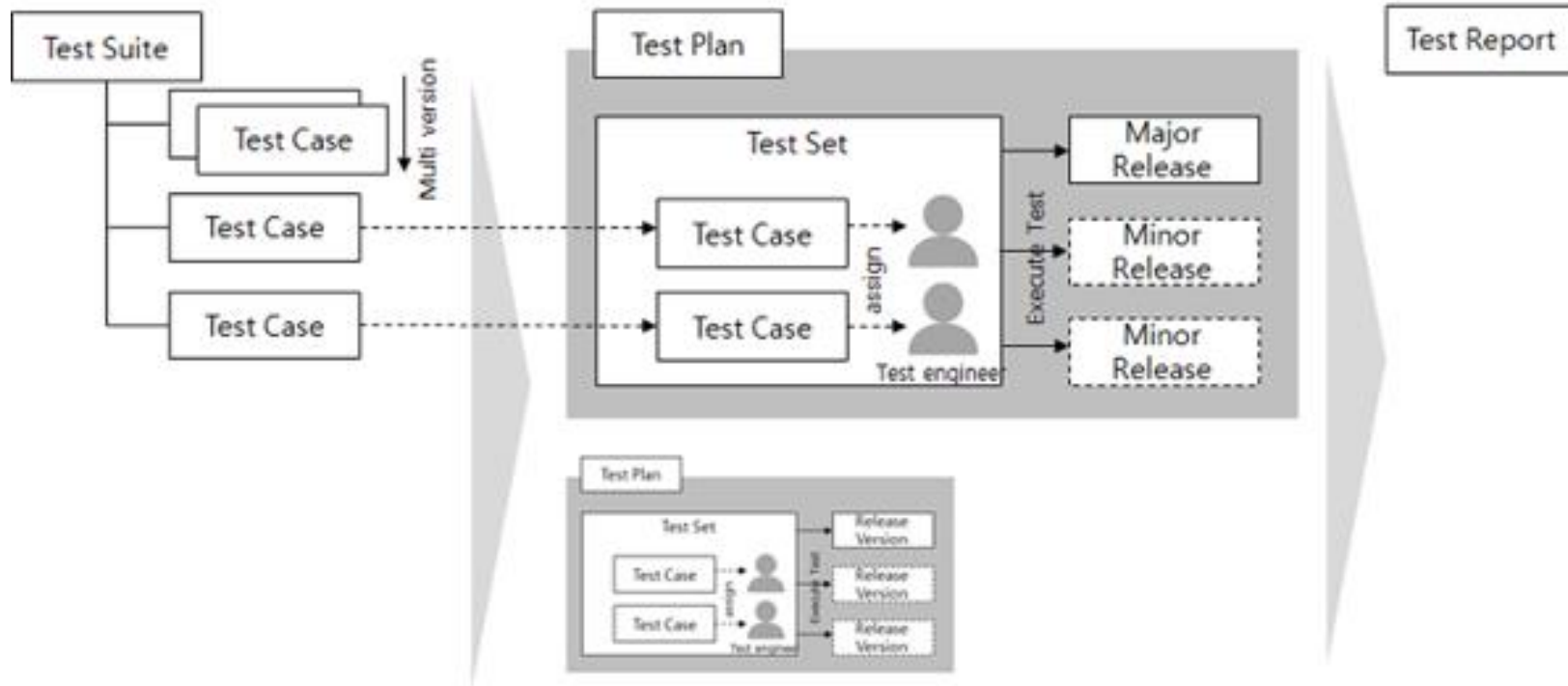
- 통합한 모듈들이 요구사항에 잘 맞게 작동하는지 판단
- 개발 프로젝트 차원(범위)에서 정의된 전체 시스템 또는 제품의 동작에 대해 테스트하는 활동
- 목적: 요구사항 확인, 시스템 연동 확인, 성능 확인, 처리량 확인, 보안성 확인
- Black Box 레벨에서 주로 수행 → Black Box 테스트 분류



TestLink

기능

- Web 환경의 인터페이스-> 사용자는 설치없이 어디서든지 접근 가능
- 테스트의 우선순위 부여 가능, 활동영역 정의(리더, 테스터)
- 다양한 종류의 DB 지원(MySQL, PostgreSQL, MS-SQL...)
- 트리 구조 이용한 테스트케이스 작성 가능
- 다양한 언어 지원
- 다양한 버그 시스템들과 직접 연동 가능
- 다양한 형식의 보고서 제공, Email에 직접 보고서 보낼 수 있는 기능



TestLink 개념



- Test Project : 테스트를 수행하는 프로젝트 그 자체
- Test Specification: 테스트를 진행하고자 하는 테스트 케이스들의 집합
 - Test Suite: 대분류의 테스트 케이스
 - Test Case: 소분류의 테스트 케이스
 - Test Case 에 대한 버전 관리 가능
- Test Plan: 실제 진행하는 테스트
 - 테스트 대상 시스템의 특정 버전을 테스트 하기 위해 Test Spec 내 Test Suite/Case 모아 놓은 것
- Test Execution: 테스트 결과의 Pass/Fail 여부 체크
- Test Report: 테스트 실패, 성공 여부, 주요 Test Category별 성공/실패 여부 등 리포트

02

시스템 테스트 도구: TestLink 설치

```
ubuntu@ip-172-31-42-86:~$ sudo apt-get install apache2
ubuntu@ip-172-31-42-86:~$ sudo apt-get install php7.0 php7.0-mysql php7.0-gd php7.0-ldap php7.0-json php7.0-curl php7.0-mbstring
ubuntu@ip-172-31-42-86:~$ sudo apt-get install mysql-server
ubuntu@ip-172-31-42-86:~$ cd /var/www/html
ubuntu@ip-172-31-42-86:/var/www/html$ sudo wget https://sourceforge.net/projects/testlink/files/TestLink%201.9/TestLink%201.9.16/testlink-1.9.16.tar.gz
ubuntu@ip-172-31-42-86:/var/www/html$ sudo tar -xvzf testlink-1.9.16.tar.gz
ubuntu@ip-172-31-42-86:/var/www/html$ cd mv testlink-1.9.16 testlink
ubuntu@ip-172-31-42-86:/var/www/html$ ls
index.html  phpinfo.php  testlink  testlink-1.9.16.tar.gz
ubuntu@ip-172-31-42-86:/var/www/html$ sudo chmod 777 /var/testlink/ -R
ubuntu@ip-172-31-42-86:/var/www/html$ cd ../..
ubuntu@ip-172-31-42-86:/var$ sudo mkdir testlink/logs
ubuntu@ip-172-31-42-86:/var$ sudo mkdir testlink/upload_area
```

```
ubuntu@ip-172-31-42-86:/var$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

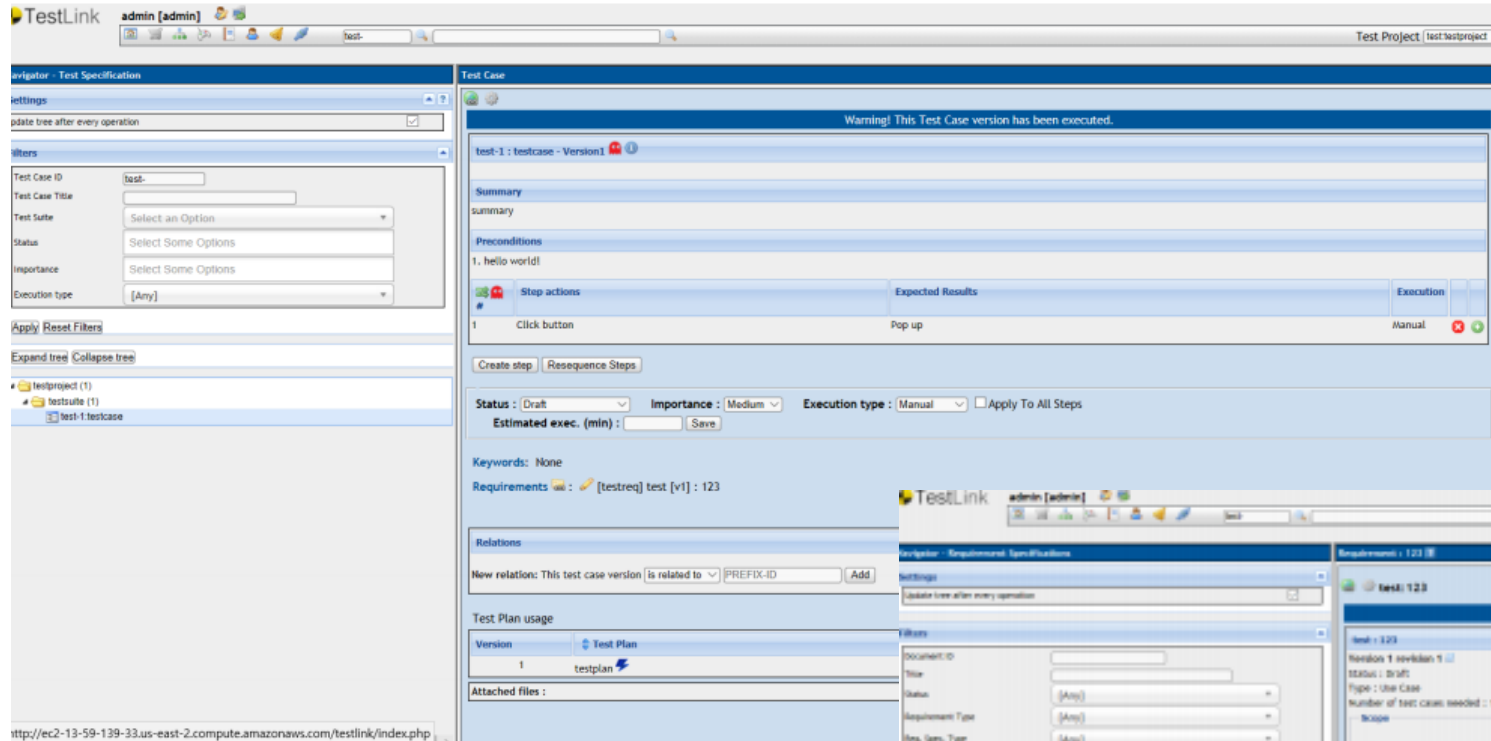
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database testlink;
Query OK, 1 row affected (0.00 sec)

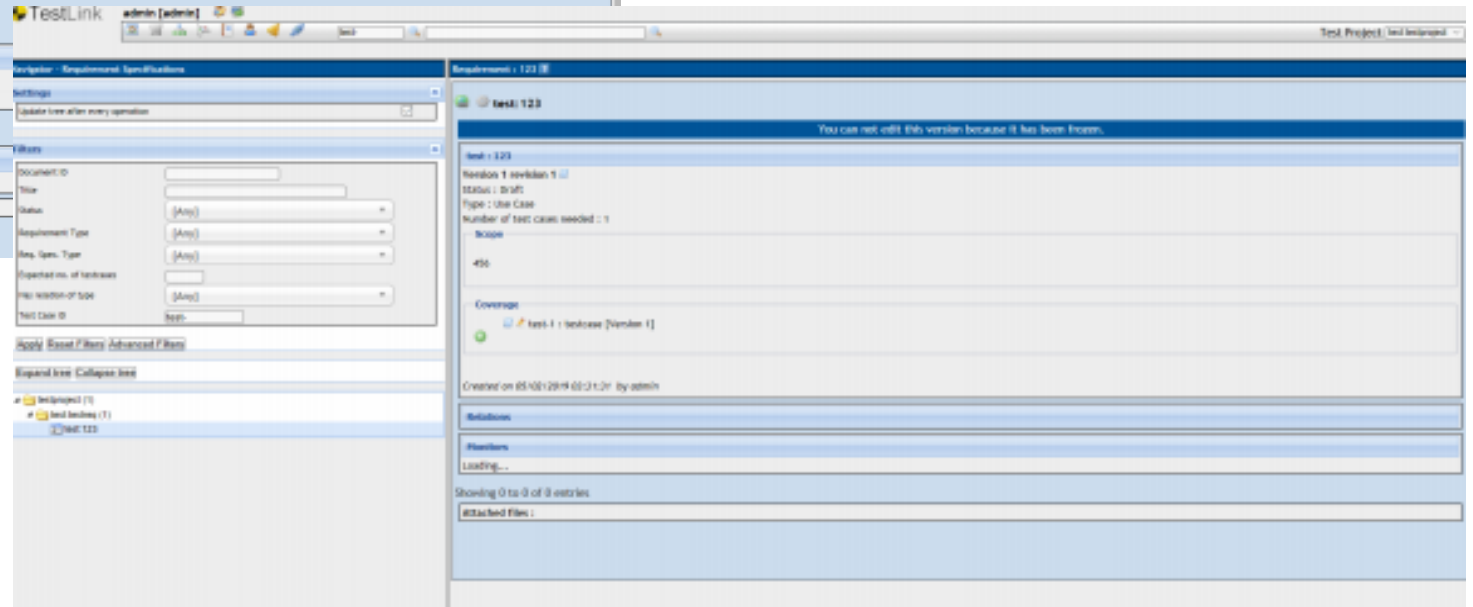
mysql> quit
Bye
ubuntu@ip-172-31-42-86:/var$ cd /var/www/html/testlink
ubuntu@ip-172-31-42-86:/var/www/html/testlink$ sudo nano config_db.inc.php
?php
define('Db_type ', 'mysql');
define('DB_USER', 'root');
define('DB_PASS ', 'password');
define('DB_HOST ', 'localhost');
define('DB_NAME', 'testlink');
define('DB_TABLE_PREFIX ', '');
?>
```

Test Case 생성



The screenshot shows the TestLink 'Test Case' creation page. The left sidebar contains filters for Test Case ID, Title, Suite, Status, Importance, and Execution type. The main area displays a 'Warning! This Test Case version has been executed.' message. Below this, there is a 'Summary' section, 'Preconditions' (e.g., '1. hello world!'), and 'Step actions' (e.g., '1. Click button' with 'Expected Results' of 'Pop up'). The 'Status' is set to 'Draft', 'Importance' to 'Medium', and 'Execution type' to 'Manual'. There are also fields for 'Keywords' and 'Requirements' (e.g., '[testreq] test [v1] : 123').

Requirements 생성



The screenshot shows the TestLink 'Requirements' creation page. The left sidebar contains filters for Document ID, Title, Status, Requirement Type, Req. Spec. Type, Expected no. of testcases, Req. version of SQR, and Test Case ID. The main area displays a 'Warning! You can not edit this version because it has been frozen.' message. Below this, there is a 'Summary' section with details like 'test: 123', 'Version 1 revision 1', 'Status: Draft', 'Type: Use Case', and 'Number of test cases seeded: 1'. There is also a 'Coverage' section showing a link to 'test-1 : testcase (Version 1)'. The 'Creator' information is 'Created on 05-06-2014 08:21:20 by admin'.

03 Overall

Requirement Management




REDMINE
flexible project management

Configuration Management




git


GitHub



IJ

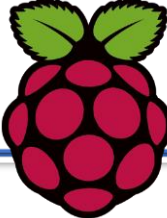
Build

Test
JUnit **5**


Team Communication





slack




CI Server




Jenkins

Build

Gradle


sonarqube


Pmd 
CS 

Static Analysis


aws


REDMINE
Issue tracking


TestLink
System Test

감사합니다!