



Software Verification

Introduction to

Sonarqube, Testlink, PMD, Checkstyle, Findbug

POWERED BY

201210908 윤성일

201311265 김상원

201214150 정성철





정적 테스트

정적 테스트?

실제 실행 없이 컴퓨터 소프트웨어를 분석

대부분의 경우에 분석은 소스코드 버전 중 하나의 형태로 수행

Static WhiteBox Test

프로그램을 실행하지 않고 소프트웨어의 설계, 코드나 구조 등에서 버그를 찾을 수 있는 방법

코드 검토 체크리스트를 사전에 정해놓고 검사

Static BlakBox Test

프로그램의 소스코드를 파악할 수도 없고, 실행시키지도 않는 상태에서 수행하는 검사

프로그램 명세서에 대한 테스트



Sonarqube

Sonarqube?

- 프로젝트의 품질을 관리할 수 있도록 여러가지 모니터링 툴을 제공하는 오픈소스 플랫폼
- 단독으로 사용되기 보다는 Jenkins 같은 CI 서버와 연동이 되어서 사용
- Java를 포함한 20가지가 넘는 프로그래밍 언어 로 제작된 프로젝트의 모니터링을 제공





Sonarqube

특징

- 심플한 Web Monitoring UI (Dashboard)
- 테이블과 차트를 이용하여 시간이 지남에 따라 프로젝트가 얼마나 개선되고 있는지 확인가능
- 소스의 중복이나 복잡도 잠재적인 버그의 정보 등을 프로젝트 단위부터 파일단위까지 제공

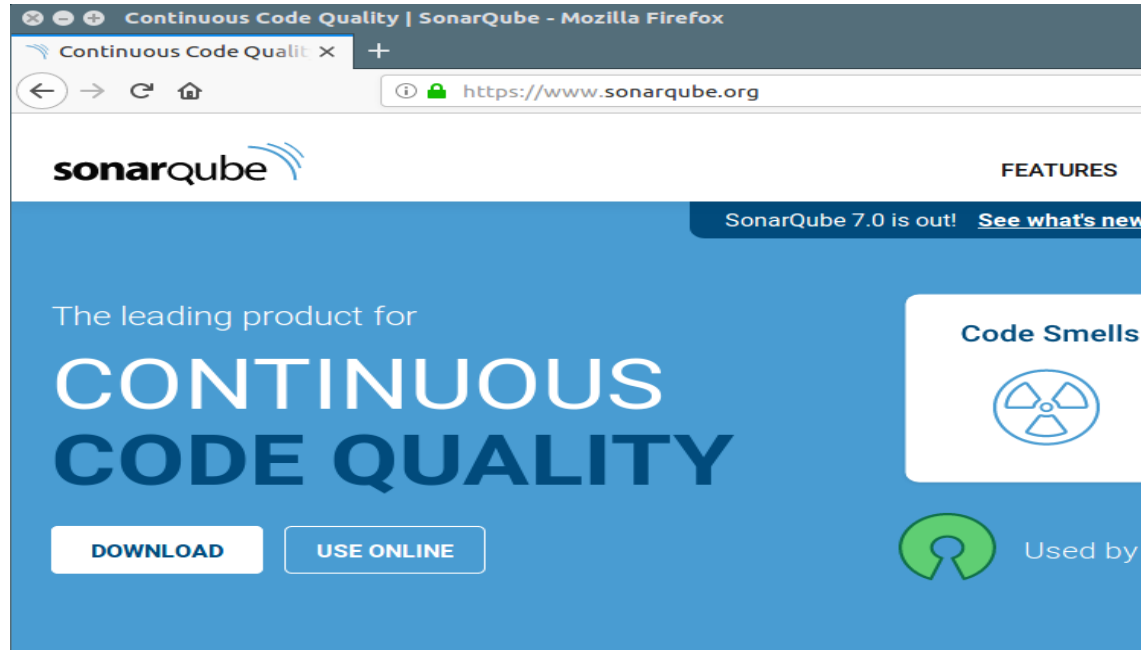




Sonarqube

설치 과정

https://www.sonarqube.org 접속후 다운로드

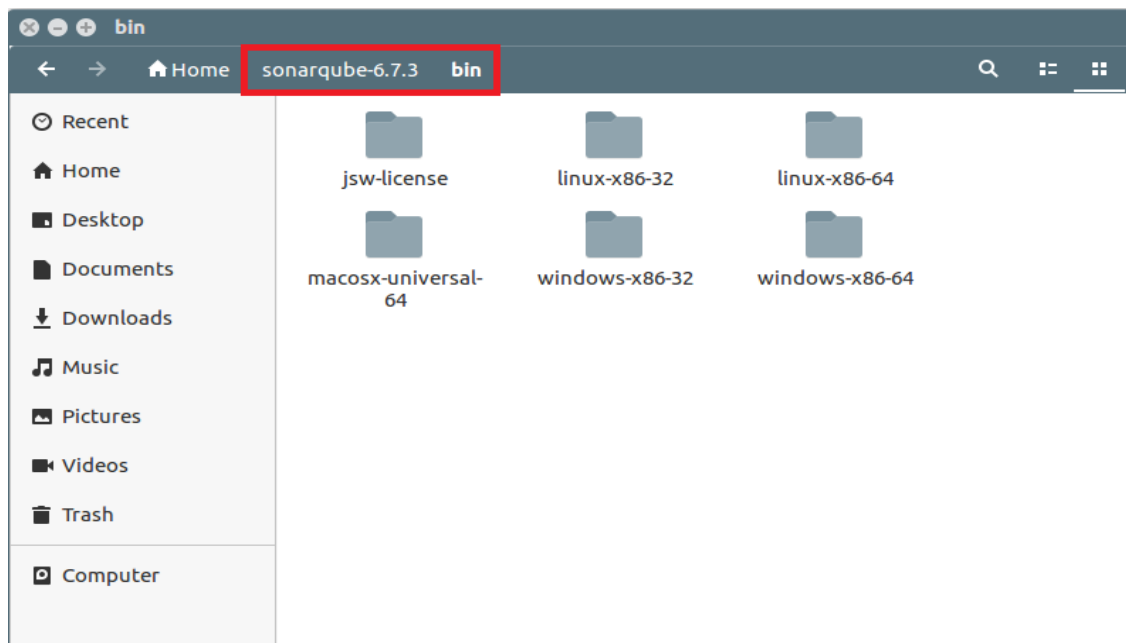




Sonarqube

Sonarqube 설치

압축을 해제한 폴더 sonarqube-x.x.x/bin폴더에서 맞는 운영체제의 폴더로 진입





Sonarqube

Sonarqube 설치

명령어입력시 소나큐브 시작(Linux 기준)

```
sangwon-ssvm@ubuntu: ~/sonarqube-6.7.3/bin/linux-x86-64
sangwon-ssvm@ubuntu:~/sonarqube-6.7.3/bin/linux-x86-64$ ./sonar.sh
Usage: ./sonar.sh { console | start | stop | restart | status | dump }
sangwon-ssvm@ubuntu:~/sonarqube-6.7.3/bin/linux-x86-64$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
sangwon-ssvm@ubuntu:~/sonarqube-6.7.3/bin/linux-x86-64$
```

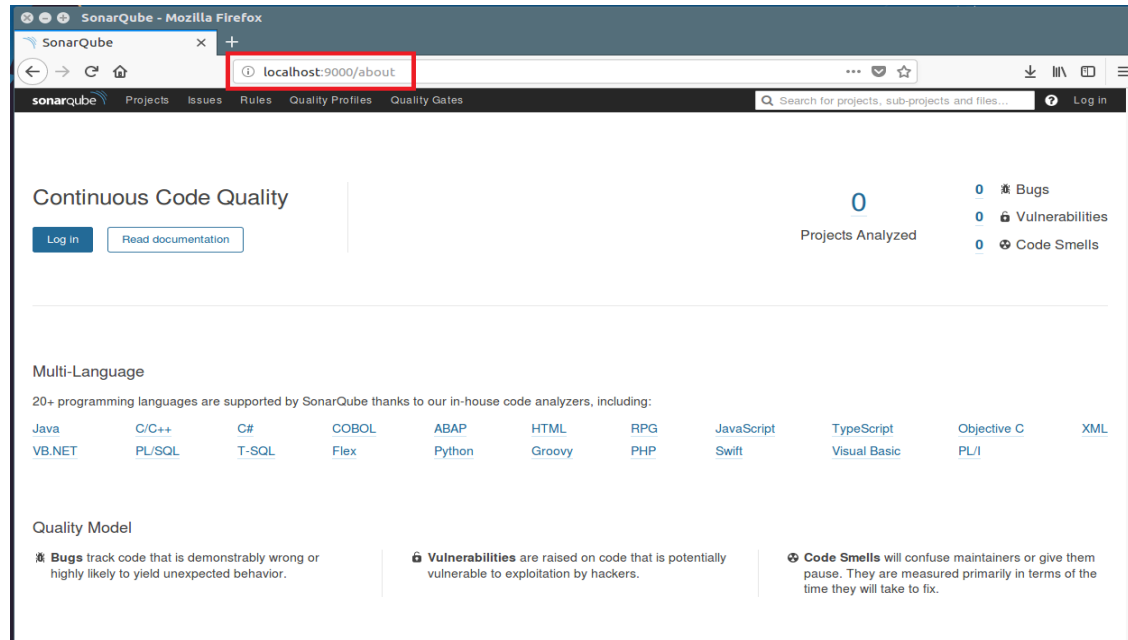
설정 정보는 sonarqube-x.x.x/conf



Sonarqube

Maven 연동

Sonarqube 접속(기본 접속 주소 <http://localhost:9000>)





Sonarqube

Maven 연동

Maven 과 연동을 위해 Sonarqube 에서 설정

The screenshot shows the SonarQube web interface in a Mozilla Firefox browser. The URL is localhost:9000/projects. The interface is in the 'Projects' view, showing a search bar and a list of projects (currently empty). The main content area displays a tutorial for running a SonarQube analysis on a project. It consists of two steps:

- 1 Provide a token**: A green checkmark indicates success. A token is displayed: `lllssss94: 75007b8646546ee028950d4b81cbc587c694736c`.
- 2 Run analysis on your project**: This step includes:
 - What is your project's main language?**: A dropdown menu is set to 'Java'. Other options include 'C# or VB.NET' and 'Other (JS, Python, PHP,...)'.
 - You are developing primarily in Java: what is your build technology?**: A dropdown menu is set to 'Maven'. Other options include 'Gradle'.
 - Execute the SonarQube Scanner for Maven from your computer**: A text box contains the command: `mvn sonar:sonar \n -Dsonar.host.url=http://localhost:9000 \n -Dsonar.login=75007b8646546ee028950d4b81cbc587c694736c`. A 'Copy' button is next to it.
 - Instructions: 'Running a SonarQube analysis with Maven is straightforward. You just need to run the following command in your project's folder.' and 'Please visit the official documentation of the SonarQube Scanner for Maven for more details.'
 - Next steps: 'Once the analysis is completed, you will be able to browse your project at the URL displayed at the end of the logs.'
 - A 'Finish this tutorial' button is at the bottom right.

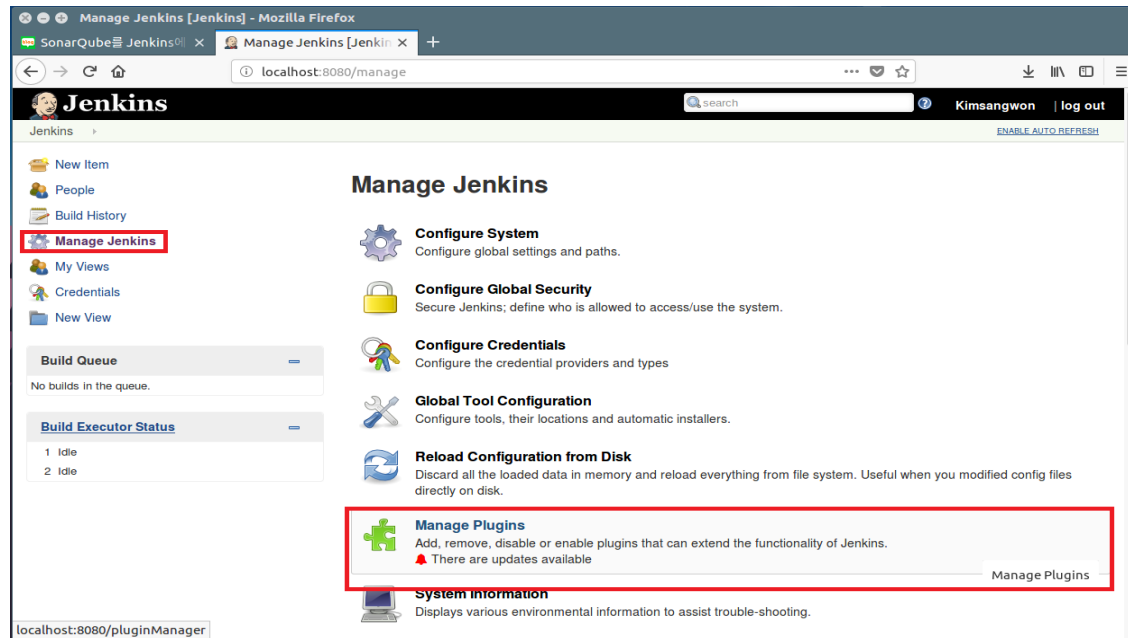
At the bottom of the interface, there is a 'Maintainability' section with a 'Code Smells' icon and a table showing '0' and '0' for different categories. A footer contains the text: 'The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine. SonarQube™ technology is powered by SonarSource SA. Version 6.7.3 (build 36370) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About'.



Sonarqube

Jenkins 연동

플러그인 관리메뉴





Sonarqube

Jenkins 연동

SonarQube Scanner 플러그인 설치

Filter:

Updates Available Installed Advanced

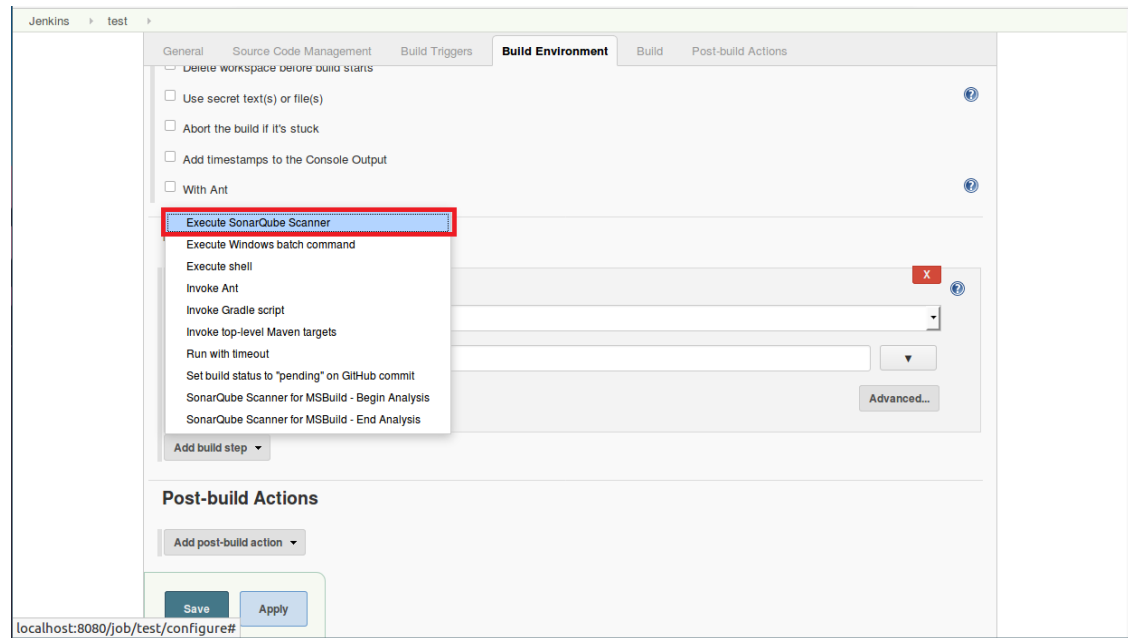
Install	Name	Version
<input type="checkbox"/>	CodeSonar	2.0.6
<input type="checkbox"/>	SonarQube Scanner	2.7.1
<input type="checkbox"/>	Sonargraph Integration	2.1.2



Sonarqube

Jenkins 연동

설정하려는 프로젝트의 설정에서 build 단계 추가
SonarQube Scanner 선택





Sonarqube

Jenkins 연동

설정 값 입력

Task to run	<input type="text"/>	?
JDK	(Inherit From Job)	?
JDK to be used for this SonarQube analysis		
Path to project properties	<input type="text"/>	?
Analysis properties	<pre>sonar.projectKey=myproject sonar.projectName=my project name sonar.projectVersion=1.0.0 sonar.sources=src/main/java sonar.java.binaries=target/classes sonar.language=java sonar.sourceEncoding=UTF-8</pre>	?
Additional arguments	<input type="text"/>	?
JVM Options	<input type="text"/>	?





Sonarqube

Jenkins 연동

연동 완료

The screenshot shows the Jenkins interface for a project named 'test'. The left sidebar contains a menu with 'SonarQube' highlighted in a red box. The main content area shows 'Project test' with a 'SonarQube' link also highlighted in a red box. Below this, there are links for 'Workspace' and 'Recent Changes'. A 'Build History' table is visible, showing two builds: #2 (Apr 12, 2018 6:30 PM) and #1 (Mar 30, 2018 1:37 PM). A 'Permalinks' section lists various build links.

Build Number	Timestamp
#2	Apr 12, 2018 6:30 PM
#1	Mar 30, 2018 1:37 PM

- [Last build \(#2\), 13 min ago](#)
- [Last stable build \(#1\), 13 days ago](#)
- [Last successful build \(#1\), 13 days ago](#)
- [Last failed build \(#2\), 13 min ago](#)
- [Last unsuccessful build \(#2\), 13 min ago](#)
- [Last completed build \(#2\), 13 min ago](#)



PMD&Checkstyle&Findbugs

PMD

- 작성한 코드에 대한 위반사항(위반되는 코딩 스타일, 불필요한 코드)를 찾음
- 위반 사항을 명시한 report파일(pmd.xml, cpd.xml file)에 대한 수정이 쉬움
- 한 번의 클릭으로 수많은 규칙에 대한 수정이 가능

Checkstyle

- 여러사람과 작업 시 손쉽게 코딩 스타일을 맞출 수 있음
- 표준코딩스타일을 손쉽게 프로젝트에 적용

Findbugs

- 프로그램 버그 찾기
- 컴파일 된 바이트 코드 검색
- 각 버그에 대한 수정 안내서 제공





Sonarqube

설치

Sonarqube 에서 PMD,Checkstyle,Findbugs 플러그인 설치

The screenshot shows the Sonarqube '환경설정' (Environment Settings) page. The 'Available' tab is selected, and a search bar is visible. The following table lists the installed plugins:

Plugin Name	Version	Status
C#	4.4	installed
Checkstyle	2.4	installed
Findbugs	3.3	installed
Korean Pack	1.2.0	installed
PMD	2.5	installed



Sonarqube

실행예시

품질 프로파일 생성

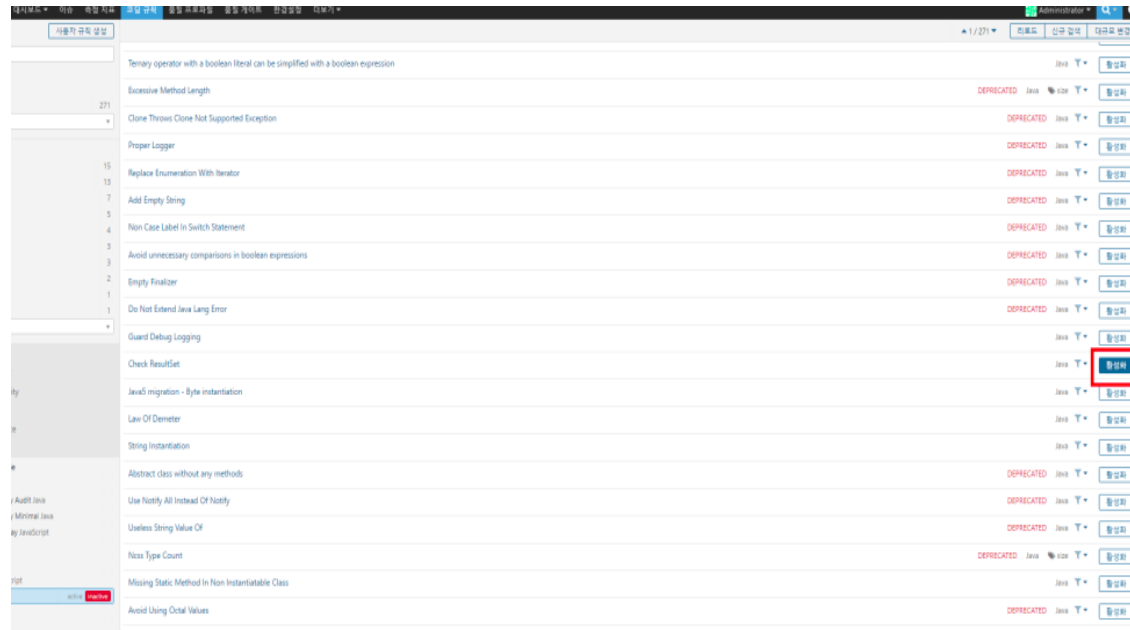




Sonarqube

실행예시

품질 프로파일 에 Coding Rule 활성화

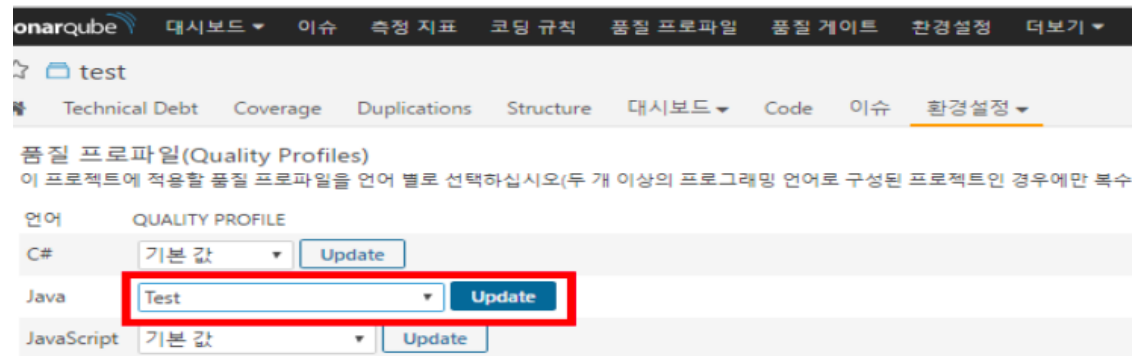
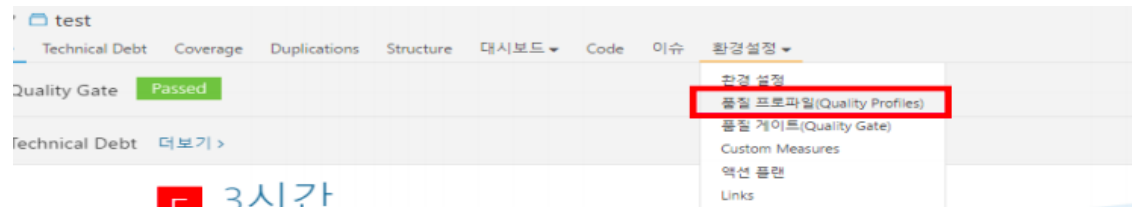




Sonarqube

실행예시

프로젝트에서 품질 프로파일 설정





시스템 테스트

시스템 테스트?

- 전체적으로 통합된 애플리케이션들을 테스트
- 수행 시간, 파일 저장 및 처리 능력, 최대 부하, 복구 및 재시동 능력, 수작업 절차 등을 점검
- 시스템의 실제 동작과 원래 의도했던 요구사항과는 차이가 없는지 등을 판단
- 시스템 테스트는 사용자 관점에서 소프트웨어 외부 동작들(external workings)을 수반





TestLink

TestLink?

- Web 환경의 인터페이스(사용자는 인스톨 없이 어디서든지 접근가능)
- 테스트의 우선순위 부여 기능 및 사용자의 활동영역 정의(리더, 테스터)
- 다양한 종류의 DB를 지원(MySQL, PostgreSQL, MS-SQL 등)
- 트리구조를 이용한 테스트 케이스 작성기능
- 다양한 형식의 보고서 제공(HTML, MS-Word, MS-Excel) 및 E-mail에 직접보고서를보낼 수 있는 기능 제공





Testlink

설치

https://bitnami.com/stack/testlink/installer 접속후 다운로드

The screenshot shows the Bitnami TestLink Installers page. At the top, there is a navigation menu with 'Overview', 'Installers', 'Virtual Machines', 'Cloud', and 'Containers'. The main heading is 'TestLink Installers' with a 'DEMO IN BROWSER' button. Below this, there is a paragraph explaining that Bitnami native installers automate the setup of a Bitnami application stack on Windows, OS X, or Linux. A list of links includes 'Readme', 'Changelog', and 'Documentation'. The 'Windows' section is highlighted, showing a table with columns for 'Version', 'Size', and 'Checksum'. The table lists 'TestLink 1.9.17-0' with a size of '123 MB' and a 'Download' button. Below the table, there is a 'Modules' section with a note about installing TestLink on top of an existing Bitnami WAMP installation.

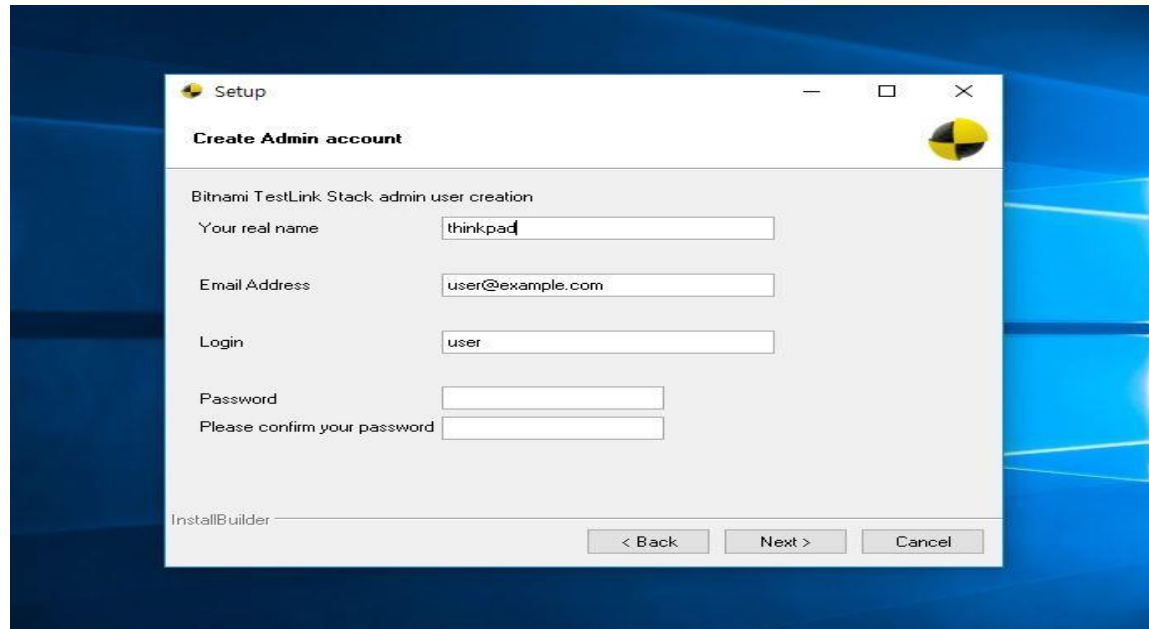
Version	Size	Checksum
TestLink 1.9.17-0	123 MB	show Download Recommended



Testlink

설치

Testlink 에서 사용할 계정 생성





TestLink

기본 용어

Test Project

- 기본적인 조직 유닛
- 특성에 따라 여러 프로젝트로 나눠서 테스트를 진행
- 분할의 기준은 자유
- 테스트 팀 단위로 분할 하는 것이 관리에 용이





Testlink

기본 용어

Test Specification

- Test Spec은 테스트를 진행하고자 하는 테스트 케이스들의 집합
- Test Suite Test Case로 분류
- Test Suite은 대분류 Test Suite의 소분류를 Test Case

Test Plan

- 실제 진행하는 테스트를 의미
- 테스트 대상 시스템의 특정 버전을 테스트 하기 위해서 Test Spec내의 Test Suite/Case의 모음
- Test Plan에 Test Case를 맵핑시 엔지니어 선택 가능





Testlink

기본 용어

Test Execution

- 테스트 결과 Pass/Fail 여부를 체크
- 실패한 케이스일 경우에는 재 테스트

Test Report

- 테스트 실패, 성공 여부, 주요 Test Category (TestSuite)별 성공 실패 여부를 리포팅





Testlink

실행예시

Bitnami Testlink Stack Manager를 실행
Go to Application 을 클릭





Testlink

실행예시

설치시 생성한 계정 입력

TestLink
1.9.16 (Moka pot)

Login Name

Password

Log in

New User? Lost Password?

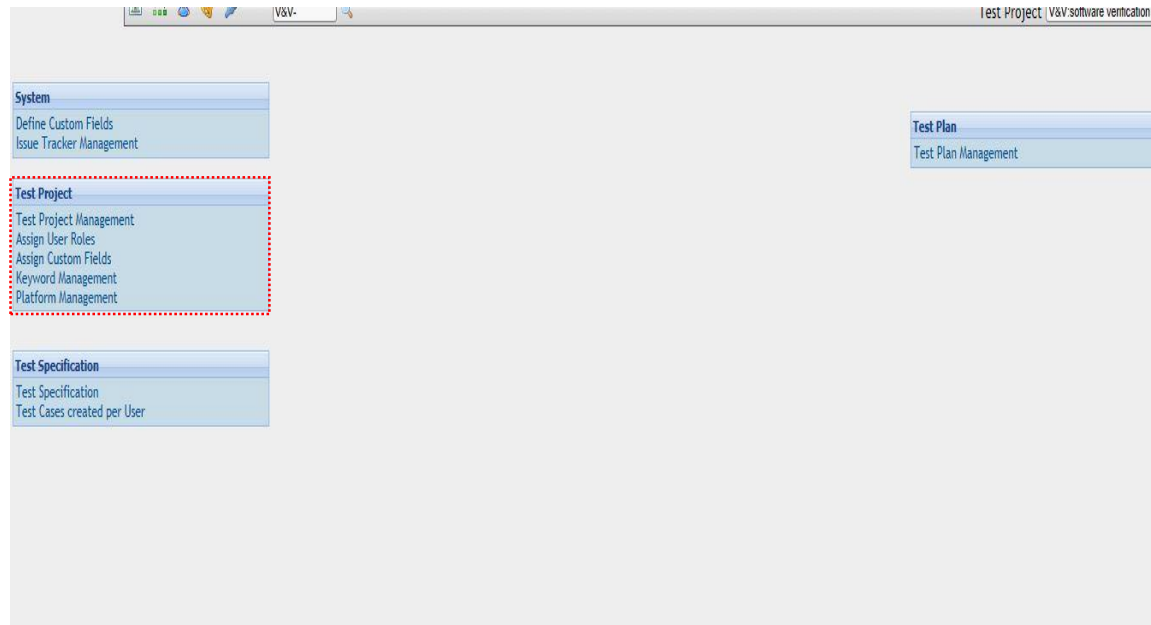




Testlink

실행 예시

Test Project 생성





Testlink

실행 예시

정보 입력후 Test Project 생성

The screenshot shows the 'Create Test Project' form in Testlink. The form includes the following fields and options:

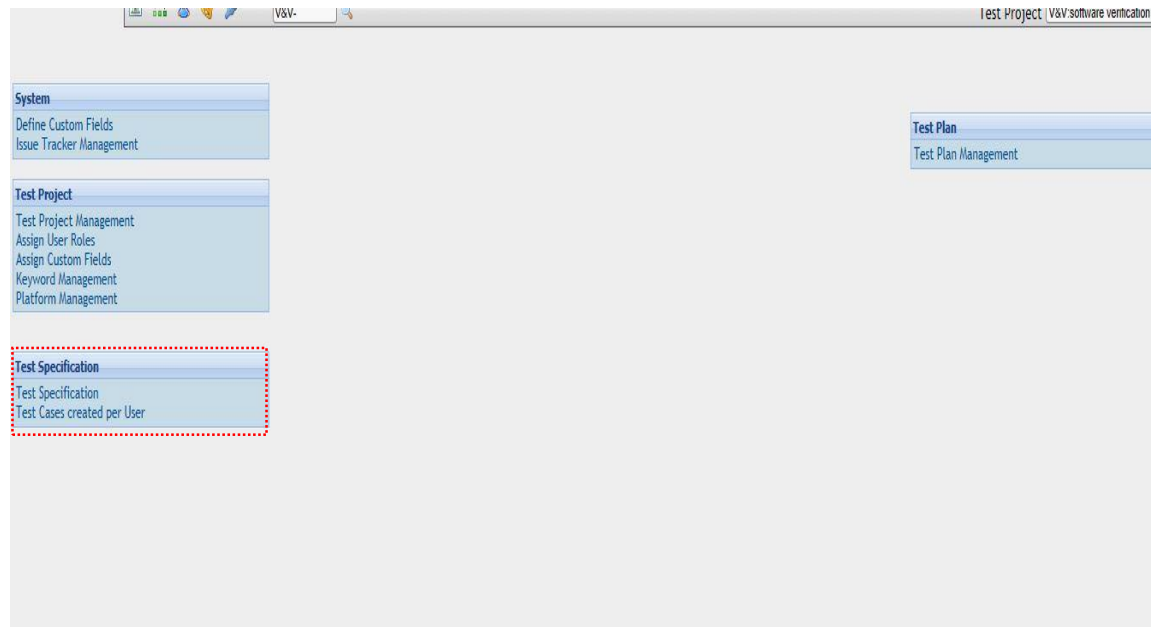
- Create from existing Test project?**: A dropdown menu with 'No' selected.
- Name ***: A text input field.
- Prefix (used for Test case ID) ***: A text input field.
- Project description**: A rich text editor with a toolbar containing icons for Source, Undo, Redo, Bold, Italic, Underline, Strikethrough, Text color, Background color, Bulleted list, Numbered list, Link, and Unlink.
- Enhanced features**: A list of checkboxes:
 - Enable Requirements feature
 - Enable Testing Priority
 - Enable Test Automation (API keys)
 - Enable Inventory



Testlink

실행 예시

Test Suite / Test case 생성

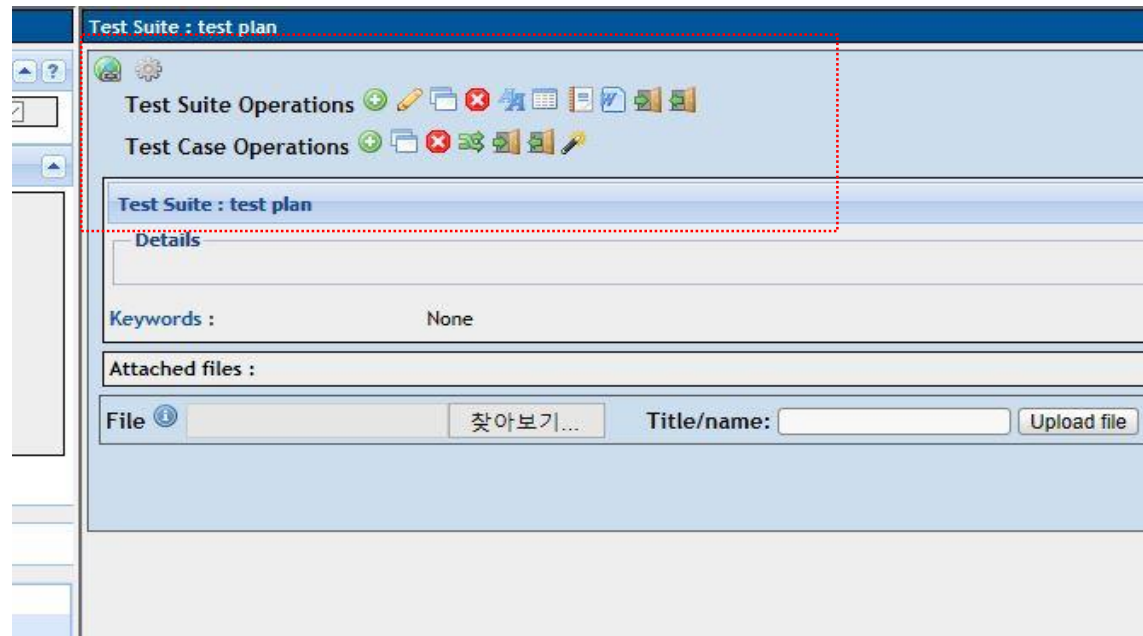




Testlink

실행 예시

정보 입력후 Test Suite / Test case 생성

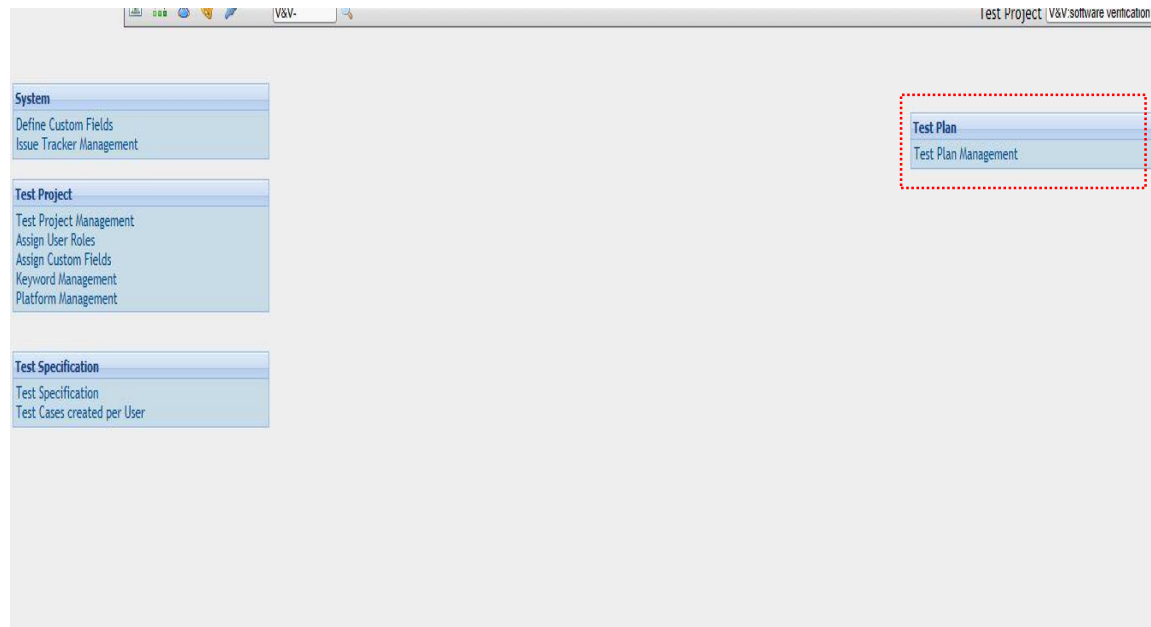




Testlink

실행 예시

Test Plan 생성

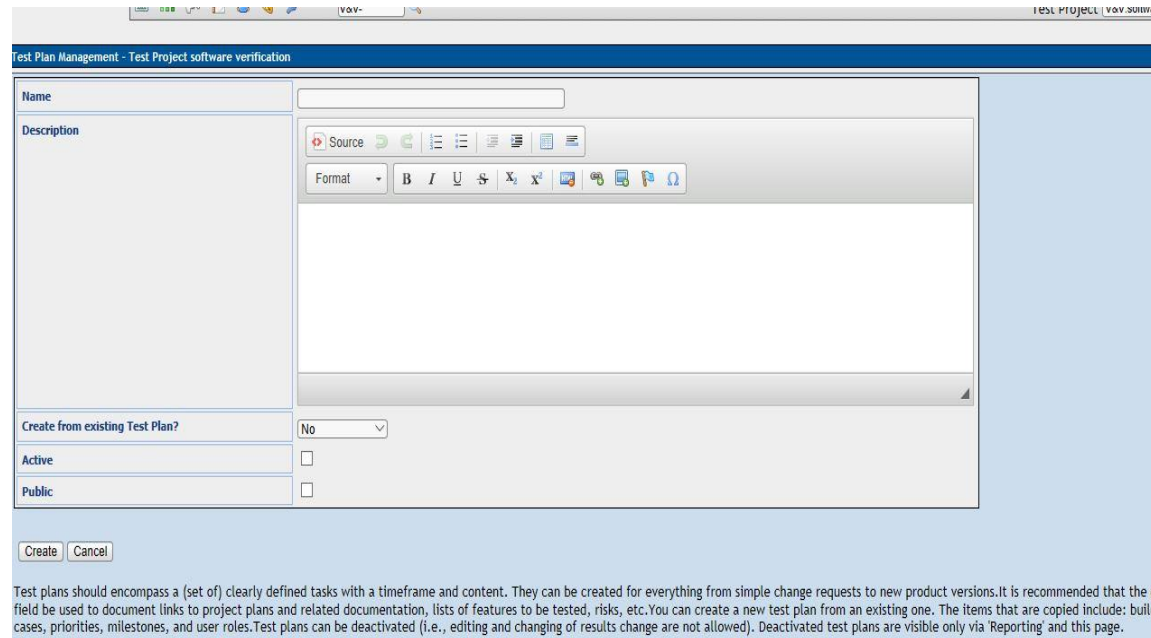




Testlink

실행 예시

정보 입력후 Test Plan 생성





Testlink

실행 예시

Test Case 활성화 Assign to 탭을 클릭
Test 진행 엔지니어 선택후 저장

The screenshot shows the TestLink 1.9.16 (Moka pot) interface. The browser title bar indicates the Test Project is 'V&V:software verification'. The main content area is titled 'Assigning Test Case execution tasks for build ver 1.0 in test plan test for testing'. It includes a 'Check/uncheck all Test cases' button, a 'Bulk user assignment' section with a text input field, 'Do' and 'Save' buttons, and an 'email Assignments' button. Below this is a table with the following data:

test plan			
<input checked="" type="checkbox"/>	Test Case [Version]	Priority	Assigned to
<input checked="" type="checkbox"/>	V&V-1 : Dependability [1]	Medium	thinkpad



Testlink

Redmine 연동

Redmine API 접근키 입력
Redmine 메인 과 일감의 URL 입력
project 명에는 이슈를 받을 Redmine Project 명을 입력

The image shows two screenshots from the Testlink interface. The left screenshot is the 'Issue Tracker Management' page, showing the configuration for an issue tracker named 'software verification'. The 'Configuration' field contains XML code for connecting to a Redmine instance. A red dashed box highlights this code. The right screenshot is the '내 계정' (My Account) settings page, showing the 'API 접근키' (API Access Key) field. A red dashed box highlights this field, and a red arrow points from it to the configuration code in the left screenshot.



Testlink

Redmine 연동

Test 결과



이전 실행한 결과들 - 빌드 : ver 1.0

날짜	담당자	상태	Exec (min)	버전	버그 관리
2018/04/14 18:46:20	thinkpad	실패		1	
빌드	Test Case Step	관련 버그			
ver 1.0	6 : [신규] : /software verification/test plan/V&V-1:Dependability - Executed ON (ISO FORMAT): 2018-04-14 18:46:20				
ver 1.0	7 : [신규] : /software verification/test plan/V&V-1:Dependability - Executed ON (ISO FORMAT): 2018-04-14 18:46:20				
ver 1.0	8 : [신규] : /software verification/test plan/V&V-1:Dependability - Executed ON (ISO FORMAT): 2018-04-14 18:46:20				
2018/04/14 18:38:02	thinkpad	실패		1	
2018/04/13 19:46:35	thinkpad	실패		1	
2018/04/13 19:40:10	thinkpad	실패	10.00	1	

테스트 케이스 V&V-1 :: 버전: 1 :: Dependability
테스터 : thinkpad





Testlink

Redmine 연동

Test 결과

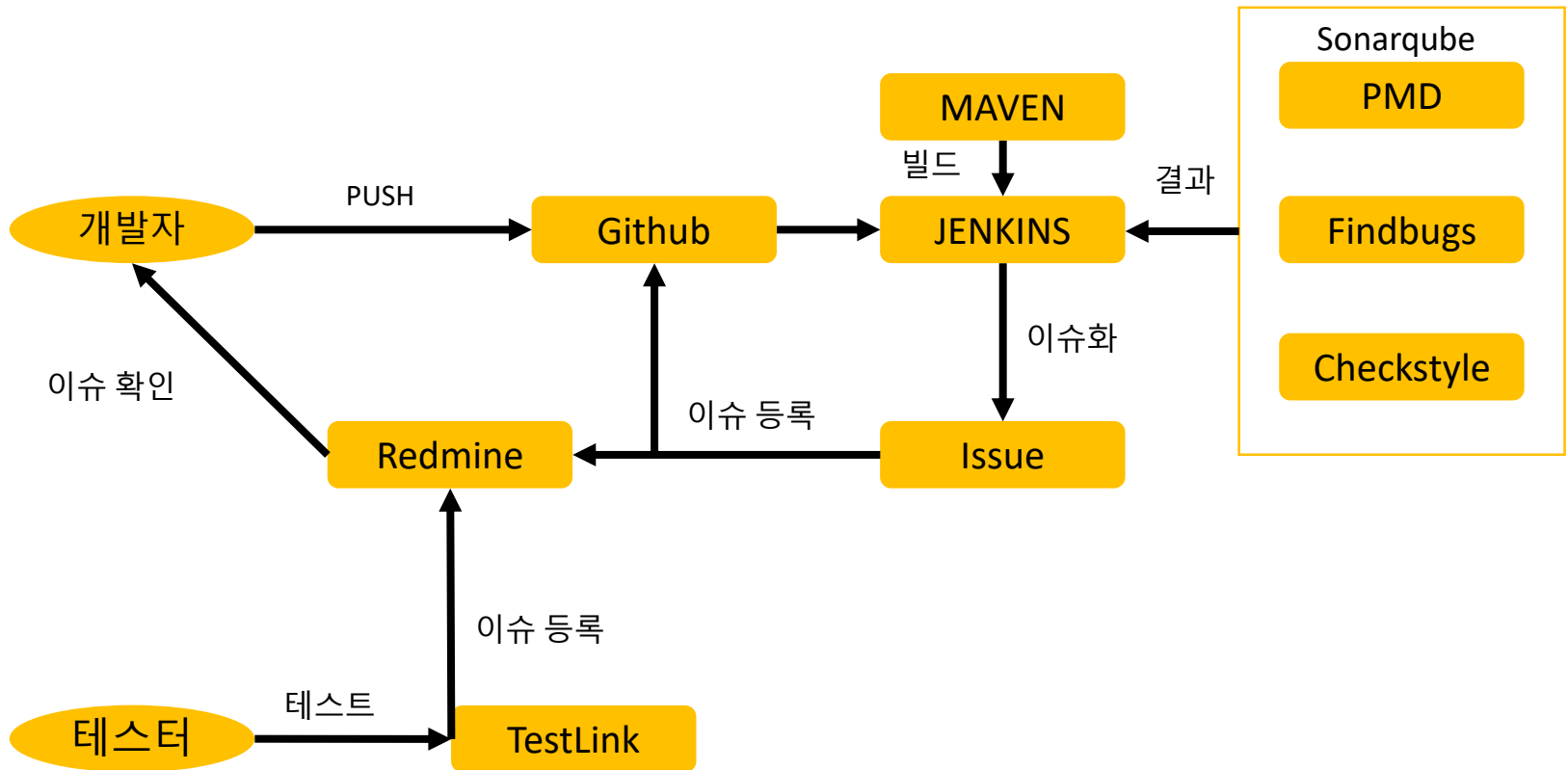
The screenshot shows a Redmine interface with a blue header bar. Below the header, there is a section titled "REDMINE Issue Created (summary:/software verification/test plan/V&V-1:Dependability)". Below this, there is a summary box containing the text: "summary*) /software verification/test plan/V&V-1:Dependability - Executed ON (ISO FORMAT): 2018-04-14 18:46:20". To the right of the summary box, there is a "추가:" label and a dropdown menu. Below the summary box, there is a "test" section. To the right of the "test" section, there are two tables. The first table is titled "같은 일감 (1)" and has columns: "프로젝트", "유형", "상태", "제목". The second table is titled "보고한 일감 (1)" and has columns: "#", "프로젝트", "유형", "상태", "제목". Below the tables, there is a "Link in Issue Tracker to Test Case Execution Feature" section with a "Close" button.

같은 일감 (1)	보고한 일감 (1)				
프로젝트	#	프로젝트	유형	상태	제목
verification-validation	6	verification-validation	결함	신규	test
		software verification/test plan/V&V-1:Dependability - Executed ON (ISO FORMAT): 2018-04-14 18:46:20			



CTIP

구축





Problems to be solved

Problems to be solved

AWS 서버로 CTIP 환경을 구축 서버가 자주 다운

Redmine Jenkins 연동





SOFTWARE

I N D E X

THANK YOU

Sonarqube | PMD | Checkstyle | Findbug | Testlink

201210908 윤성일
201311265 김상원
201214150 정성철

김 상 원

정 성 철

윤 성 일



Powered by dslab