

Introduction to SVN, MantisBT, JFeature, JUnit 사용법 및 CTIP 개론

과 목 명: 소프트웨어 모델링 및 분석
교 수 명: 유준범 교수님
제 출 일: 2016.04.07.(목)
팀 원: 201211341 김태현
201411269 고수창
200911411 이상규

Contents

1. SVN

- a. SVN 개요
- b. SVN 설명
- c. SVN 사용법

2. MantisBT

- a. MantisBT 개요
- b. MantisBT 설치
- c. MantisBT 사용법

3. JFeature

- a. JFeature 개요
- b. JFeature 사용법

4. JUnit

- a. JUnit 개요
- b. JUnit 사용법

5. CTIP 개론

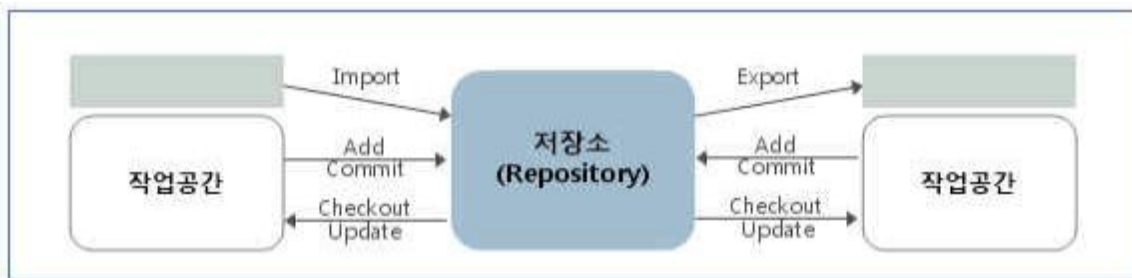
1. SVN

a. SVN 개요

- Subversion(SVN, 버전관리 시스템)은 프로젝트 전반적으로 사용하는 파일들을 관리하기 위한 시스템이다. Subversion은 다수의 개발자가 프로그램을 개발할 때 개발된 버전 관리는 통해 History 를 관리할 수 있도록 하며 전체 팀의 공통 모듈을 일관성 있게 관리할 수 있도록 한다.

b. SVN 설명

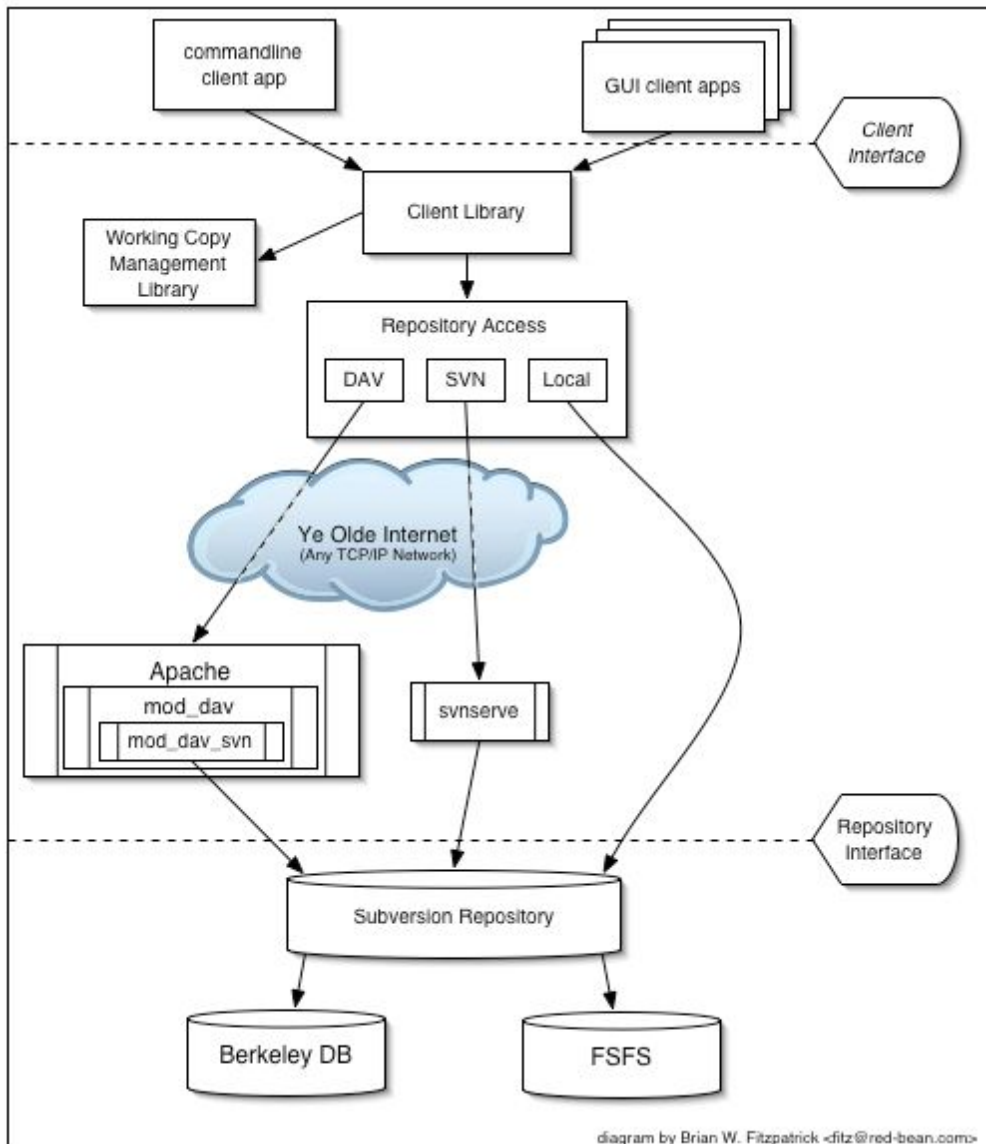
- Subversion은 실제의 작업용 디렉토리(작업공간) 이외에, 저장소(Repository)라고 하는 영역에 최초의 데이터와 그 이후의 갱신 이력이나 변경 분 정보를 저장한다.



- 저장소는 여러 개 생성 가능
- 사용할 수 있는 파일이나 디렉토리는 import로 저장소에 등록
- checkout을 실행하면 저장소 내용이 작업공간에 복사되고 동시에 관리 정보를 수록하는 .svn 디렉토리가 생성 (이 작업은 최초 1회만 필요하다.)
- commit으로 작업공간의 변경 내용이 저장소에 등록되며, 저장소에 있는 변경 내용을 작업공간에 반영하려면 update를 사용
- 관리 정보 없이 저장소 내용을 가져오려면 export를 사용

- Subversion의 아키텍처

저장소(Subversion Repository)에서는 모든 데이터의 버전들을 관리하며, 클라이언트 프로그램은 로컬 환경의 복사된 데이터를 관리한다. 이 사이에서 다양한 저장소 액세스(RA, Repository Access)를 통하여 통신한다.



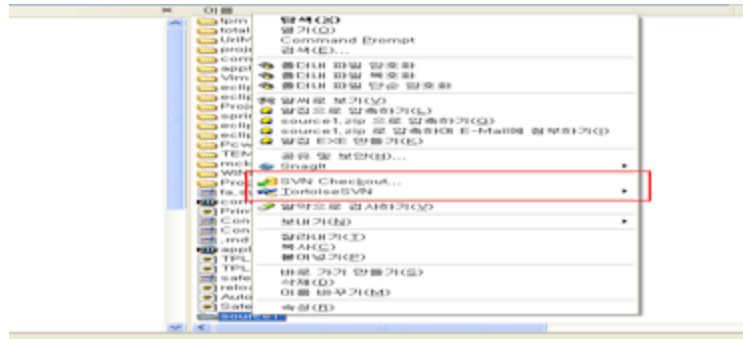
- Subversion의 구성요소

- svn: 커맨드라인 클라이언트 프로그램
- svnversion: 복사된 작업공간의 버전을 관리하는 프로그램
- svnlook: Subversion의 저장소를 조사하기 위한 툴
- svnadmin: Subversion의 저장소를 조정하거나 복원하기 위한 시스템 관리자용 프로그램
- svndumpfilter: Subversion 저장소의 덤프 파일 형식 데이터에 대한 필터 프로그램
- mod_dav_svn: 네트워크의 다른 사용자가 저장소를 사용할 수 있도록 하는 Apache HTTP 서버 플러그인 모듈
- svnserve: demon이나 SSH를 통해 사용하는 독립 서버 프로그램으로 네트워크에 있는 저장소를 사용할 수 있도록 함

c. SVN 사용법

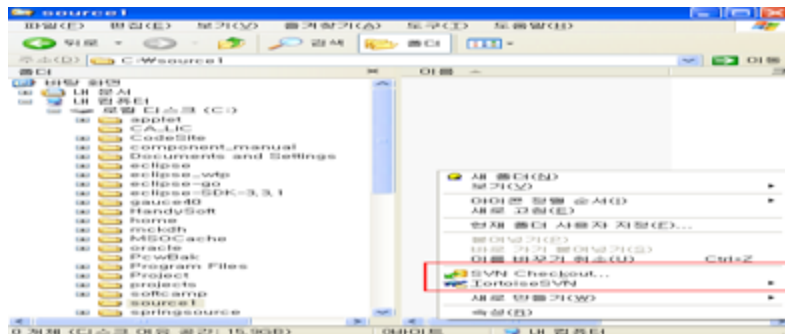
(1) SVN 클라이언트 설치

<http://tortoisesvn.net/downloads>에서 시스템에 알맞은 버전을 설치한다. 설치 후 탐색기 컨텍스트 메뉴에서 SVN 작용 메뉴가 추가된 것을 확인할 수 있다.

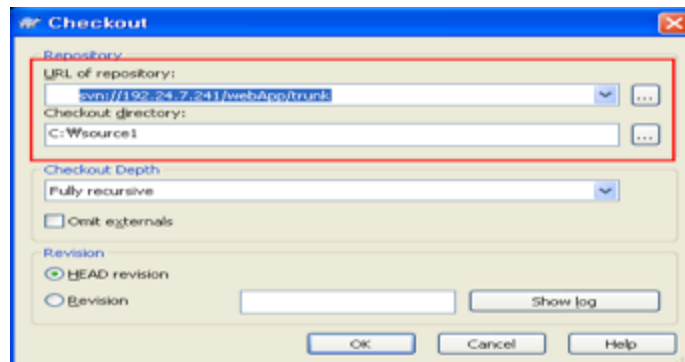


(2) 프로젝트 Checkout (저장소 내용을 작업공간에 반영)

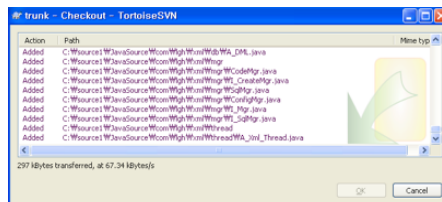
1. 작업 디렉토리를 만든다. 예) D:\SMA_project
2. 해당 디렉토리 폴더의 컨텍스트 메뉴에서 SVN Checkout을 선택한다.



3. 저장소 주소를 입력한다. 예) svn://192.24.7.241/SMA/trunk



4. Checkout 완료까지 대기한다.



(3) 프로젝트 Commit (작업공간의 변경 내용을 저장소에 반영)

1. 작업 디렉토리 폴더 내에 Commit할 파일을 선택하고 컨텍스트 메뉴에서 SVN Commit을 누른다.
2. add할 파일을 체크하고 확인을 누른다.

- Subversion의 주요 기능

- 디렉토리의 버전 관리
 - Subversion은 시간과 함께 디렉토리 트리 전체의 변경에 확인할 수 있는 가상적인 버전관리 시스템을 구현하고 있다.
 - 디렉토리에도 버전 정보가 추가된다.
- 버전 이력 기능 강화
 - Subversion에서는 파일과 디렉토리의 모두에 대해서 추가, 삭제, 복사, 명칭 변경을 할 수 있다.
 - 신규 추가되는 모든 파일은, 그 상태에서 새롭게 시작되는 이력을 갖는다.
- Atomic Commit
 - 변경점의 모임은, 그것 전체가 저장소에 완전하게 반영되는지, 전혀 반영되지 않는지 중 하나다.
 - 개발자는 논리적으로 한 묶음의 변경을 만들어 Commit할 수 있어 일부만이 저장소에 반영되어 버리는 문제가 발생하지 않는다.
- 메타데이터 버전관리
 - 파일과 디렉토리는 각각 관련한 속성 키와 값의 조합이며 임의의 키/값의 조합을 생성해 보존할 수 있다.
 - 속성도 파일의 내용과 동일하게 버전관리 된다.
- 네트워크 레이어의 선택
 - Subversion은 저장소에 액세스 하기 위한 추상 레이어가 있어, 새로운 네트워크 프로그램을 간단하게 구현할 수 있다.
 - Subversion은 HTTP 서버의 확장 모듈 플러그인으로 사용할 수 있어 신뢰성과 상호호환성을 가진다. 서버가 제공하고 있는 기존의 기능(인증, 인가, 데이터 압축 등)을 곧바로 이용할 수 있다.
 - 쉽게 단독 Subversion 프로세스를 이용할 수 있다. 이 서버는 독자적인 프로토콜에 의하여 SSH를 이용한 터널 통신을 실행할 수 있다.
- 데이터 처리의 일관성
 - Subversion은 동일한 알고리즘을 사용해 파일의 변경 내용을 표현한다.
- 효율적인 브랜치, 태그의 작성
 - 브랜치와 태그를 작성하는 비용은 프로젝트의 크기에 비례하지 않는다. Subversion은 하드 링크로서 알려져 있는 방법과 비슷한 방법을 사용하여 단지 프로젝트를 복사하는 것으로 브랜치와 태그를 만든다. 따라서 브랜치, 태그의 작성에는 상대적으로 짧은 시간이 소모된다.
- 확장하기 쉬움
 - Subversion은 잘 설계된 API로 완성된 C의 공유 라이브러리의 모임으로 구현되어 있으며, 이는 유지보수나 다른 어플리케이션이나 언어와 연동하여 사용하기가 쉽다.

- SVN의 주요 명령어

- checkout
 - 저장소를 checkout하면 로컬 컴퓨터에 원하는 프로젝트의 사본을 생성하며 이 사본은 지정된 저장소의 최신 개정판을 포함함
 - 최신 버전이 아닌 원하는 버전으로 checkout하는 것도 가능함
- commit
 - 체크인(checkin)이라고도 하며 checkout한 소스를 수정, 파일 추가, 삭제한 후 로컬 사본에 대한 변경 사항을 저장소에 갱신하는 것임
 - commit을 하면 Subversion의 경우 전체 리비전이 1 증가하게 됨
 - commit 이후에 사용자는 update로 특정 파일의 최신 버전을 볼 수 있음
- update
 - 사용자의 가장 최근 변경 사항이 있는 작업 사본을 저장소와 동기화하는 것으로, 파일의 가장 최근 작업 사본을 로컬 드라이브로 가져옴(바뀐 부분만 가져옴)
 - 파일을 변경하기 전에 언제나 갱신을 수행하는 것이 최우선 원칙임
- import
 - 아무것도 들어있지 않은 저장소에 맨 처음 소스를 넣는 작업임
- export
 - checkout과 달리 버전 관리 파일들을 뺀 순수한 소스 파일을 받아올 수 있음
 - 오픈소스 프로젝트의 경우 소스를 압축하여 릴리즈 할 때 사용함
- conflict
 - 한 파일의 일부 행을 두 명 이상의 사용자가 변경했을 때 또는 저장소에 파일을 갱신할 때 충돌이 발생함
- revision
 - 소스파일 등을 수정하여 commit하게 되면 일정한 규칙에 의해 숫자가 증가하게 되는데, 이 숫자가 저장소에 저장된 각각의 파일 버전임
 - 여러 개의 파일 중 하나만 변경하여 commit 하더라도 모든 파일의 버전은 올라가며 버전은 프로젝트 단위로 증가함
- trunk
 - 프로젝트에서 가장 중심이 되는 디렉토리로 모든 프로그램 개발 작업은 trunk 디렉토리에서 이루어짐
 - trunk 디렉토리 아래에는 바로 소스들의 파일과 디렉토리가 들어가게 됨
- branches
 - trunk 디렉토리에서 프로그램을 개발하다 보면 큰 프로젝트에서 또 다른 작은 분류로 빠져 따로 개발해야 할 경우가 생기는데, 이 때 branches 디렉토리 안에 또 다른 디렉토리를 두어 그 안에서 개발함
 - 브랜치는 대부분 주요 개발 브랜치를 컴파일러 오류나 버그로 방해하지 않고 새로운 기능을 테스트할 경우에 생성함
- tag
 - 태그 디렉토리는 프로그램을 개발하면서 정기적으로 릴리즈를 할 때 그때그때 발표한 소스를 따로 저장하는 공간임
 - 태깅은 기본적으로 개정판 번호(Revision Number)와 상관없이 각 파일에 라벨을 붙이는 것이며 이 작업은 작업 사본에 할 수도 있고 저장소 자체에 할 수도 있음

- hook
 - 후크는 새로운 개정판의 생성이나 버전 관리가 되지 않은 속성의 수정과 같은 저장소 이벤트에 의해 실행되는 프로그램임
 - 후크는 그 이벤트가 무엇인지, 어떤 목표로 작동하는지, 이벤트를 실행한 사람의 사용자 이름이 무엇인지 알려줌
- lock
 - 사용자가 작업 사본 파일을 수정할 독점적 권리를 요구하는 장치임
- merge
 - 병합은 하나의 브랜치에 대한 변경 사항을 다른 브랜치와 합쳐서 트렁크로 반영하는 것(반대의 경우도 가능)
 - 이때의 다른 브랜치 역시 트렁크일 수 있음

2. MantisBT

a. MantisBT 개요

- MantisBT는 2000년 11월에 공개된 웹 기반의 버그 추적 시스템이다. 시간이 지나면서 이 시스템은 엄청나게 인기를 얻게 되었고, 지금은 가장 인기 있는 오픈 소스 기반의 버그 추적 시스템 중의 하나가 되었다. MantisBT는 PHP로 개발되었으며, MySQL, MSSQL, PostgreSQL, DB2 등의 다중 데이터베이스를 지원한다. MantisBT는 PHP script와 위의 언급된 DBMS중 하나를 지원하는 어떤 OS에서도 실행이 가능하다.

b. MantisBT 설치

New installation	Upgrade
<ol style="list-style-type: none">1. 시스템 요구사항을 확인한다.2. MantisBT를 다운로드한다.3. 설치 스크립트를 실행한다.4. 시스템을 설정한다.5. Post-installation을 실행한다.	<ol style="list-style-type: none">1. MantisBT를 다운로드한다.2. 시스템을 백업한다.3. 유지를 위해 사이트를 닫는다.4. 업그레이드 스크립트를 실행한다.5. Post-installation을 실행한다.

c. MantisBT 사용법

- i. mantisbt.org에서 MantisBT를 다운로드를 한다.
- ii. 압축해제한다.
- iii. /usr/local/apache로 옮긴다.
- iv. <http://dev.viskit.co.kr:20080/Mantis/admin> 세팅 후 삭제
- v. Config_inc.php 파일 수정

```
# set these values to match your setup
$g_hostname = 
$g_db_username = 
$g_db_password = 
$g_database_name = 
$g_db_type = 

# --- email variables -----
$g_administrator_email = 
$g_webmaster_email = 

# the "From: " field in emails
$g_from_email = 

# the return address for bounced mail
$g_return_path_email = ' ' ;
```

vi. <http://dev.viskit.co.kr:20080/Mantis> 접속

vii. 관리자 계정으로 로그인



Login	
Username	<input type="text" value="admin"/>
Password	<input type="password" value="*****"/>
Save Login	<input type="checkbox"/>
<input type="button" value="Login"/>	

[[Signup for a new account](#)] [[Lost your password?](#)]

WARNING: Admin directory should be removed.

viii. 상단 메뉴 Manage 클릭



Logged in as: admin (어드민 - administrator) 2014-04-17 19:32 KST Project: [All Projects] [Switch] [OFF]

[Main](#) | [My View](#) | [View Issues](#) | [Report Issue](#) | [Change Log](#) | [Roadmap](#) | [Summary](#) | [Docs](#) | **Manage** | [Edit News](#) | [My Account](#) | [Logout](#) Issue #

Recently Visited: 0000002, 0000001

Unassigned [^] (1 - 2 / 2) 0000002 test [testtttt] - 2014-04-17 18:30 0000001 test [testtttt] - 2014-04-17 16:29	Reported by Me [^] (1 - 2 / 2) 0000002 test [testtttt] - 2014-04-17 18:30 0000001 test [testtttt] - 2014-04-17 16:29
Resolved [^] (1 - 1 / 1) 0000001 test [testtttt] - 2014-04-17 16:29	Recently Modified [^] (1 - 2 / 2) 0000002 test [testtttt] - 2014-04-17 18:30 0000001 test [testtttt] - 2014-04-17 16:29
Monitored by Me [^] (0 - 0 / 2)	

new feedback acknowledged confirmed assigned resolved closed



ix. 계정 테이블 위에 Create New Account 클릭

Mantis 1.1.6 [v]

Copyright © 2000 - 2008 Mantis Group
 tgbnhy02@gmail.com
 20 total queries executed.
 18 unique queries executed.

Username	Real Name	E-mail	Access Level	Enabled	Date Created	Last Visit
admin	어드민	tgbnhy02@gmail.com	administrator	X	2014-04-16 12:56	2014-04-17 19:32
administrator	관리자	tgbnhy02@gmail.com	administrator	X	2014-04-16 12:47	2014-04-16 13:00
user1	naver	tgbnhy0@naver.com	developer	X	2014-04-17 08:04	2014-04-17 08:04
user2	gmail	tgbnhy02@gmail.com	reporter	X	2014-04-17 08:07	2014-04-17 16:10

x. ID, e-mail, access level 작성 후 생성

2014-04-17 19:53 KST

Main | My View | View Issues | Report Issue | Change Log | Roadmap | Summary | Docs | Manage | Edit News | My Account | Logout

[Manage Users] [Manage Projects] [Manage Custom Fields] [Manage Global Profiles] [Manage Configuration]

Create New Account

Username:

Real Name:

E-mail:

Access Level:

Enabled:

Protected:

- xi. E-mail 접속 후, mantis 가입 메일에서 링크 클릭
- xii. PW 설정 후 update 유저 클릭

Your account information has been verified. The account confirmation message you have received is now invalid. You must set a password here to allow you to log in again.

My Account | Preferences | Profiles

Edit Account

Username: user3

Password:

Confirm Password:

E-mail: pmk3745@gmail.com

Real Name: user3

Access Level: developer

Project Access Level: developer

Assigned Projects: SMA [developer] (public)

xiii. Manage - Manage Projects - Create New Project 클릭



Logged in as: admin (어드민 - administrator) 2014-04-17 19:57 KST Project: All Projects Switch

[Main](#) | [My View](#) | [View Issues](#) | [Report Issue](#) | [Change Log](#) | [Roadmap](#) | [Summary](#) | [Docs](#) | [Manage](#) | [Edit News](#) | [My Account](#) | [Logout](#) Issue # Jump

[[Manage Users](#)] [[Manage Projects](#)] [[Manage Custom Fields](#)] [[Manage Global Profiles](#)] [[Manage Configuration](#)]

Projects <input type="button" value="Create New Project"/>				
Name ▲	Status	Enabled	View Status	Description
testtttt	development	X	public	

xiv. Project 정보 입력 후 Add project

[Main](#) | [My View](#) | [View Issues](#) | [Report Issue](#) | [Change Log](#) | [Roadmap](#) | [Summary](#) | [Docs](#) | [Manage](#) | [Edit News](#) | [My Account](#) | [Logout](#)

[[Manage Users](#)] [[Manage Projects](#)] [[Manage Custom Fields](#)] [[Manage Global Profiles](#)] [[Manage Configuration](#)]

Add Project

Project Name SMA

Status development

View Status public

Upload File Path

Description Good Luck!!

xv. 다음 프로젝트에서 사용자 추가

Add user to project

Username	Access Level	
user1	developer	<input type="button" value="Add User"/>
user2		
user3		

testtttt

xvi. Report Issues에서 Project 선택

Select Project

Choose Project SMA

Make Default

Mantis 1.1.0 [6.2]
 Copyright © 2000 - 2008 Mantis Group
 mantis@sourcefire.com
 13 total queries executed.
 11 minisession events

xvii. 세부 정보 입력 후 이슈 등록

Enter Report Details [\[Advanced Report \]](#)

Category	(select) ▾
Reproducibility	have not tried ▾
Severity	minor ▾
Priority	normal ▾
Summary	Hello
Description	Good luck!!
Additional Information	
Upload File (Max. size: 2,000k)	파일 선택 선택된 파일 없음
View Status	<input checked="" type="radio"/> public <input type="radio"/> private
Report Stay	<input type="checkbox"/> (check to report more issues)

* required

xviii. View issuesd에서 이슈를 확인한다.

Viewing Issue Simple Details [\[>> \]](#) [\[View Advanced \]](#) [\[Issue History \]](#) [\[Print \]](#)

ID	Category	Severity	Reproducibility	Date Submitted	Last Update
0000006	[SMA]	minor	have not tried	2014-04-18 02:00	2014-04-18 02:00
Reporter	admin	View Status	public		
Assigned To					
Priority	normal	Resolution	open		
Status	new				
Summary	0000006: Hello				
Description	Good luck!!				
Additional Information					
Tags	No tags attached.				
Attach Tags	(Separate by ",") <input type="text"/> Existing tags ▾ <input type="button" value="Attach"/>				
Attached Files					

[Myself] ▾
 feedback ▾

3. JFeature

a. JFeature 개요

- 요구사항들을 기록하고 이들을 JUnit Test Case와 매치함으로써 프로세스를 단순화 시키는데 사용하는 도구
- 요구사항을 기록하고 보고서를 생성한다.

- JFeature 특징

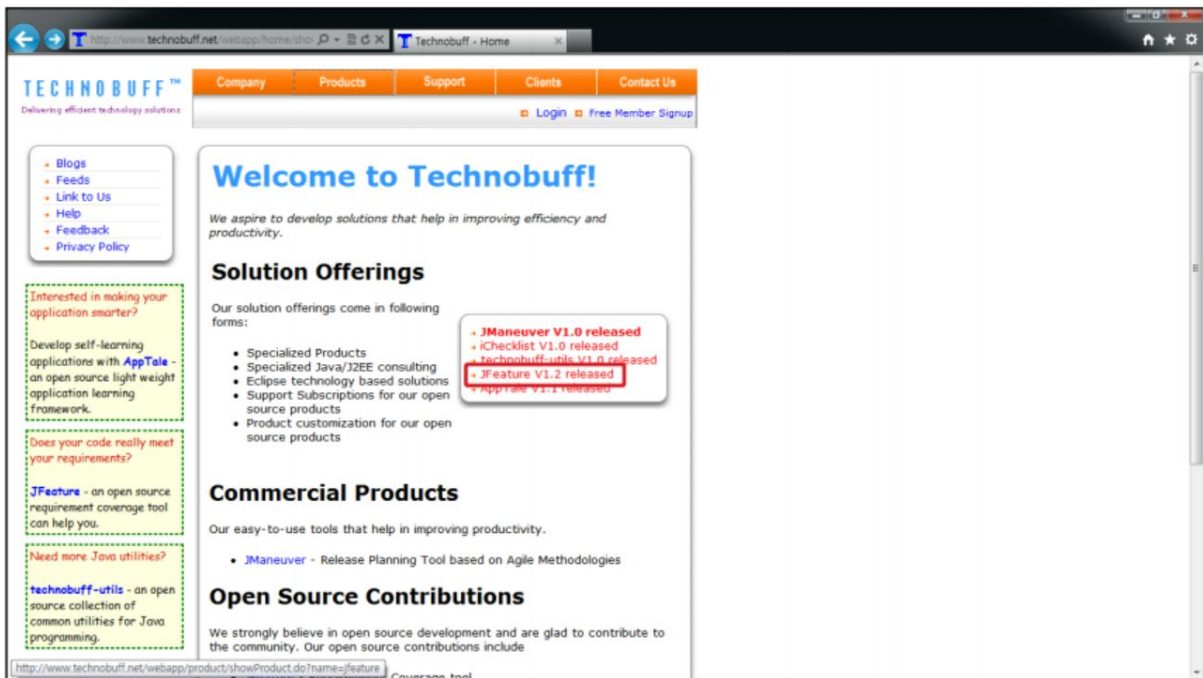
- JUnit의 Test case 메소드와 통합
- 요구사항의 수정사항이 생겼을 때, 즉시 뷰를 제공하므로 빠른 편집이 가능
- 요구사항이 반영여부를 Coverage형태로 쉽게 확인

- JFeature 기능

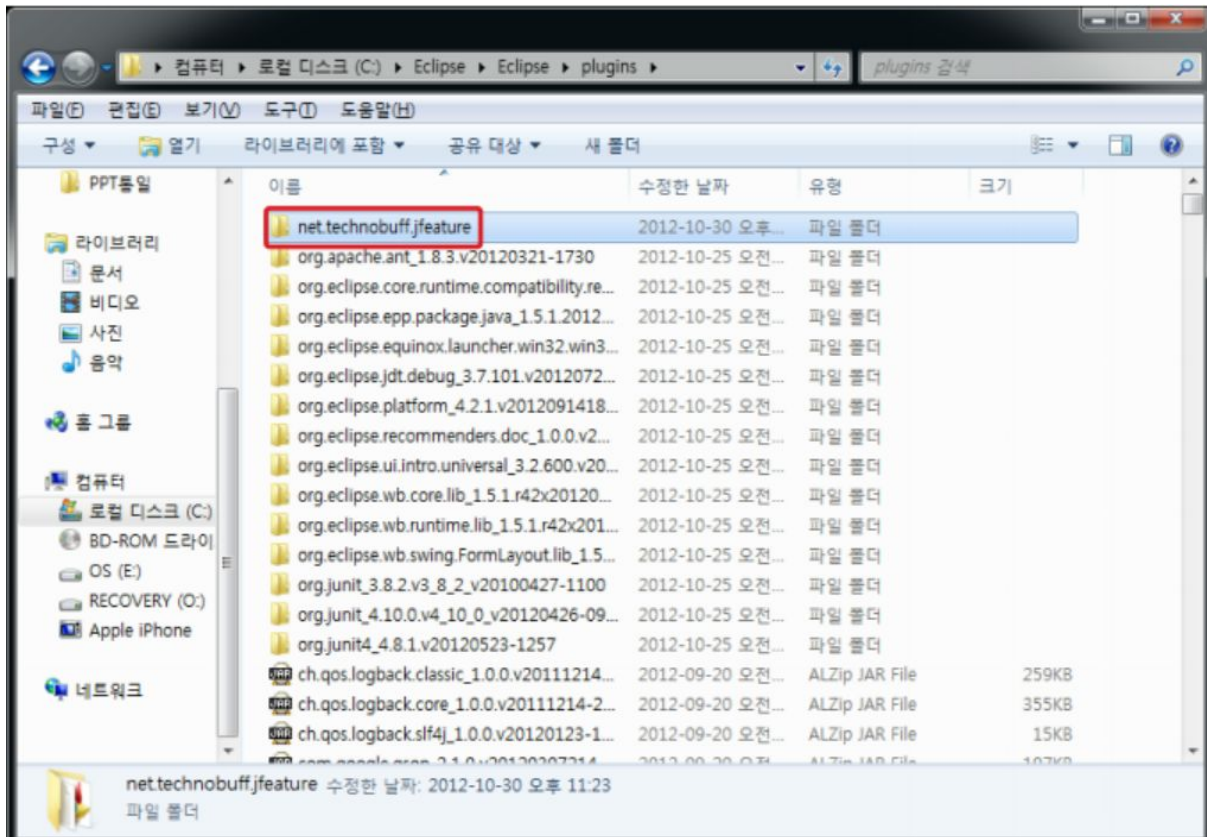
- 요구사항 반영여부 확인
- 타 플러그인 환경 지원
- 코드 추적
- IDE 내 별도 편집기 제공
- 요구사항 스펙 반출 (Commercial Version)
- 리포팅 (Commercial Version)
- Round-trip engineering 지원

b. JFeature 사용법

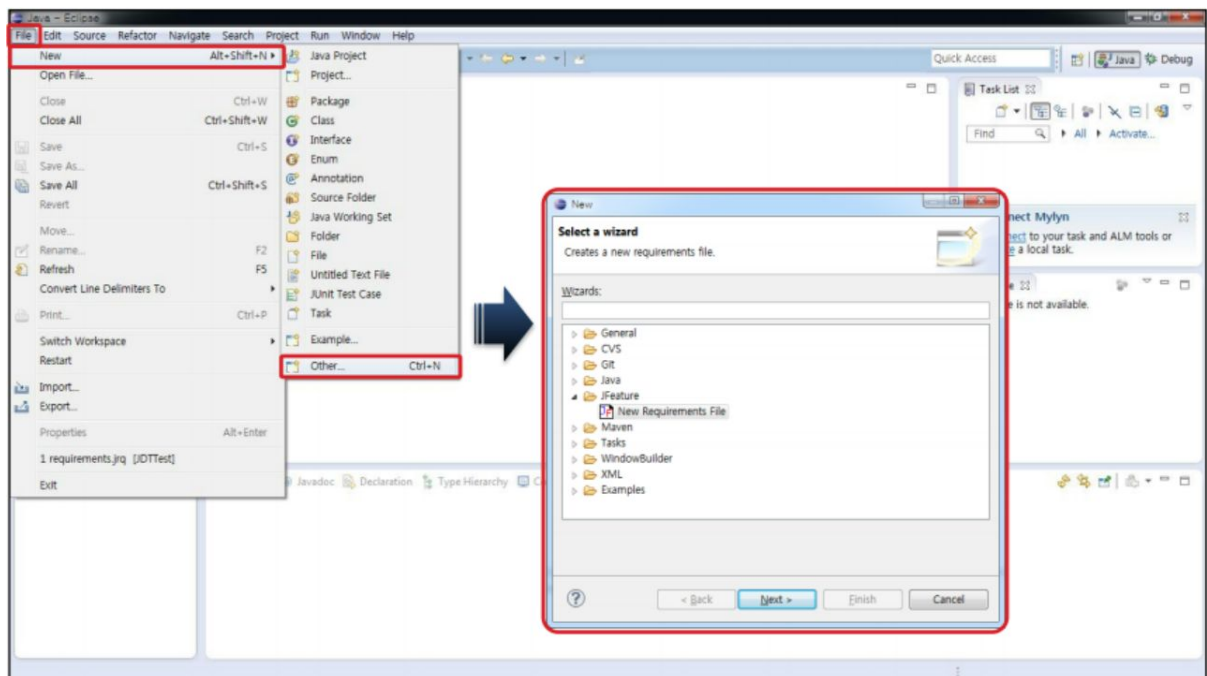
- <http://www.technobuff.net/>에서 JFeature를 다운받는다.



- 압축을 풀고, Eclipse의 plugins 폴더에 net.technobuff.jfeature 폴더를 붙여넣는다.



- Eclipse를 실행하여 JFeature가 추가되었는지 확인한다.



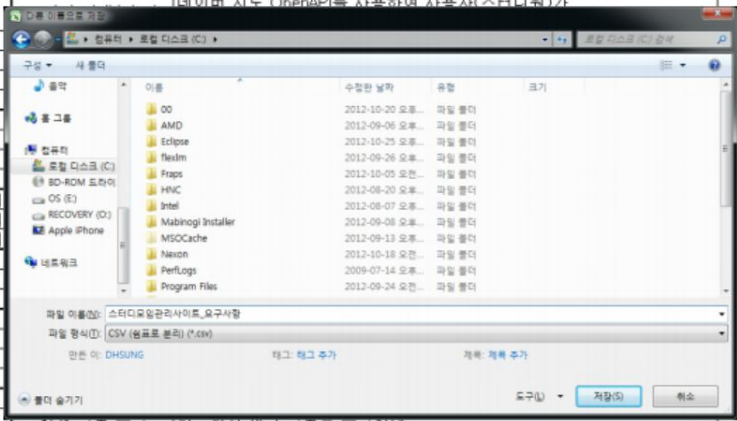
- JFeature 사용 예제

1. 엑셀에 요구사항을 입력하여 정리한다.

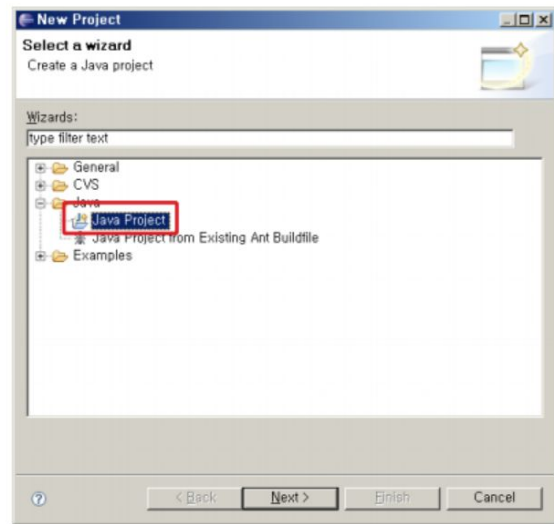
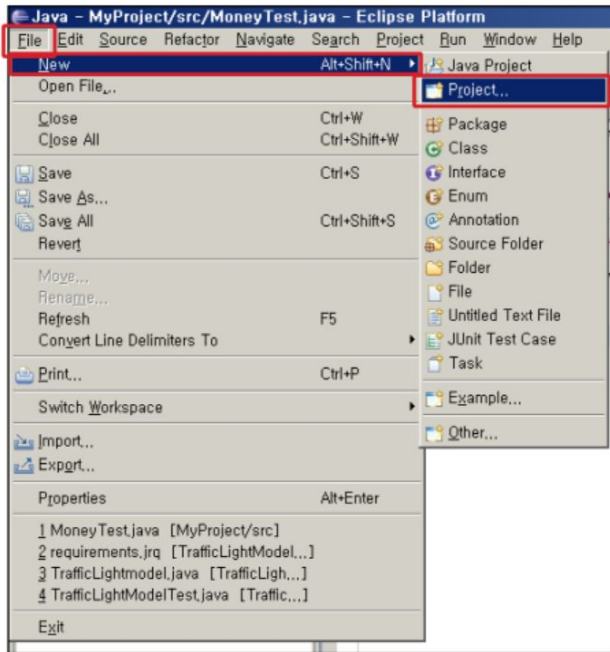
요구사항 ID	상위분류	하위분류	요구사항 설명	우선순위	비고
REQ_0100_001	스터디 개설 관리	스터디 개설	사용자(스터디장)가 개설할 스터디의 정보를 등록한다.	1	
REQ_0100_002	스터디 개설 관리	스터디 등록	모집일시 이전에 모집인원 충족시 스터디방 생성 통보를 하고 모집일시까 지 모집인원 미달시에 폐쇄 및 연기 통보 메시지를 전송한다.	2	
REQ_0100_003	스터디 개설 관리	스터디 수정	사용자(스터디장)가 개설할 스터디의 정보를 수정한다.	2	
REQ_0100_004	스터디 개설 관리	스터디 삭제	사용자(스터디장)가 개설할 스터디의 정보를 삭제한다.	2	
REQ_0100_005	스터디 개설 관리	스터디 조회	사용자(스터디장)가 개설할 스터디의 정보를 조회한다.	2	
REQ_0200_001	개설된 스터디 이용	스터디 검색(텍스트)	사용자(스터디원)가 가입을 할 스터디를 검색한다.	2	
REQ_0200_002	개설된 스터디 이용	스터디 검색(지도)	네이버 지도 OpenAPI를 사용하여 사용자(스터디원)가 해당지역안에 가입을 할 스터디를 검색한다.	2	
REQ_0200_003	개설된 스터디 이용	스터디 조회	사용자(스터디원)가 개설할 스터디의 정보를 조회한다.	2	
REQ_0200_004	개설된 스터디 이용	스터디 가입	사용자(스터디원)가 개설할 스터디에 가입한다.	1	
REQ_0200_005	개설된 스터디 이용	스터디 탈퇴	사용자(스터디원)가 가입한 스터디에 탈퇴한다.	2	
REQ_0200_006	개설된 스터디 이용	스터디 추천	사용자(스터디원)가 가입한 스터디를 추천한다.	2	
REQ_0300_001	스터디방 일정관리	일정 등록	스터디의 일정항목을 등록한다.	2	
REQ_0300_002	스터디방 일정관리	일정 수정	스터디의 일정항목을 수정한다.	2	
REQ_0300_003	스터디방 일정관리	일정 삭제	스터디의 등록된 일정을 삭제한다.	2	
REQ_0300_004	스터디방 일정관리	일정 통보	스터디의 등록된 일정이 도달시 스터디원에게 메시지를 통보한다.	2	
REQ_0400_001	스터디방 게시판 관리	게시판 등록	스터디의 게시판을 등록한다.	2	
REQ_0400_002	스터디방 게시판 관리	게시판 수정	스터디의 게시판을 수정한다.	2	
REQ_0400_003	스터디방 게시판 관리	게시판 삭제	스터디의 게시판을 삭제한다.	2	
REQ_0500_001	스터디방 회원관리	회원 조회	합성물이 저조한 사용자(스터디원)를 감퇴한다.	2	
REQ_0500_002	스터디방 회원관리	회원 조회	사용자(스터디장)은 스터디원을 조회한다.	2	
REQ_0500_003	스터디방 회원관리	과제 조회	사용자(스터디장)은 스터디원이 올린 과제를 확인한다.	2	
REQ_0600_001	스터디방 이용	회원 조회	사용자가 다른 회원의 목록을 조회한다.	3	
REQ_0600_002	스터디방 이용	출석 확인	사용자가 출석을 확인한다.	3	
REQ_0600_003	스터디방 이용	과제 등록	사용자가 과제를 제출한다.	3	
REQ_0600_004	스터디방 이용	과제 조회	사용자가 등록된 과제를 확인한다.	3	
REQ_0700_001	회원 관리	회원 가입	사용자가 서비스 사용을 위해 개인정보를 등록한다.	1	
REQ_0700_002	회원 관리	회원 정보 수정	사용자가 개인 정보를 수정한다.	1	
REQ_0700_003	회원 관리	ID / 비밀번호 찾기	사용자가 가입시 등록한 이메일로 ID와 비밀번호를 전송한다.	1	
REQ_0700_004	회원 관리	회원 탈퇴	사용자가 서비스 사용권한을 잃는다.	1	
REQ_0800_001	로그인/아웃	로그인	사용자가 개인 서비스 사용권한을 얻는다.	1	
REQ_0800_002	로그인/아웃	로그아웃	사용자가 개인 서비스 사용권한을 잃는다.	1	

2. CSV 파일로 저장한다.

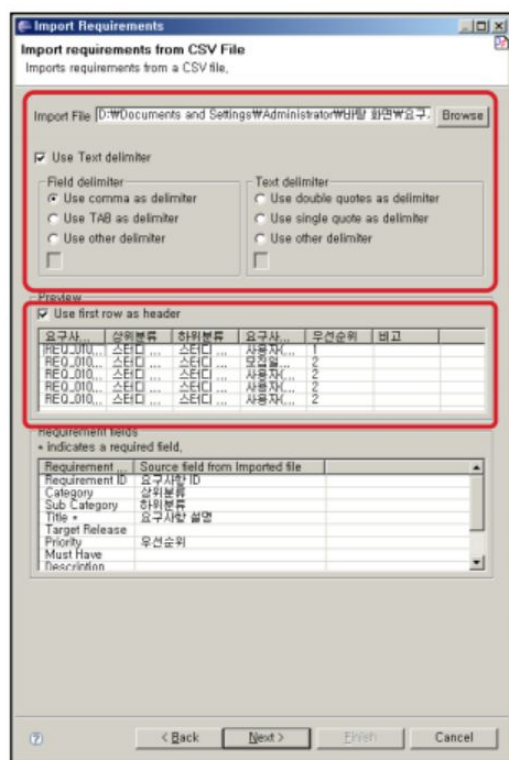
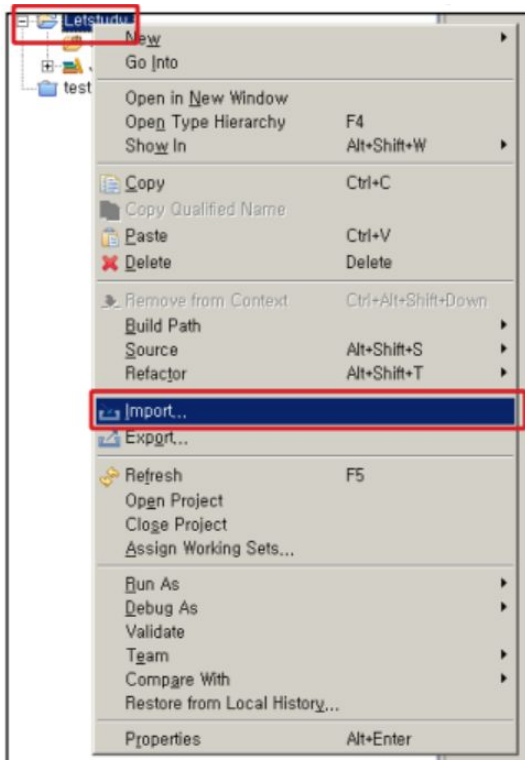
요구사항 ID	상위분류	하위분류	요구사항 설명	우선순위	비고
REQ_0100_001	스터디 개설 관리	스터디 개설	사용자(스터디장)가 개설할 스터디의 정보를 등록한다.	1	
REQ_0100_002	스터디 개설 관리	스터디 등록	모집일시 이전에 모집인원 충족시 스터디방 생성 통보를 하고 모집일시까 지 모집인원 미달시에 폐쇄 및 연기 통보 메시지를 전송한다.	2	
REQ_0100_003	스터디 개설 관리	스터디 수정	사용자(스터디장)가 개설할 스터디의 정보를 수정한다.	2	
REQ_0100_004	스터디 개설 관리	스터디 삭제	사용자(스터디장)가 개설할 스터디의 정보를 삭제한다.	2	
REQ_0100_005	스터디 개설 관리	스터디 조회	사용자(스터디장)가 개설할 스터디의 정보를 조회한다.	2	
REQ_0200_001	개설된 스터디 이용	스터디 검색(텍스트)	사용자(스터디원)가 가입을 할 스터디를 검색한다.	2	
REQ_0200_002	개설된 스터디 이용	스터디 검색(지도)	네이버 지도 OpenAPI를 사용하여 사용자(스터디원)가	2	
REQ_0200_003	개설된 스터디 이용	스터디 조회	사용자(스터디원)가 개설할 스터디의 정보를 조회한다.	2	
REQ_0200_004	개설된 스터디 이용	스터디 가입	사용자(스터디원)가 개설할 스터디에 가입한다.	1	
REQ_0200_005	개설된 스터디 이용	스터디 탈퇴	사용자(스터디원)가 가입한 스터디에 탈퇴한다.	2	
REQ_0200_006	개설된 스터디 이용	스터디 추천	사용자(스터디원)가 가입한 스터디를 추천한다.	2	
REQ_0300_001	스터디방 일정관리	일정 등록	스터디의 일정항목을 등록한다.	2	
REQ_0300_002	스터디방 일정관리	일정 수정	스터디의 일정항목을 수정한다.	2	
REQ_0300_003	스터디방 일정관리	일정 삭제	스터디의 등록된 일정을 삭제한다.	2	
REQ_0300_004	스터디방 일정관리	일정 통보	스터디의 등록된 일정이 도달시 스터디원에게 메시지를 통보한다.	2	
REQ_0400_001	스터디방 게시판 관리	게시판 등록	스터디의 게시판을 등록한다.	2	
REQ_0400_002	스터디방 게시판 관리	게시판 수정	스터디의 게시판을 수정한다.	2	
REQ_0400_003	스터디방 게시판 관리	게시판 삭제	스터디의 게시판을 삭제한다.	2	
REQ_0500_001	스터디방 회원관리	회원 조회	합성물이 저조한 사용자(스터디원)를 감퇴한다.	2	
REQ_0500_002	스터디방 회원관리	회원 조회	사용자(스터디장)은 스터디원을 조회한다.	2	
REQ_0500_003	스터디방 회원관리	과제 조회	사용자(스터디장)은 스터디원이 올린 과제를 확인한다.	2	
REQ_0600_001	스터디방 이용	회원 조회	사용자가 다른 회원의 목록을 조회한다.	3	
REQ_0600_002	스터디방 이용	출석 확인	사용자가 출석을 확인한다.	3	
REQ_0600_003	스터디방 이용	과제 등록	사용자가 과제를 제출한다.	3	
REQ_0600_004	스터디방 이용	과제 조회	사용자가 등록된 과제를 확인한다.	3	
REQ_0700_001	회원 관리	회원 가입	사용자가 서비스 사용을 위해 개인정보를 등록한다.	1	
REQ_0700_002	회원 관리	회원 정보 수정	사용자가 개인 정보를 수정한다.	1	
REQ_0700_003	회원 관리	ID / 비밀번호 찾기	사용자가 가입시 등록한 이메일로 ID와 비밀번호를 전송한다.	1	
REQ_0700_004	회원 관리	회원 탈퇴	사용자가 서비스 사용권한을 잃는다.	1	
REQ_0800_001	로그인/아웃	로그인	사용자가 개인 서비스 사용권한을 얻는다.	1	
REQ_0800_002	로그인/아웃	로그아웃	사용자가 개인 서비스 사용권한을 잃는다.	1	



3. 새로운 자바 프로젝트를 생성한다.



4. CSV 파일을 Import 한다.



5. CSV를 Import한 결과가 나온다.

Requirement ID	Category	Sub Category	Title	Target ...	Priority	Must Have
REQ_0100_001	스터디	스터디	사용자(스터디장)가 개설할 스터디의 정원을 등록한다.		1	No
REQ_0100_002	스터디	스터디	모집 시 이력에 접근이 될 수 있는지 스터디장을 생성 등록을 하고 모집일시까...		2	No
REQ_0100_003	스터디	스터디	사용자(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0100_004	스터디	스터디	사용자(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0100_005	스터디	스터디	사용자(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0200_001	개설	스터디	사용자(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0200_002	개설	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0200_003	개설	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0200_004	개설	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0200_005	개설	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0200_006	개설	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0300_001	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0300_002	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0300_003	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0300_004	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0400_001	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0400_002	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0400_003	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0500_001	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0500_002	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0500_003	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0600_001	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0600_002	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0600_003	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0600_004	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0700_001	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0700_002	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0700_003	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0700_004	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0800_001	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No
REQ_0800_002	스터디	스터디	스터디(스터디장)가 개설할 스터디의 정원을 생성 등록한다.		2	No

6. 회원 로그인에 대한 Test case를 작성한다.

```

import java.util.regex.Pattern;

import junit.framework.*;

public class LoginTest extends TestCase
{
    private MemVO loginVO;

    public void setUp()
    {
        loginVO = new MemVO("admin", "1234");
    }

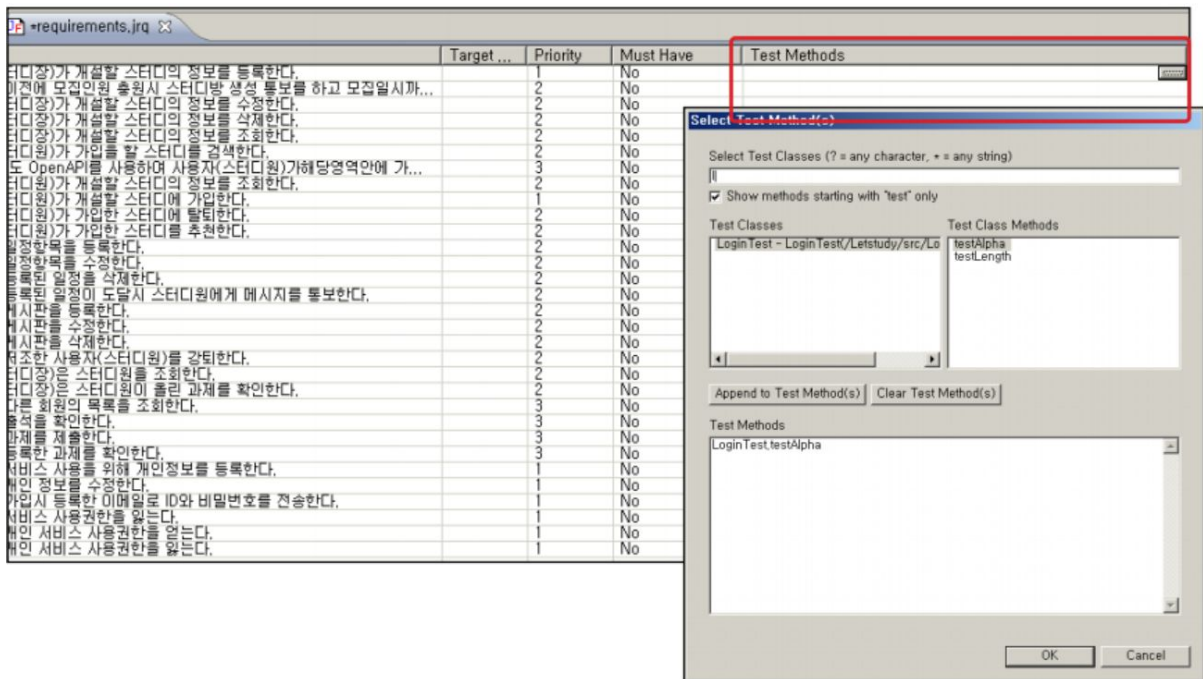
    public void testAlpha()
    {
        boolean check = false;
        Pattern pstr=Pattern.compile("[a-zA-Z]+");
        if(pstr.matcher(loginVO.getId()).matches() && pstr.matcher(loginVO.getPw()).matches())
        {
            check = true;
        }

        assertEquals(true, check);
    }

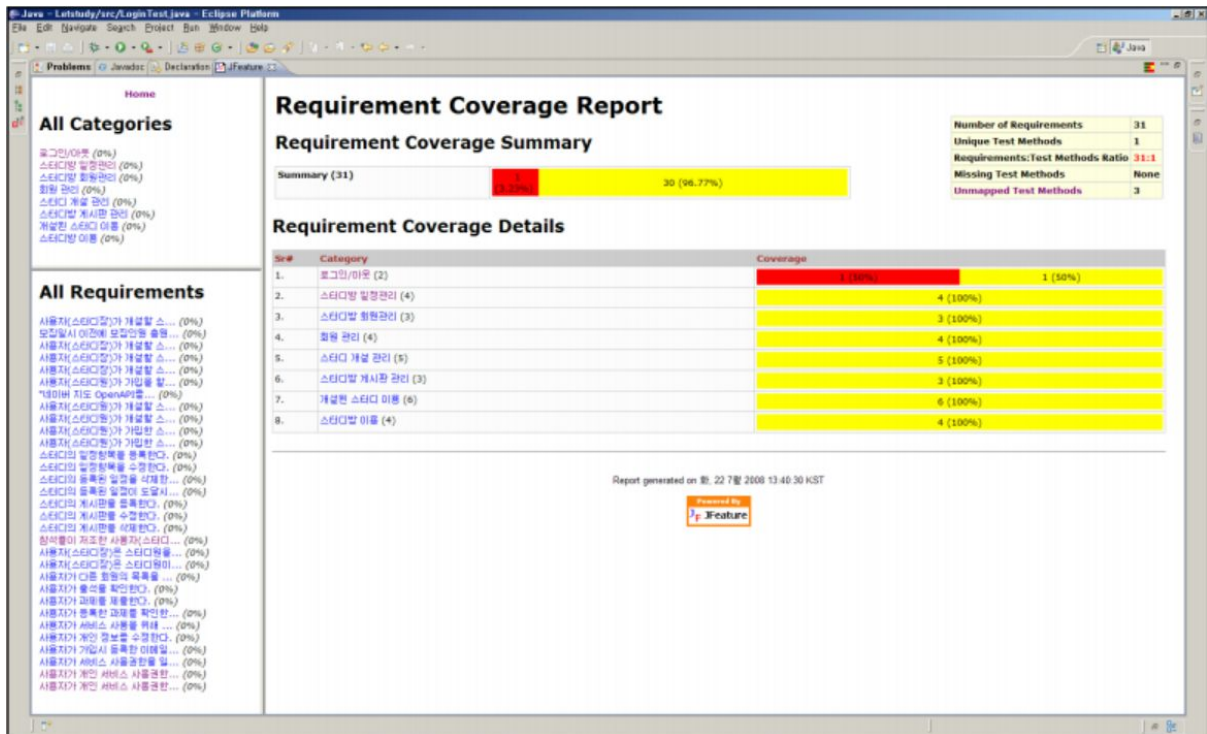
    public void testLength()
    {
        boolean check = false;
        Pattern plen=Pattern.compile("(5,8)");
        if(plen.matcher(loginVO.getId()).matches() && plen.matcher(loginVO.getPw()).matches())
        {
            check = true;
        }
    }
}

```

7. 요구사항과 매칭되는 Test 메소드를 연결한다.



8. 보고서 결과가 나온다.



4. JUnit

a. JUnit 개요

- JUnit은 자바 프로그래밍 언어로 짜여진 유닛 테스트 프레임워크로, 테스트 주도 개발(TDD, Test-Driven Development)에서 중요하게 여겨진다. 자동화 테스트를 위한 전용 프레임워크의 필요성을 증히 여긴 에릭 감마와 켄트 벡이 1997년에 제작하였다.
- JUnit 개념

JUnit 개념	역할												
Assert	<p>테스트하려는 조건을 명시. assert 메소드는 조건이 만족 되면 그대로 진행하고 만족 되지 않으면 예외를 던진다. JUnit의 Assert 클래스에 정의된 assert 메소드는 다음과 같다.</p> <table border="1"> <thead> <tr> <th>assertXXX 메소드</th> <th>사용 목적</th> </tr> </thead> <tbody> <tr> <td>assertArrayEquals("message", A, B)</td> <td>배열 A와 B가 일치함을 확인</td> </tr> <tr> <td>assertEquals("message", A, B)</td> <td>객체 A와 B의 값이 일치함을 확인. B를 파라미터로 하여 A의 equals() 메서드를 호출함</td> </tr> <tr> <td>assertSame("message", A, B)</td> <td>객체 A와 B가 같은 객체임을 확인.</td> </tr> <tr> <td>assertTrue("message", A)</td> <td>조건 A가 참임을 확인</td> </tr> <tr> <td>assertNotNull("message", A)</td> <td>객체 A가 null이 아님을 확인</td> </tr> </tbody> </table>	assertXXX 메소드	사용 목적	assertArrayEquals("message", A, B)	배열 A와 B가 일치함을 확인	assertEquals("message", A, B)	객체 A와 B의 값이 일치함을 확인. B를 파라미터로 하여 A의 equals() 메서드를 호출함	assertSame("message", A, B)	객체 A와 B가 같은 객체임을 확인.	assertTrue("message", A)	조건 A가 참임을 확인	assertNotNull("message", A)	객체 A가 null이 아님을 확인
assertXXX 메소드	사용 목적												
assertArrayEquals("message", A, B)	배열 A와 B가 일치함을 확인												
assertEquals("message", A, B)	객체 A와 B의 값이 일치함을 확인. B를 파라미터로 하여 A의 equals() 메서드를 호출함												
assertSame("message", A, B)	객체 A와 B가 같은 객체임을 확인.												
assertTrue("message", A)	조건 A가 참임을 확인												
assertNotNull("message", A)	객체 A가 null이 아님을 확인												
Test	@Test 주석이 부여된 메소드. JUnit은 먼저 메소드를 포함하는 클래스를 만들고, 주석 처리된 메소드를 찾아 호출한다.												
Test Class	@Test 메소드를 포함한 클래스이다.												
Suite	여러 테스트 클래스를 하나로 묶는 수단을 제공한다.												
Runner	테스트를 실행시킨다.												

b. JUnit 사용법

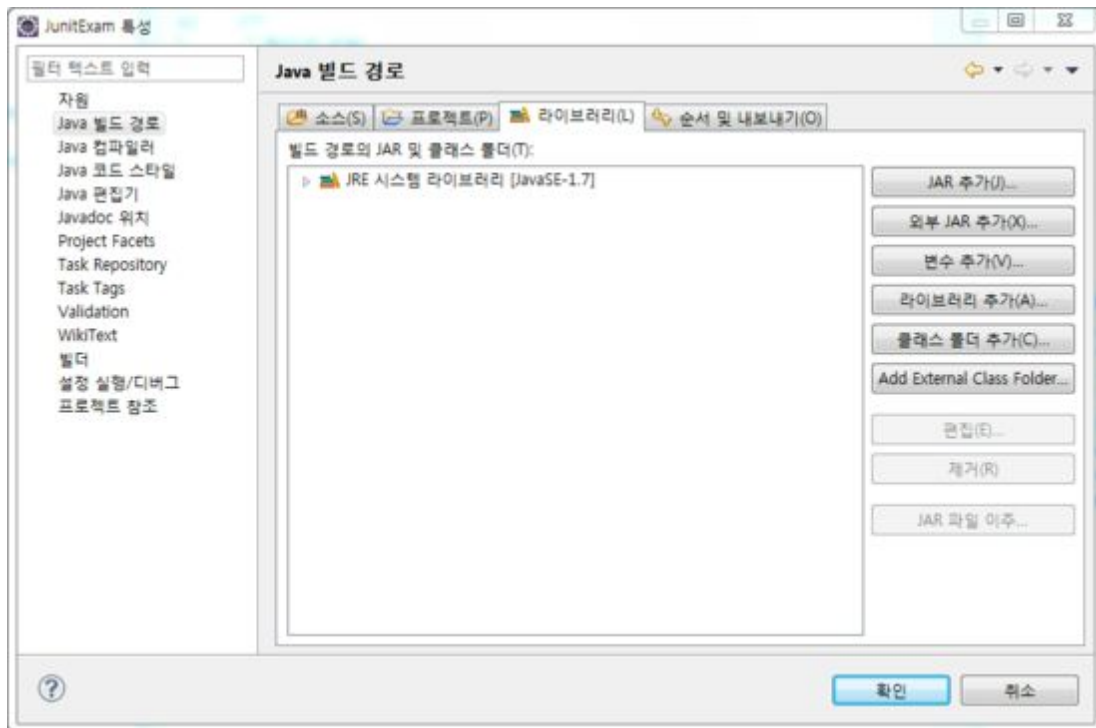
- <http://www.junit.org>에서 junit.jar을 다운로드 받아서 사용할 수 있다.

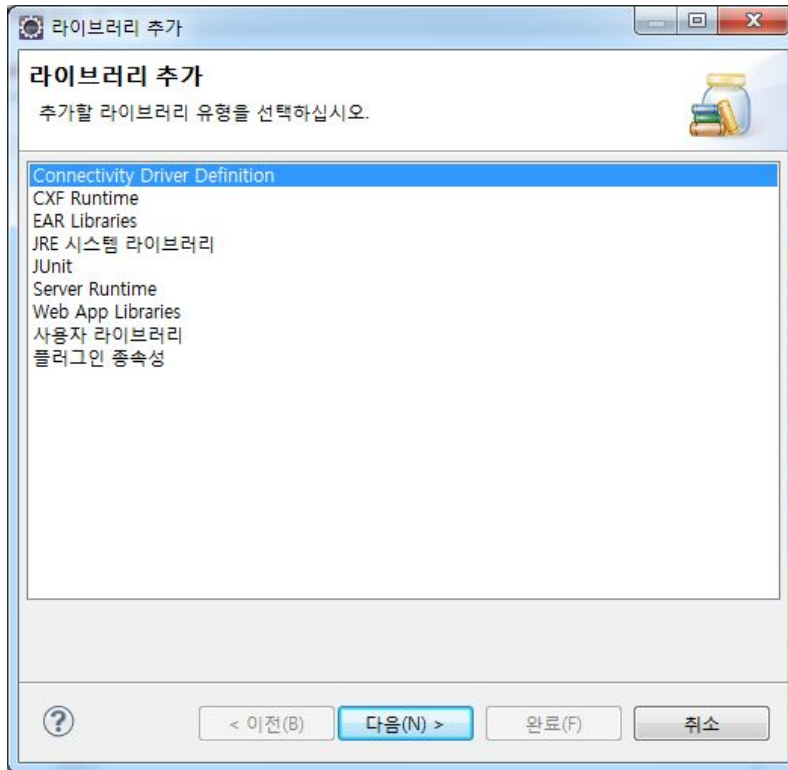
- Eclipse IDE를 사용할 경우의 사용예

1. 자바 프로젝트를 만들고 main package를 만들어서 그 아래 예제 클래스 JunitExamMain.java를 만든다.

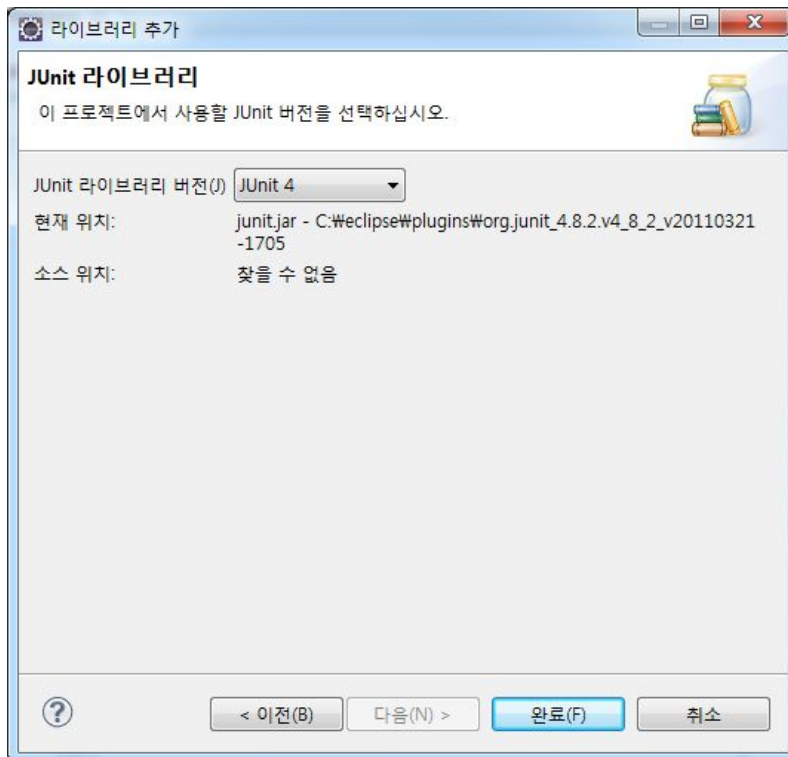


2. 프로젝트 특성에서 라이브러리 추가를 선택한다.





3. JUnit을 추가한다.



4. JunitExamMain.java에 예제 메소드 add를 작성한다.

```
package main;

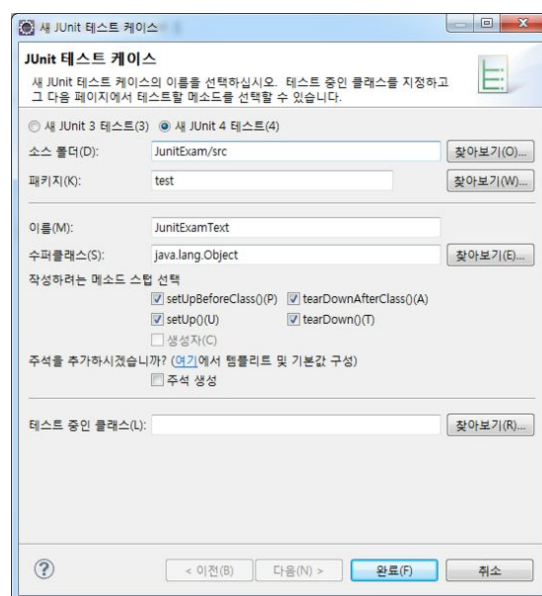
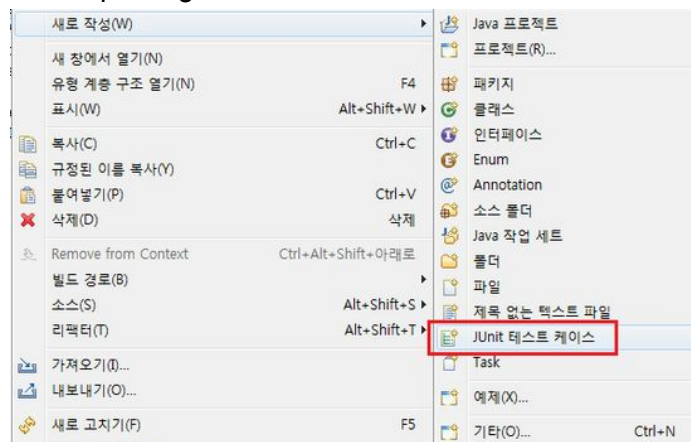
public class JunitExamMain {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

    public int add(int a, int b){
        return a+b;
    }
}
```

5. test package를 만들어 그 밑에 JUnit Test Case 파일을 만든다.



6. JunitExamTest.java 클래스를 작성한다.

```
package test;

import static org.junit.Assert.*;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

import main.JunitExamMain;

public class JunitExamTest {
    JunitExamMain jem;

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
        System.out.println("setUpBeforeClass()");
    }

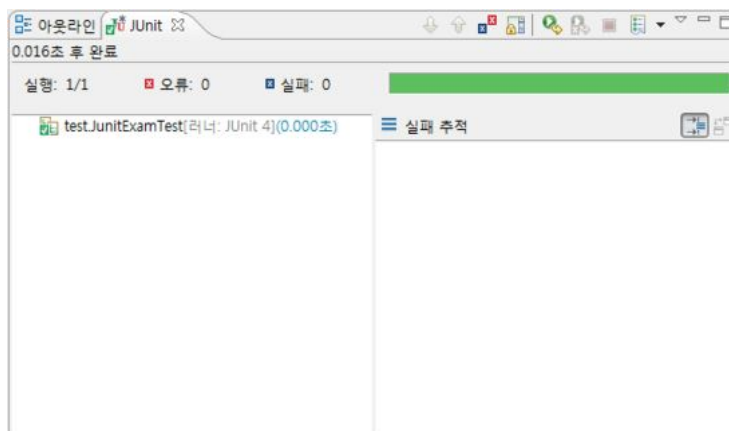
    @AfterClass
    public static void tearDownAfterClass() throws Exception {
        System.out.println("tearDownAfterClass()");
    }

    @Before
    public void setUp() throws Exception {
        System.out.println("setUp()");
    }

    @After
    public void tearDown() throws Exception {
        System.out.println("tearDown()");
    }

    @Test
    public void test() {
        jem = new JunitExamMain();
        assertEquals(3, jem.add(1, 2));
    }
}
```

7. 실행 도구 -> JUnit test를 실행한다.



- 테스트의 준비와 실행 정리

- setUp()
 - 테스트 대상 클래스의 메소드 구조와 상관 없이 테스트 메소드가 실행되기 전에 수행된다.
 - 코드 내에서 사용할 리소스를 초기화 할 때 이용한다.
 - (테스트 대상 클래스의 객체를 생성, 네트워크 연결 시작, DB에 연결)
- testXXX()
 - XXX는 테스트 대상 클래스의 메소드 이름이다.
 - 모든 테스트 대상 메소드 이름은 test로 시작해야 한다.
 - 이 메소드의 내에 assertXXX()를 이용하여 테스트 결과를 확인하는 작업을 수행한다.
- tearDown()
 - 메소드는 각각의 테스트 메소드들이 실행되고 난 다음에 호출된다.
 - 사용한 리소스를 정리할 때 이용한다.
 - (테스트 대상 클래스의 객체를 제거, 네트워크 연결 종료, DB 연결 종료)

- 테스트의 준비와 실행 정리 (주석 기반)

- 메소드에 부여되는 Annotation (해당 메소드가 반드시 public이어야 한다)
 - @Before
 - 이 Annotation이 부여된 메소드는 @Test 메소드가 호출되기 직전에 실행된다.
 - @Test
 - 실제 테스트를 할 대상 메소드에 작성한다.
 - @After
 - 이 Annotation이 부여된 메소드는 @Test 메소드가 호출된 후에 실행된다.
 - @Ignore
 - 해당 메소드를 실행하지 않도록 한다.

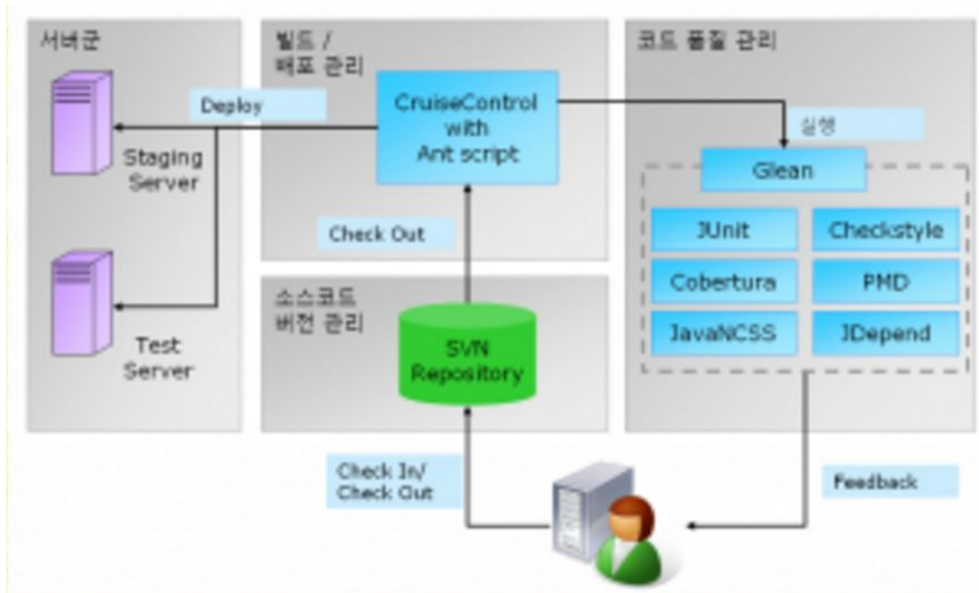
※ 원한다면 여러 개의 @Before와 @After method를 정의할 수도 있지만, 이들 사이의 실행 순서를 정하는 방법은 없다.

- 메소드에 부여되는 클래스 대상 Annotation (해당 메소드가 반드시 public이고 static이어야 한다.)
 - @BeforeClass
 - 클래스 안에 정의된 모든 @Test method들이 수행하기 전 한번만 호출
 - @AfterClass
 - 클래스 안에 정의된 모든 @Test method들이 수행된 후 한번만 수행

5. CTIP

a. CTIP 개요

- CTIP은 CI 개념을 Java 기반의 개발 프로젝트에 쉽게 적용하기 위한 Platform이다. CTIP이 제공하는 기능은 다음과 같다.
 - CI 서버를 통한 지속적 통합 및 빌드
 - 품질 도구들을 통한 코드 품질 검토(테스트 및 정적 분석)
 - 빌드 결과의 배포 및 관련자에게 통보
- CI (Continuous Integration)
 - 소프트웨어 개발에서 Build의 프로세스를 지속적으로 수행하는 것
 - 지속적으로 개발된 unit코드에 대한 integration 작업을 수행하여, 항상 최신의 상태를 유지한다.
 - Agile 개발 방법론/프로세스 중의 하나인 XP(eXtream Programming)의 Practice 중 하나
 - 지속적인 통합은 팀의 구성원들이 자신들의 작업한 내용을 자주 통합하는 개발 지침을 말한다.
- 다음은 이를 도식화 한 CTIP의 전체적인 구조이다.



- CTIP은 크게 코드 품질 관리 영역, 빌드 및 배포 관리 영역, 소스코드 버전 관리 영역, 그리고 대상 서버군으로 구성되어 있다. 다음은 각 영역에 대한 설명이다.
 - 소스코드 버전 관리: CVS, SVN 등의 소스코드 버전 관리 시스템을 사용하여 프로젝트 전체 소스코드의 일관성을 유지한다. 개발자는 버전 관리 시스템의 repository로부터 최신 소스를 checkout하고, 작업내용을 check in 한다.
 - 빌드 및 배포 관리: CI 서버를 통해 지속적으로 빌드를 수행하고, 대상 서버에 대한 배포작업을 수행한다. 빌드 주기는 정해진 시점에 수행하거나 (nightly build가 이에 해당한다), 버전 관리 시스템의 repository에 변경이 있을 경우 즉시 수행될 수 있다.
 - 코드 품질 관리: 오픈소스 코드 검토 도구를 활용하여 코드 품질을 확인하고, 결과를 개발자에게 통보한다. 다양한 오픈소스 및 상용 품질 관리 도구가 개발되어 있으므로 필요에 따라 선택하여 적용할 수 있다.
 - 서버군: 운영 서버 배포를 위한 스테이징 서버, 테스트 실행을 위한 테스트 서버 등을 운영한다.
- CTIP의 각 영역에서는 적절한 도구를 선정하여 영역에서 필요한 기능을 제공해야 한다. 다양한 오픈소스 및 상용 도구들이 개발되어 있기 때문에 특별한 기능이 필요하지 않은 경우, 굳이 직접 이러한 도구를 개발할 필요는 없다. 도구는 도구 자체의 기능, 프로젝트 특성, 팀의 도구에 대한 익숙함 등을 고려하여 선정한다.

- CTIP의 예

