

# Introduction to

# SVN, Junit, Jfeature, Mantis, CTIP

Project Team

T4

Date

2016-04-07

---

Team Information

201411258 강태준

201411265 김서우

201411321 홍유리

# Contents

1. SVN 이란?

2. Mantis 이란?

3. JFeatures 이란?

4. JUnit 이란?

5. CTIP 개론

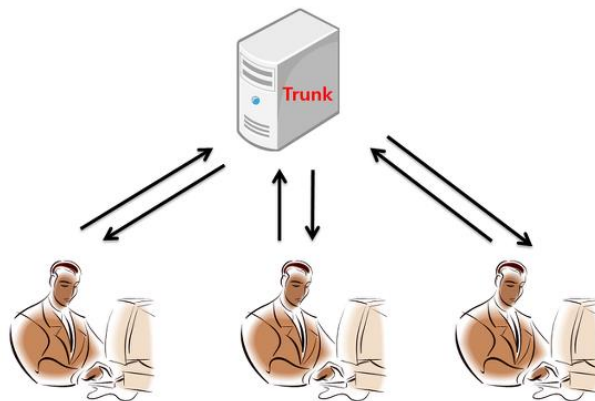
# 1. SVN 이란?

## 1) 개요

자유 소프트웨어 버전 관리 시스템으로 제한이 있던 CVS 를 대체하기 위해 개발된 것으로, 여러 개발자가 하나의 프로젝트를 진행함에 있어 통합적인 소스 관리를 편리하게 하기 위한 툴로 사용된다. 서버버전, 소스 공유 프로그램, 프로젝트 관리 프로그램, 코드 관리 및 공유 프로그램, 코드 저장소 프로그램 등 다양하게 불린다. 처음 파일을 저장시킬 때는 파일 원본 그대로 저장을 시키고 그 다음 부터는 실제 파일이 아닌 처음 파일과이 차이점을 저장시키는 시스템이다.

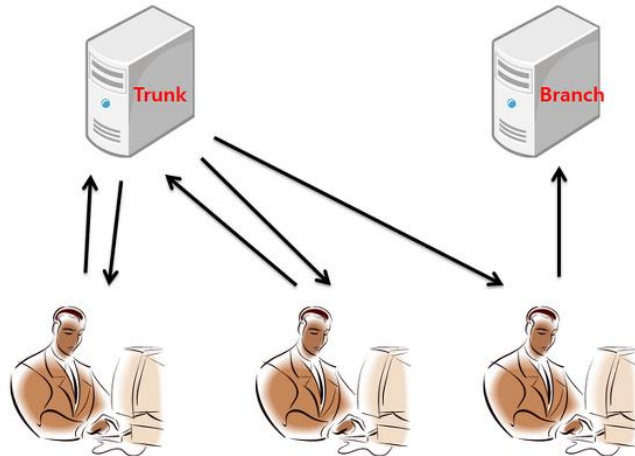
## 2) 대표적인 기능

- 여러 개발자가 함께 프로젝트를 진행 할 때 소스 관리를 할 수 있게 하기 위해 하나의 서버에서 관리 할 수 있도록 도와준다. 이 프로젝트 소스는 SVN 서버의 Trunk 에 위치한다.

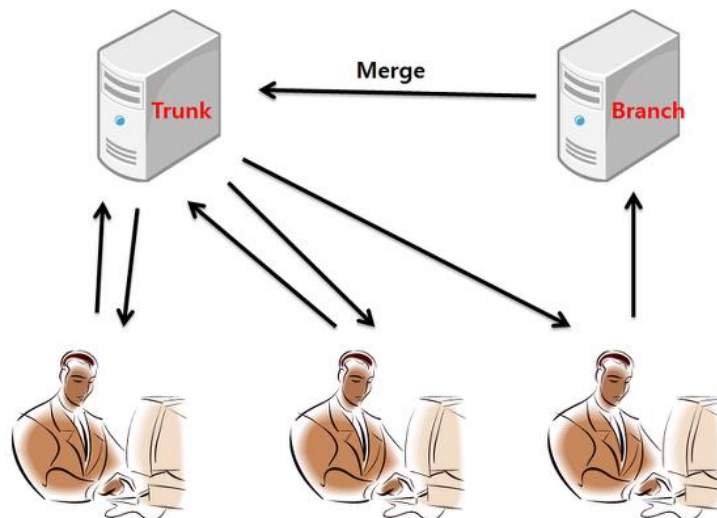


자신의 로컬에 Trunk 소스를 다운받아 수정 및 추가한 후에 다시 업로드(커밋, commit)하는 형식이다. 또한 로그를 통하여 누가 업로드를 하였는지 알 수 있다.

- 동일한 프로젝트의 특정 모듈이나 기존 소스에 영향을 끼치는 다른 기능을 개발하여야 할 경우 Branch 를 만들어 준다.



Branch 를 만들어서 자신이 원하는 소스를 자신만의 소스로 만들어 작업하면 된다. 다른 개발자들이 설정하지 않는 한 자신의 Branch 는 본인만 접근 권한을 가진다. 또한 이 Branch 에서 개발이 완성되면 Trunk 소스와 합치는 작업을 한다.



### 3) 장점

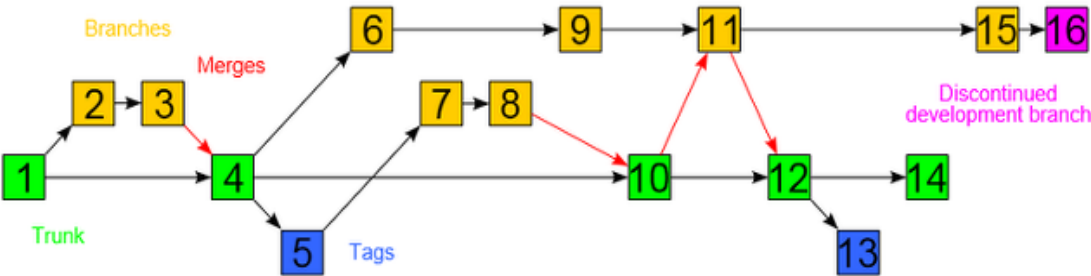
- 여러 명의 개발자들이 같은 코드 기반을 통제된 방식으로 다룰 수 있다.
- 파일 이름 변경, 이동, 디렉토리 버전 관리를 지원한다.
- 시간의 진행에 따라 변경 내역을 유지할 수 있다.
- 낫선 코드를 보더라도 코드를 누가, 언제, 왜 그렇게 고쳤는지 알아낼 수 있다.

- 주된 개발 흐름을 유지하면서도 여러 개의 Release 들을 만들어내는 것이 가능하여 개발 버전과 유저들에게 배포되는 릴리즈 버전이 서로 섞이지 않고 관리 가능하다.
- 여러 사용자가 동시에 커밋을 하더라도 충돌이 일어나지 않아, 팀이 작업을 멈출 필요가 없어진다.
- 서버와 클라이언트 간 원본을 유지함으로써 백업이 가능하다
- 파일 하나당 버전이 붙지 않고, 한번 커밋을 할 때 마다 revision 이 올라간다.

#### 4) SVN 의 기본 용어

항목	설명
Repository	모든 프로젝트의 프로그램 소스들이 저장되는 장소
Check Out	저장소에서 소스 받아오기
Commit	개발자가 소스를 수정, 파일 추가, 삭제등을 한 뒤 저장소에 갱신하는 것. Revision 번호가 증가한다.
Update	업데이트를 하여 저장소에 있는 최신 버전의 소스를 가져오는데 바뀐 부분만 받아온다.
Revision	소스 파일을 수정하여 커밋하면 숫자가 증가한다.
Import	아무것도 들어있지 않은 저장소에 맨 처음 소스를 넣는 작업
Export	버전 관리 파일들을 뺀 순수한 소스 파일 받아오기
Tag	배포판
Trunk	프로젝트에서 가장 중심이 되는 디렉토리
Branch	Trunk 에서 뿔어나온 나뭇가지
Merge	Branch 수정본을 Trunk 에 병합

5) SVN 기본 구조



## 2. Mantis 이란?

### 1) 도구 개요

소개	웹 기반의 버그 및 이슈 관리 시스템. 프로젝트 관리 및 소프트웨어 설계 시, 단위 별로 사용자가 작업한 내용을 기록할 수 있으므로 버그 추적까지도 유용하게 사용 가능
주요 기능	이슈 보고 이슈 상태 변경 이슈 알림 메일 보내기
카테고리	Configuration & Version Management
세부카테고리	버그 및 이슈 추적
커버리지	Bug Tracker
도구난이도	하
라이선스 형태/비용	GNU General Public License v2 / 무료
운영체제	Windows, Linux, Mac OS X, UNIX
특징	<ul style="list-style-type: none"> <li>• 각각의 작업이나 전체 프로젝트에 대해서 작업 진행 상태를 도식화 가능</li> <li>• 프로젝트 변경 이력에 대한 추적/관리 및 유지보수</li> <li>• 프로젝트 참여자들의 작업 내용을 추가/보고/관리</li> </ul>
적용회사/프로젝트	다양한 SW 개발업체들이 사용
관련 도구	Bugzilla
제작사	Mantis Group
공식 홈페이지	<a href="http://www.mantisbt.org">http://www.mantisbt.org</a>
사전 설치도구	PHP, MySQL, Apache2
도구버전	1.2.12 (2012. 11)

## 2) 기능 요약

웹 기반의 버그 및 이슈 관리 시스템으로 그룹 개발 시에 발생한 이슈나 버그들을 관리하기 위한 도구

주요기능	지원내용
오류 추적/관리	지원
버그 처리 우선순위	지원
프로젝트 별 관리	지원
다른 포맷으로의 출력	지원
E-mail 설정	지원

## 3) 도구 실행 환경

대부분의 운영체제 에서 실행 가능

- 다양한 OS를 지원
  - Windows : Windows XP / Windows 7 (32, 64-bit 모두 지원)
  - Linux : 32, 64-bit 지원
  - Mac OS X : 32, 64-bit 지원
  - UNIX : 32, 64-bit 지원
- MySQL, Apache2, PHP 을 필요로 함

## 4) Autoset 다운로드

Mantis 사전설치도구인 Autoset 다운

>[www.autoset.org](http://www.autoset.org) 무료 다운로드 클릭 → 설치 프로그램 다운 → 오토셋 6.3.1 선택 → 링크 클릭 → 다운 받은 후 설치를 시작

-설치는 C:\AutoSet 폴더에 함

설치 중, 옵션은 기본 설정에 따름

- 기본설정에 프로그램 바로가기는 시작 메뉴 폴더에 지정

- 설정을 마친 후, 완료 버튼을 클릭하여 설치 마칩



## 5) Mantis 다운로드

Mantis를 다운

- [www.mantisbt.org](http://www.mantisbt.org) 에 접속 → 메뉴 중 Download 클릭 → Download MantisBT 1.2.12 를 클릭하여 다운 → 클릭하면 SourceForge 사이트로 이동 → 1.2.12 버전 폴더 클릭 →

mantisbt-1.2.12.tar.gz 를 클릭하여 다운 → 다운받은 후 압축 해제 → mantisbt-1.2.12 라는 폴더가 생성 됨

경로를 쉽게 하기 위해 mantisbt-1.2.12 폴더 이름을 mantis로 바꾼 후, mantis 폴더 안에 upload 라는 폴더를 생성

- Upload 폴더는 앞으로 upload 할 파일이 저장되는 장소

mantis 폴더를 C:\AutoSet\public\_html 로 이동

## 6) Database 설정하기

• mantis 폴더에서 config\_defaluts\_inc.php 를 실행하여 DB정보를 설정함

- `$g_hostname = „localhost“; //hostname은 localhost로 함`

- `$g_db_username = „root“; //PHP myadmin 접속을 위한 username`

- `$g_db_password = „autoset“; //PHP myadmin 접속을 위한 password`

- `$g_database_name = „bugtracker“; //새로 만들 database 의 이름`

• `$g_administrator_email = „관리자 email 주소“`

- 관리자 email 주소에는 해당 관리자의 email 주소나 본인의 email 주소 작성

> 본 매뉴얼에서는 „rubyit0624@gmail.com“으로 입력 //테스트를 위해 본인 계정의 메일주소 작성

• `$g_fallback_language = „Korean“`

- Fallback 기본 언어를 Korean 으로 변경

• 파일 업로드 설정을 반드시 변경

• 기본 값은 DB에 업로드되는 설정이지만, 파일로 저장하는 것이 좋음

- `$g_file_upload_method = „DISK“`

- `$g_allow_file_upload = „ON“`

• 앞서 생성한 Upload 폴더의 절대경로 입력 및 최대 업로드 용량, 금지파일 목록 작성

- `$g_absolute_path_default_upload_folder = „C:/AutoSet/public_html/mantis/upload/“;`

- `$g_max_file_size = „5000000“;`

- \$g\_disallowed\_files = „php,html,htm“;

## 7) Database 만들기

- AutoSet 을 관리자 권한으로 실행시켜서 웹서버와 MySQL 을 시작
  - 새 Database 생성
  - 파일→ `phpMyAdmin 접속`을 통해 localhost에 연결
  - 앞에서 입력한 DB 사용자이름: root, 암호: autoset 을 입력하여 연결
  - 접속하여 „새 데이터베이스 만들기`란에, 앞에서 입력한 Database 이름 bugtracker를 입력하여 만들기 버튼 클릭
  - 설정 파일 생성
  - http://localhost/mantis/admin/install.php 로 접속
  - > Admin Username과 Admin Password는 앞에 입력한 Username과 Password와 동일하게 입력
  - > 입력 후, Install/Upgrade Database 버튼 누름
  - 아래 화면은 설정이 완료된 화면
- 완료 후에 mantis 폴더에 config\_inc.php 파일이 생성됨.
- http://localhost/mantis 로 접속
  - Username에는 administrator 를, Password 에는 root 를 입력하고 로그인
  - root 는 기본 password 이므로, 접속 후에 변경을 해야 함

## 8) Mantis 접속

- 만약 상단에, timezone 관련 에러가 발생 시에는 다음과 같이 파일을 수정
- mantis 폴더의 config\_defaults\_inc.php 파일을 열어서 다음 내용 삽입
- > date\_default\_timezone\_set("Asia/Tokyo");

## 9) Mantis 한글 설정하기

- 한글로 된 메뉴를 사용하기 위해 My Account - Preferences 누름
  - Language 를 Korean 로 선택한 뒤 Update Prefs 버튼을 눌러 설정 적용
  - 만약 한글 설정이 되지 않는 경우에는 임의의 프로젝트를 하나 생성 후, 적용함
  - DB에 저장되는 데이터에서 한글이 올바르게 표시되게 하기 위한 작업
  - mantis 폴더 안에 core 라는 폴더가 있고, 그 안에 database\_api.php 파일을 열어 다음을 추가
- ```
> $g_db-> Execute("set names utf8");
```

## 10) Mantis 정보 갱신

- 계정관리를 눌러 비밀번호, 실명, 메일을 각각 채운 뒤 정보 갱신
- 메일은 앞에서 생성한 이메일과 동일한 것으로 입력

## 11) Mantis 메일 설정하기

- C:\AutoSet\Server\conf\php.ini 파일을 다음과 같이 수정
- ```
> :extension=php_openssl.dll → extension=php_openssl.dll
```
- AutoSet 에서 웹서버 재시작
  - mantis 폴더의 config\_defaults\_inc.php 파일 수정
  - 메일 전송 테스트
  - [http://localhost/mantis/admin/test\\_email.php](http://localhost/mantis/admin/test_email.php) 에 접속하여 테스트

## 12) admin 폴더 삭제하기

- Mantis의 설정이 완료되면, 보안상의 문제가 있을 수 있으므로 admin 폴더 삭제
- mantis 폴더 안의 admin 폴더를 삭제함.

## 13) 프로젝트 생성

- 프로젝트 생성
- Mantis 로그인 후 관리 → 프로젝트 관리 → '새로운 프로젝트 생성' 버튼 클릭
- 프로젝트 이름: 웹 사이트 개발 프로젝트 //현재 진행중인 프로젝트 명을 기입
- 상태: 개발버전 //선택 항목들 중에 해당하는 것을 항목을 선택
- 상태보기: 공개 // 공개, 비공개 중 선택
- 분류 상속: 체크해제 // 독립적으로 관리 할 경우 체크해제
- 설명: 프로젝트 팀원 들이 이 프로젝트에 대해 알 수 있게 간단한 설명 기입

## 14) 계정 생성

- 프로젝트에 참여하는 팀원 입력
- 관리 → 사용자 관리 → 계정생성 버튼 클릭
- 항목 정보를 입력
- 필수 사항들을 기입한 후 사용자 생성 버튼 클릭

## 15) 프로젝트에 계정 할당

- 관리 → 사용자관리 메뉴에서 생성된 계정(팀원)을 확인하고 프로젝트에 추가
- 사용자 ID를 클릭하여 추가
- 프로젝트를 선택하여 접근 레벨을 지정하고 사용자 추가 버튼을 클릭

## 16) 계정 설정 상태

- 기본 계정 설정상태는 초기에 체크된 상태를 따름 (이메일 보고에 제한을 두고 싶을 경우 체크상태를 바꿀 수 있음)

- '이메일'란 을 '모두'로 설정하였기 때문에 이슈 상태 변경 시마다 메일이 도착

## 17) 계정 생성 확인

- 계정을 추가하고 나면 추가된 팀원에게 계정생성 메일이 발송됨
- 메일에 링크되어 있는 주소로 접속

- 비밀번호를 설정

## 18) 이슈 보고

- 프로젝트 진행 도중 일정보다 작업이 지연되는 경우 이슈가 발생했다고 말할 수 있음

- 이 때, 이슈보고하기 기능을 이용하여 팀원과 PM 에게 이슈를 보고 할 수 있음

- 이슈 보고하기

- 이슈 보고하기 메뉴에서 웹사이트 개발 프로젝트를 선택한 뒤 프로젝트 선택 버튼 클릭

> 만약 프로젝트 선택 화면이 나오지 않을 경우, 우측 상단의 프로젝트에서 '모든 프로젝트' 선택

- 이슈에 대한 정보를 입력한 뒤 보고서 제출 버튼을 클릭

- 메뉴의 내 페이지 화면에서 이슈 보고를 확인

- 화면 하단에 색으로 구분하여 이슈상황에 대해 분류하고 있음

> 새로운 이슈: 이슈로 등록하여 아직 확인되지 않은 이슈

> 정보 부족: 정보가 더 필요하며 처음 보고한 사람은 주목해야 함

> 이슈 검토: 이슈에 대해 조사했으나 확인이나 할당되지 않은 상태이며, 해당 개발자는 이슈로 인정하지 않을 수 있음을 유저에게 알림

> 확인된 이슈: 개발자가 이슈 내용을 확인하고 재현할 수 있음. 보통 정보를 갱신하는 사람(updater) 이나 다른 개발자가 설정

> 할당된 이슈: 현재 개발자에게 할당되어 처리 중

> 해결된 이슈: 이슈를 수정한 것으로 판단하며, 이에 대한 확인을 기다림

> 닫음: 이슈가 완전 해결되어 폐쇄

## 19) 이슈 보고서 출력

- 이슈보기에서 이슈에 대한 보고서를 출력할 수 있고, CSV 내보내기 기능이 있음
  - 보고서 출력 기능
- Explorer 창에 이슈에 대한 간단한 보고서 출력이 가능
  - 표시된 아이콘을 클릭
  - Explorer 창에 보고서가 출력
- Word로 이슈에 대한 자세한 보고서 출력이 가능
  - 표시된 아이콘을 클릭
  - 열기 버튼을 클릭
- Word에 보고서가 출력 됨
  - 이슈에 대한 이슈번호, 해당 프로젝트, 중요도, 재 발생 가능성, 보고날짜, 최종 갱신날짜 등 이슈에 대한 자세한 보고가 가능
- 현재 프로젝트의 이슈에 대한 전체적인 상황보기가 가능
  - 요약클릭
  - 전체적인 이슈를 요약한 보고가 가능
- 보고서 출력 버튼을 클릭하면 보고서가 출력

## 20) 이슈 보고 알림 메일 보내기

- 이슈번호를 클릭하면 이슈를 갱신할 수 있고, 팀원들에게 알림 메일을 보낼 수 있음
  - 이슈 번호 클릭
  - 알림 메일 보내기 클릭
  - 간단한 메시지를 입력하여 전송 버튼 클릭

## 21) 이슈 보고 알림 메일 확인

- 이슈보고에 대한 메일이 왔는지 확인
- 프로젝트 관리자로부터 이슈 발생 메일이 도착함
  
- 메일에 링크되어 있는 주소로 접속하여 이슈를 확인

## 22) 할당된 이슈 확인

- 나에게 할당된 이슈 번호를 클릭
  
- 현재 website\_member3 이 로그인한 상태
  
- 본인에게 할당된 이슈라면 확인 후에 이슈의 상태를 '확인된 이슈' 상태로 변경
- '이슈노트 추가란'에 다른 팀원들이 볼 수 있도록 상태 변경에 대한 이유 작성
  
- 이슈의 상태가 '확인된 이슈' 로 바뀜

## 23) 이슈 해결

- 할당자는 이슈를 확인한 뒤 이슈를 해결(프로젝트 관리자의 지시대로 작업에 참여)
  
- 이슈보기 메뉴에서 이슈번호 클릭
  
- 이슈를 해결한 뒤에 '해결된 이슈' 상태로 변경
  
- 상태 변경에서 '해결된 이슈' 로 변경하고 상태 변경 버튼을 클릭
  
- 이슈 해결 상태 선택
- 해결상태
  - > 수정됨: 시험 완료 됐으며, 이슈 상태는 해결되었다고 표시
  - > 문제 제기: 이전에 해결했지만, 당시 처리 결과가 만족스럽지 않거나 정확하게 처리되지 않아 다시 발생한 상태
  - > 재현할 수 없음: 이슈를 재현할 수 없으며 코드를 살펴봐도 파악할 수 없는 상태이고 추가적인 정보가 들어오면 다시 할당하도록 함
  - > 수정할 수 없음: 이번 버전에서는 수정할 수 없는 문제
  - > 중복: 기본의 이슈와 중복된 것으로 중복된 이슈 번호 필요 ('#이슈번호'로 표시)
  - > 보류(suspended): 중요하지 않은 문제로 수정을 보류
  - > 수정 계획 없음
- 중복된 이슈 번호: 해결 상태를 중복으로 선택하였을 경우 중복된 이슈번호를 알려줌

- 이슈노트 추가: 해결 상태에 대해서 프로젝트 팀원들이 볼 수 있도록 Comment 를 작성
- 모든 입력란을 작성하고 이슈해결 버튼을 클릭

#### 24) 이슈 해결 상태 확인

- 관리자가 이슈가 해결된 후 로그인한 화면
- 이슈의 상태가 해결된 이슈의 색으로 변경됨
- 관리자(PM)은 해결 보고 된 이슈에 대해 검토한 후 이슈의 상태를 변경
- 이슈가 제대로 해결되지 않고 부족하다면 '정보 부족' 상태로 변경
- 이슈가 제대로 해결되었다면 '다음' 상태로 변경
- 이슈가 제대로 해결되었다고 가정하고 '다음' 상태로 변경
- 이슈번호를 클릭

#### 25) 이슈 폐쇄

- 이슈의 상태를 '다음' 를 선택한 뒤 상태 변경 버튼을 클릭
- 이슈 노트를 작성 후 '이슈 닫기' 버튼을 클릭

#### 26) 용어 정리

- Bug Tracking

SW 개발 및 테스트 단계에서 발생하는 일종의 문제점 또는 오동작 건들을 Bug라고 하며 이러한 Bug들을 지속적으로 추적/관리하는 것을 Bug Tracking 이라 함

- Issue

프로젝트에 비교적 큰 차질을 가져올 수 있는 "발생된(Realized)" 위험으로 정의됨. 물론 미리 예상해 두지 못했던 문제점(Issue)도 있을 수 있음

- MySQL

DBMS종류 중 하나이며, Oracle사에서 인수하여 지속적인 Update가 이루어 지고 있음.

Linux에서 가장 보편적으로 사용하는 DBMS임

- Apache2

2.x 버전의 Apache 서버를 말하며, Apache서버는 Linux OS에서 가장 보편적으로 사용되는 웹서버. 무료로 배포되며 perl, cgi, tomcat등과 연동하여 서비스가 가능함



### 3. JFeature 이란?

#### 1) 개요

JFeature는 요구사항들을 기록하고 이 들을 Junit Test Case와 Match함으로써 Process를 단순화 하는 도구이다. 주요 기능으로는 요구사항기록, Report 생성하기가 있다. 이를 사용하기 위해서는 Eclipse 3.4 이상의 버전이 사전에 설치되어 있어야 하며 Windows, Linux, MAC OS, UNIX에서 지원이 된다.

#### 2) 특징

- Junit의 Test Case 메소드와 통합이 가능하다.
- 요구사항의 수정 사항이 생겼을 때, 즉시 뷰를 제공하므로 빠른 편집이 가능하다.
- 요구사항의 반영 여부를 Coverage 형태로 쉽게 확인할 수 있다.
- 요구사항들을 편집하는 확장 인터페이스, 추가 Reporting 포맷, 요구사항 스펙 Export가 가능하다.

#### 3) 기능 요약

주요 기능	지원 여부
요구사항 반영여부 확인	Coverage 형태로 지원
타 플러그인 환경 지원	Junit을 지원
코드 추적	지원
IDE 내 별도의 편집기 제공	지원
요구사항 스펙 반출	일부 지원
리포팅	기본 포맷 지원
Round Trip Engineering	지원

#### 4) 도구 실행 환경

- 지원 OS

OS	32-bit	64-bit
Windows XP	0	0
Windows 7	0	0
Linux	0	0
MAC OS X	0	0
UNIX	0	0

- 사전 설치 프로그램

JDK, Eclipse가 필요하다. 코드 및 Eclipse 플러그인 형태, 도구에 포함되어 있는 형태로 JFeature의 서비스가 제공된다.

#### 5) JFeature 설치 방법

- <http://www.technobuff.net/> 에서 JFeature 다운을 받는다.
- 파일을 다운로드 받은 뒤, 압축을 풀고 JFeature를 Eclipse가 설치된 폴더 내의 plugins폴더에 복사한다.
- Eclipse를 실행하여 File->Other->New 창에서 추가된 JFeature를 확인 한다.

## 6) JFeature의 주요기능

- 요구사항 작성 및 Junit의 Test Case 메소드와 요구사항의 연결이 가능
- 요구사항 입력 : 직접 입력하는 방법, 파일을 통한 입력(CSV, XML 파일 읽기)
- 요구사항 저장 : JFeature로 정의된 요구사항을 파일로 저장(CSV나 XML)
- JFeature의 항목 및 구성

### a. 화면 구성

항목	기능
Requirement ID	요구사항의 고유의 식별 번호
Category	요구사항의 상위 분류
Subcategory	요구사항의 하위 분류
Title	요구사항 명칭
Target release	요구사항의 타겟 배포 버전
Priority	요구사항의 구현우선순위
Must Have	필수 요구사항인지 아닌지에 대한 판별 여부
Test methods	요구사항과 매핑 될 테스트케이스
Dependencies	요구사항과 연관된 다른 요구사항
Description	요구사항의 상세한 설명

### b. 도구 모음 기능

항목	기능
Add Requirement	요구사항추가
Delete Requirement	선택된 요구사항 삭제
Copy Requirement	복사하기
Paste Requirement	붙여넣기
Cut Requirement	잘라내기
Move Requirement Up	요구사항 목록에서 위로 이동
Move Requirement Down	요구사항 목록에서 아래로 이동
Round Trip Engineering	현재 JFeature 내에서 쓰이고 있는 요구사항 파일의 변경 시 불러온 CSV 파일에도 반영되게 함. 또는 CSV 파일의 변경시 JFeature 내에서 쓰이고 있는 요구사항을 파일에 반영

## 7) JFeature 사용 예시

- 클래스 생성  
프로젝트 선택 → 마우스 우 클릭 → New → Class → Class 이름 설정 → Finish
- 클래스 구현
- 요구사항 작성  
File → New → Other → JFeature → New Requirements File → Next → File name
- 요구사항 입력  
우 클릭 → Insert Requirement Below → 해당 항목에 입력
- 요구사항을 불러온다면  
File → Import → Other → Import Requirements → Next → 형식 선택 → Next → Use Text Delimiter → 텍스트 구분 형식 선택 → Indicates a required field → Source field from Imported File → Next → 파일 위치 지정
- 요구사항을 저장하고 싶을 때  
File → Export → Other → Export Requirements → 형식 선택 → 저장 위치, 파일 이름 지정 → Finish
- Round Trip Engineering 사용 : Import 한 파일과 요구사항의 변경사항에 대한 동기화  
Round Trip Engineering 선택 → 항목 선택 → OK
- Round Trip Engineering의 각 기능

항목	기능
Synchronize JFeature requirements file	JFeature 요구사항 파일의 변경사항을 파일에 변경
Synchronize external requirements file	CSV, XML 파일을 JFeature 요구사항 파일에 적용
Synchronize using Requirement ID	요구사항 ID를 이용하여 비교 후 적용
Synchronize using Requirement Title	요구사항 제목을 이용하여 비교 후 적용
Append non-matching requirements	일치하지 않는 요구사항 추가

- Report 생성

a. Test Case Method와 연결

테스트할 클래스 선택 → 우 클릭 메뉴의 New → Junit Test Case → 그대로 두고  
Next → 테스트할 메소드 체크 → Finish → 메소드의 내용 구현 → Test Methods의 오  
른쪽 버튼을 클릭 → 앞서 작성한 Test Class 찾기 → 관련이 있는 Test Class Method  
선택 → Append to Test Method

b. Report 생성

프로젝트 → Properties → JFeature → Associate Requirements File → 반영할 파  
일 선택 → Run → Run As → JUnit

## 4. JUnit 이란?

### 1) 도구 개요

도구 개요	JUnit은 Java기반 테스트를 위한 프레임워크로, 단위모듈 (method)이 정확히 구현 되었는지를 확인할 수 있는 도구
소개	Test Case 생성 및 실행, 오류추적
주요기능	Testing
카테고리	Testing
커버리지	테스트 설계 및 실행
세부카테고리	하
도구난이도	Windows, Linux, Max OS X, UNIX
운영체제	Java 기반 IDE
사전설치도구	<ol style="list-style-type: none"><li>1. 메소드와 같은 단위 모듈 별 테스트를 가능케 함으로서 코드 품질을 보장</li><li>2. 정확한 단위 테스트를 가능케 함으로써 통합 테스트 시 회귀결함(모듈통합에 의해 발생하는 결함)을 줄임</li><li>3. 다른 모듈에 의존하지 않고, 원하는 모듈만 임의의 순서대로 수행할 수 있다</li><li>4. JFeature(요구사항개발도구)와 통합되어 요구사항의 정확한 구현 비율을 알 수 있다</li></ol>

## 2) 기능요약

주요기능	지원내용
테스트 범위	단위테스트
TDD 환경 지원	지원
코드지원	지원
Test Suite	지원
UI Report	지원
Code Trace	지원, 실시간 추적 가능

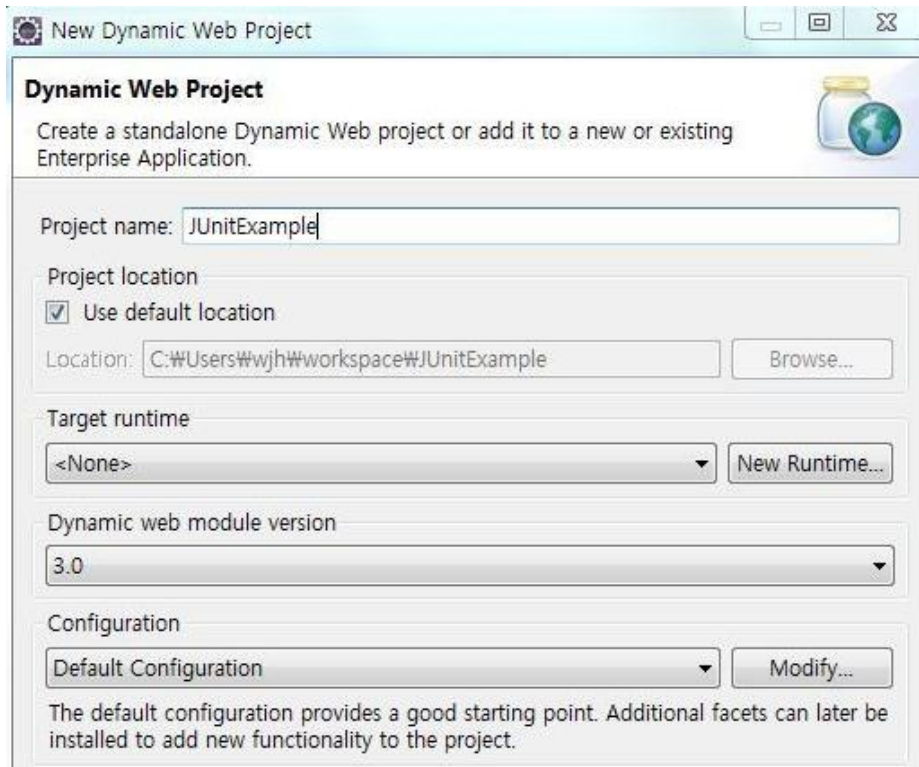
## 3) 도구 실행 환경

- Window, Linux, Max OS X, UNIX : 32, 64-bit 지원

JUnit
Java IDE (Eclipse, NetBeans 등)
JDK (Java development kit)
Windows, Linux, Max OS X, UNIX

#### 4) 도구 설치 방법

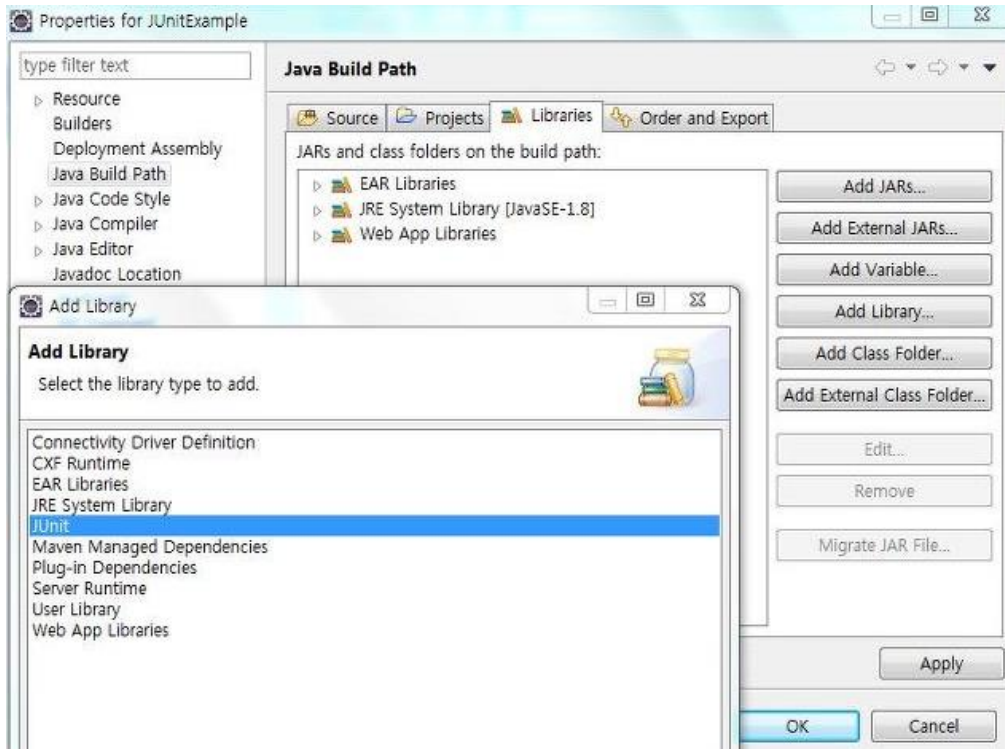
Eclipse를 설치하면, Junit library를 추가해서 사용할 수 있다.



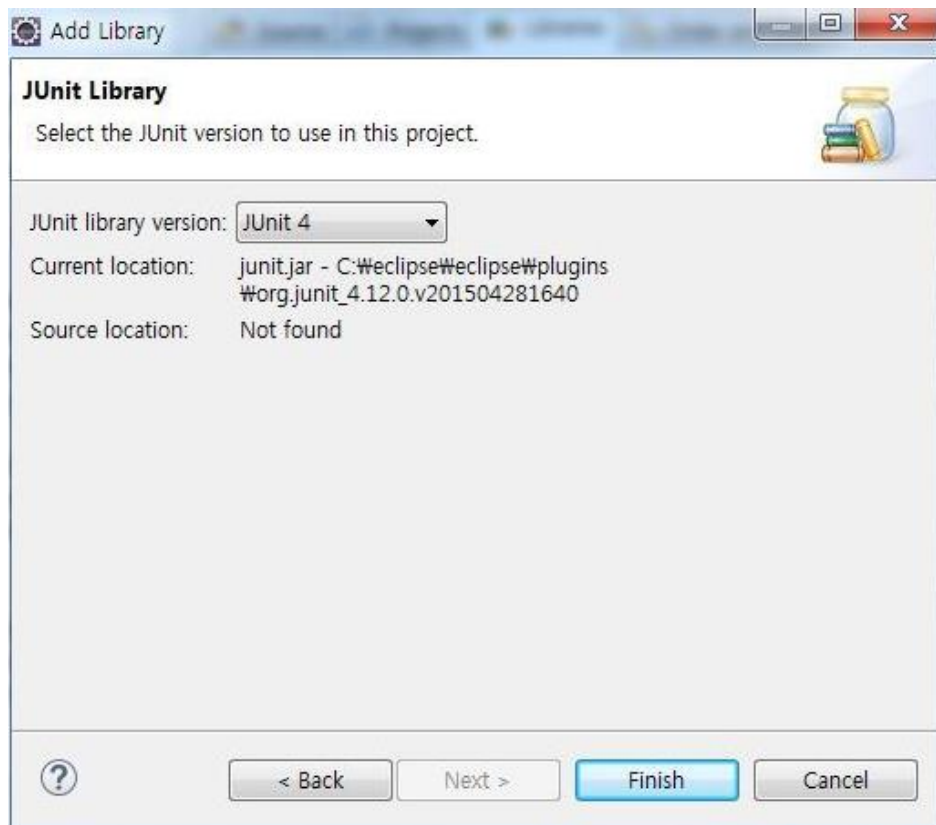
ㄱ. 프로젝트 우 클릭 -> Properties -> Java Build Path -> Libraries 탭 클릭

ㄴ. 프로젝트 우 클릭 -> Build Path -> Configure Build Path... -> Libraries 탭 클릭



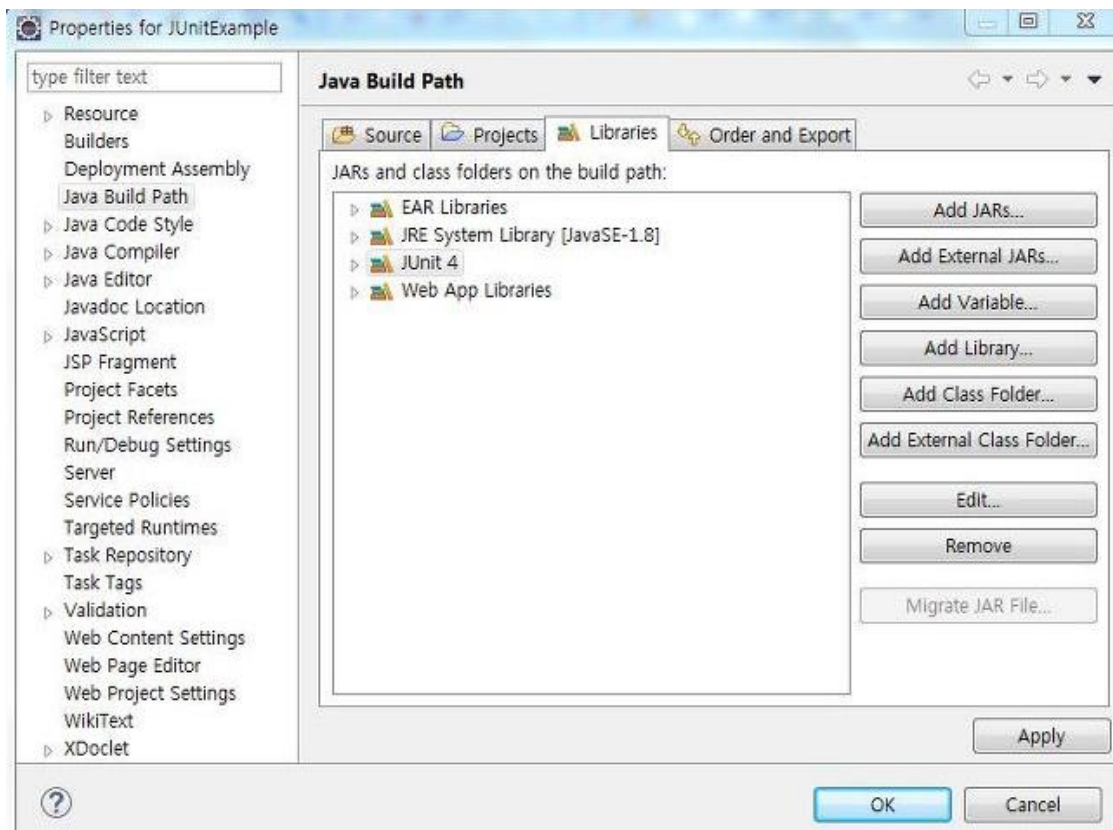


Libraries 탭에 가서 Add Library 버튼을 클릭하면 추가할 수 있는 Library 목록이 보이는데 그 중  
간 정도에 Junit이 있다.



Location 경로를 보면 Eclipse plugin에 내장되어 있다. 없는 경우에는 Junit jar 파일을 별도로 다운 받으면 된다.

Jar 파일을 다운로드 받아서 Class Path 경로에 넣어주거나, maven을 사용하는 경우에는 maven 설정에 Junit dependency 값을 넣어주어도 된다.



추가 된 모습을 볼 수 있다.

## 5) 도구 기능 소개

- 주요기능 :

- (1) 테스트 하고자 하는 메소드에 대해 Test Case를 만들 수 있다,
- (2) 일부의 특정 테스트 메소드를 실행하지 않거나 특정 테스트 메소드만 실행하고 싶을 때, 혹은 테스트 클래스를 한데 묶어서 실행하고 싶은 경우 Test Suite를 사용한다,
- (3) 단위테스트를 실행하고 테스트 결과를 빠르게 확인할 수 있다

- Test Case 작성하기

(1) 테스트 하고자 하는 클래스를 생성

(프로젝트선택 -> 마우스 우 클릭 메뉴의 New -> Class -> Class이름 설정 -> Finish)

(2) 클래스의 코드를 작성한다

(3) 테스트 하고자 하는 클래스에 대해 Test Case를 생성

(테스트를 실행할 클래스 선택 -> 마우스 우 클릭 메뉴의 New -> Junit Test Case)

(4) Source folder, Package, Name, Superclass 등을 그대로 두고 Next

(5) 클래스 내의 테스트하고자 하는 메소드 체크 Finish

(6) 테스트 하고자 하는 메소드의 내용을 구현

(전체 메소드에 대해서 구현한다)

- Test Suite 작성하기

(1) Test 클래스를 한데 묶어서 테스트(본 예시), 특정 메소드만 실행하거나 하지 않을 때  
사용 (메뉴의 File -> New -> Other -> Junit 선택 -> JUnitTestSuite -> Next)

(2) 테스트 할 클래스를 선택 Finish









- 테스트 실행 및 결과

(1) JUnit 테스트의 실행 (Test Case선택 -> RUN -> RUN As -> Junit Test )

(2) 단위테스트에 성공 시(초록색 막대), 실패 시(빨간색 막대) 가 표시된다.

- 테스트 실행을 위한 메뉴 소개

(1) Tool bar 기능소개

 Nest Failed Test	테스트 메소드 목록에서 실패한 테스트 메소드 목록간에 하위 이동을 수행
 Previous Failed Test	테스트 메소드 목록에서 실패한 테스트 메소드 목록간에 상위 이동을 수행
 Show Failures Only	실패 항목만 보임
 Scroll rock	스크롤바를 고정
 Return Test	테스트를 다시 실행
 Return Test (Failures First)	테스트를 다시 실행
 Test Run History	테스트 실행 히스토리를 보여줌
 Stop JUnit Test Run	JUnit Test 실행을 멈춤

## 6) 도구 평가

- 활용성: 대부분의 자바를 지원하는 도구에서 사용이 가능하다
- 범용성: 뛰어난 성능과 넓은 활용범위에 의해 대부분의 도구에 기능이 들어있다.
- 호환성: 도구의 버전을 가리지 않는다,  
  - 도구가 가능한 환경이면 어떤 환경에서도 작동이 가능하다.
- 성능: 빠르다  
  - 도구에 같이 로드 되어도 성능상으로 문제가 없다.
- 기타: 기본으로 들어있는 플러그인 이더라도, 필요한 경우 버전업이 가능하며 여러 버전을 동시에 설치할 수 있다.

## 7) 용어정리

용어	설명
Test Case	특정한 프로그램부분 및 경로를 실행해 보거나 특정한 요구사항에 준수하는지를 확인하기 위해 개발된 입력 값, 실행조건, 그리고 예상된 결과로 이루어져 있는 하나의 테스트 세트
Test Suite	일정한 순서에 의하여 수행될 개별 테스트들의 집합, 또는 패키지. Suite는 응용분야나 우선순위, 내용에 연관된다
TDD	자동화된 테스트로 개발을 이끌어가는 개발방식을 테스트 주도 개발이라 칭함. TDD는 분석기술이며, 설계기술이며, 개발의 모든 활동을 구조화 하는 기술
단위테스트	단위테스트는 단위코드에서 문제발생소지가 있는 모든 부분을 테스트하는 작업

## 5. CTIP 개론

### 1) CI 개요

기존 개발 단계의 후반에 수행되던 통합 활동을 일상적인 개발 활동에 자연스럽게 녹인 것이 지속적 통합(CI - Continuous Integration) 이다.

### 2) CI 활동

- 개발자는 소스 저장소로부터 최신 소스를 내려 받는다 (Check out)
- 개발자는 코드를 작성한 후 자신의 개발기 에서 정상적으로 동작하는지 충분히 검증한다
- 검증이 끝난 후 개발자는 작업 내용을 소스 저장소에 올린다 (Check in)
- 개발자(또는 통합 관리자 또는 자동화된 시스템)는 통합 서버에서 방금 작업한 코드가 반영된 전체 코드 내용에 대한 빌드를 수행한다.
- 만약 빌드가 실패할 경우, 실패 원인을 분석하고 문제를 해결하여 빌드가 성공될 수 있도록 한다.

### 3) CI 활동의 장점

- 위험을 줄일 수 있다
- 수동으로 수행해야 하는 반복 작업을 줄일 수 있다
- 언제, 어느 장소에서도 배포 가능한 소프트웨어를 만들 수 있다
- 프로젝트에 대한 더 나은 가시성을 제공한다
- 개발 막바지의 통합 작업이 필요 없어지게 된다.

#### 4) CI 수행 요건

- 단일 소스 저장소(Source Repository)를 유지한다
- 빌드를 자동화한다.
- 빌드가 자체적으로 테스트 가능하도록 한다
- 모든 사람은 매일 작업 내용을 커밋(Commit)한다
- 모든 커밋들은 통합 서버의 메인 라인에 반영한다
- 각 빌드가 빨리 수행한다
- 운영환경과 되도록 비슷한 환경에서 테스트 한다
- 최신 결과물에 쉽게 접근할 수 있도록 한다
- 현 빌드 상황을 쉽게 알 수 있도록 한다

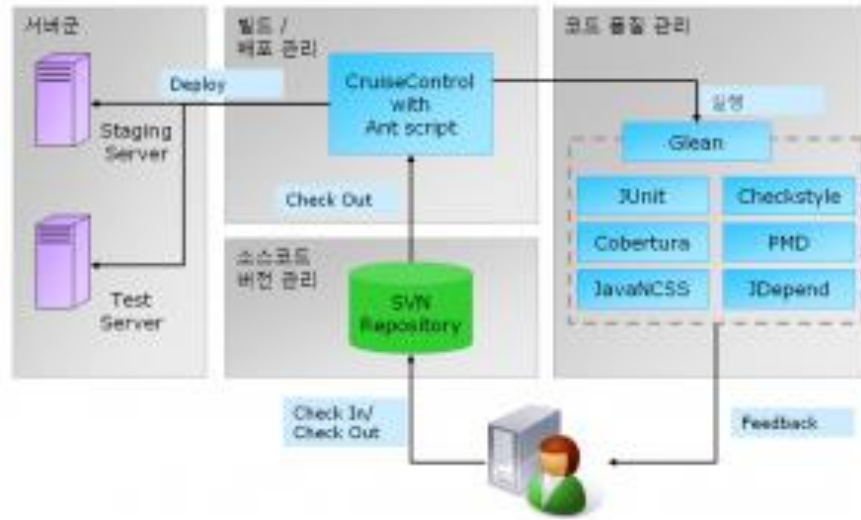
#### 5) CTIP 개요

- CI 개념을 Java 기반의 개발 프로젝트에 쉽게 적용하기 위한 Platform 이다.

#### 6) CTIP 기능

- CI 서버를 통한 지속적 통합 및 빌드
- 품질 도구들을 통한 코드 품질 검토 (테스트 및 정적 분석)
- 빌드 결과의 배포 및 관련자에게 통보

## 7) CTIP의 구조



]

## 8) CTIP의 구성

구성요소	내용
소스코드 버전 관리 영역	CVS, SVN 등의 소스코드 버전 관리 시스템을 사용하여 프로젝트 전체 소스코드의 일관성을 유지한다. 개발자는 버전 관리 시스템의 repository로부터 최신 소스를 check out하고, 작업내용을 check in 한다.
빌드 및 배포 관리 영역	CI 서버를 통해 지속적으로 빌드를 수행하고, 대상 서버에 대한 배포작업을 수행한다. 빌드 주기는 정해진 시점에 수행하거나(nightly build), 버전 관리 시스템의 repository에 변경이 있을 경우 즉시 수행될 수 있다 (아래와 같은 변경) <ul style="list-style-type: none"> <li>- 소스코드 버전관리 시스템으로부터 최신 소스를 받아온다</li> <li>- 코드 품질 관리 도구를 실행한다</li> <li>- 빌드 결과를 개발자에게 통보한다</li> <li>- 서버에 빌드 된 어플리케이션을 배포한다</li> </ul>
코드 품질 관리 영역	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
대상 서버군	운영 서버 배포를 위한 스테이징 서버, 테스트 실행을 위한 테스트 서버 등을 운영한다