CM & RE tools

Software Verification Team Project #2

KU KONKUK UNIVERSITY

MISELL KIM EUNJEONG KIM JUNYONG UHM LEE JISUNG GYEONGMI HAN

The Concept of CM

- (Software) Configuration Management
  - (소프트웨어) 형상 관리
  - 시스템이나 제품의 성능과 기능 및 물리적 특성들을 지속적으로 관리
  - 관리를 위한 표준과 절차를 개발하고 적용하는 것

- Software Configuration
  - Software Engineering의 Process로부터 생성된 모든 정보 항목의 집합체
  - 프로젝트 계획서, 명세서, 설계서, 구조도, 프로그램 코드, 데이터, 테스트 케이스, 유지 보수 문서 등

The Purpose of CM

- 시스템 파악
  - 시스템은 개발과 이용 중 항상 변한다.
    -> 여러 다른 형상으로 존재한다.
  - 상이한 컴퓨터에 관해, 상이한 운영체제에 관해, 클라이언트의 특정 기능을 수용하기 위해 다른 형상 생성 가능
- 어떤 변경 사항이 어떤 시스템 버전에 수용 되었는지 추적 가능
  - 소프트웨어 버전 사이의 차이점 파악
  - 새로운 버전이 통제된 방식으로 개발되는 것을 보증
- 새 버전이 제 시간에 고객에게 확실하게 인도되는 것을 보증

Configuration Management Process

- 형상 관리 계획 수립
  - 형상 관리를 위해 사용되어야 하는 표준과 절차를 기술
  - 형상 항목 식별
  - 형상 데이터베이스 구축
  - 책임자 선정, 정책 정의, 도구 및 프로세스 명시

- 변경 관리
  - 통제된 방식으로 시스템에 변경이 적용됨을 보장
  - CRF(Change Request Form) 작성
  - 형상 데이터베이스에 등록
  - CCB(Configuration Control Board)에 의한 검토

Configuration Management Process

- 버전 관리
  - 시스템의 버전들을 식별하고 기억
  - 필요한 시스템 버전 검색 가능
  - 개발팀(공동 작업자)에 의해 우연히 변경되지 않도록 보장
  - variation : 작은 차이가 있는 버전

- 릴리스 관리
  - 고객에게 배포된 시스템의 버전을 관리
  - 설정 파일, 데이터 파일, 설치 프로그램, 문서, 포장 및 광고

- 시스템 구축
  - 시스템의 컴포넌트들을 컴파일하고 연결

Tools for CM

- SE Tool의 필요성
  - 툴을 도입함으로써 일정 수준의 프로세스와 품질 수준을 담보
  - 툴을 통해 정량적인 프로젝트 관리와 효율적인 리소스 관리가 가능
  - 툴을 통해 기술 인력이 아닌 프로세스가 개발 업무를 수행(개발자 부재 시 다른 개발 인력이 인수 개발 가능)
  - 툴을 통해 인력을 통제하는 것이 아닌 프로세스를 통제 함으로서 보다 효율적인 SW 개발 프로젝트 추진 가능

Tools for CM

- 버전 관리 시스템
  - 기능
    - 버전과 릴리스 식별
    - 기억 장소 관리
    - 변경 이력 기록
    - 독립적인 개발
    - 프로젝트 지원
  - Folder 공유 타입
    - RCS, SCCS
  - Client/Server 타입
    - **Subversion(SVN)**, CVS, Perforce, ClearCase, TFS
  - 분산 저장소 타입
    - **Git**, Mercurial, BitKeeper, SVK, Darcs

SVN

- Client / Server 타입의 버전 관리 시스템

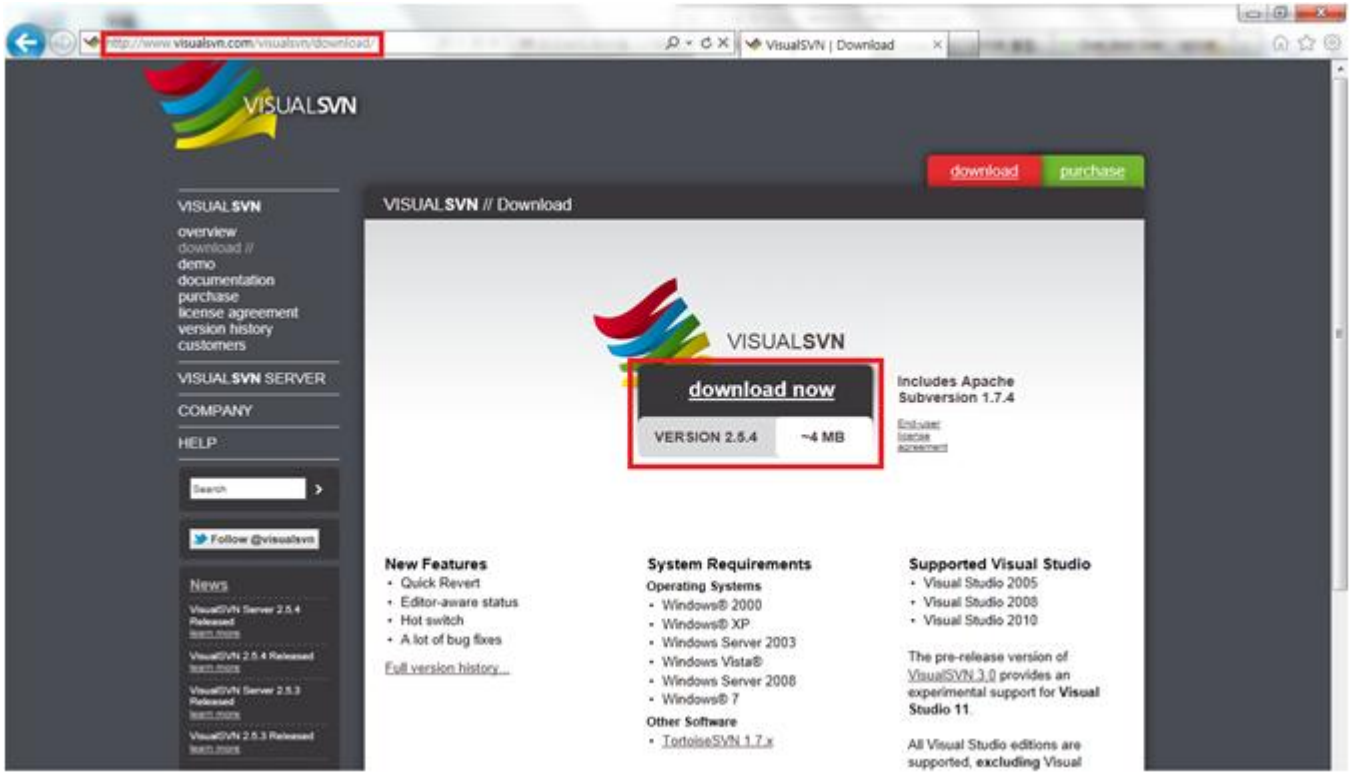
- CVS의 단점을 보완하기 위한 오픈소스 프로젝트로 개발
  - CVS에 빠른 업데이트 및 브랜칭 태킹 시간
  - Commit 단위가 파일이 아니라 체인지 셋이라는 점
  - CVS와거의동일한사용법
  - 원자적Commit
  - 양방향 데이터 전송을 통한 네트워크 트래픽 최소화
  - 트리별, 파일 접근 제어 리스트 (권한에 따른 접근)

SVN

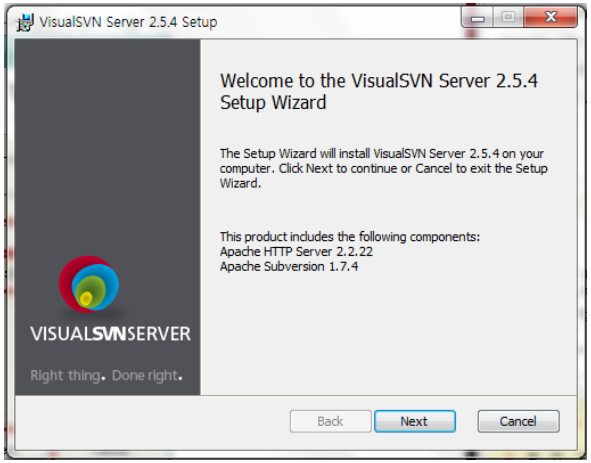
- Terms of SVN
  - Repository - 네트워크를 통해서 여러 사람이 접근 할 수 있는 저장소.
  - Check out - 저장소에서 소스를 받아오는 것.
  - Commit - Check out한 소스를 수정한 뒤 저장소에 갱신 하는 것.
  - Revision - 저장소에 저장된 각각의 파일 버전.
  - Head Rivision - 최신 리비전
  - Import - 아무것도 없는 저장소에 맨 처음 소스를 넣는 작업
  - Export - 버전 관리 파일을 뺀 순수한 소스 파일을 다운로드
  - Revert - 로컬 작업을 버리고 서버의 내용으로 되돌림

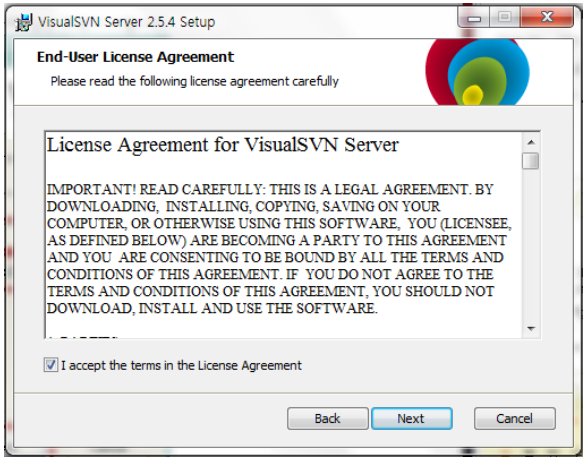
VisualSVN

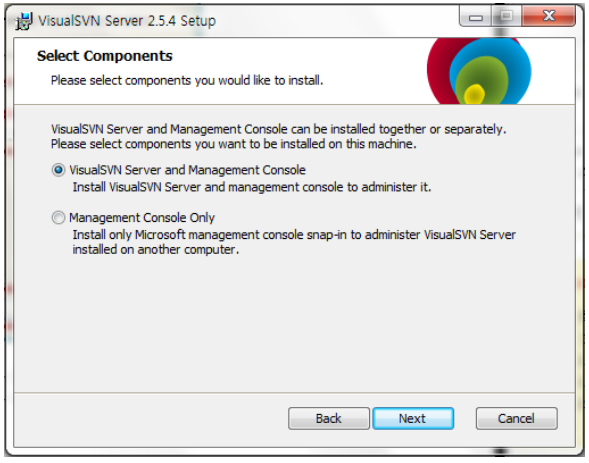
http://www.visualsvn.com/visualsvn/download/

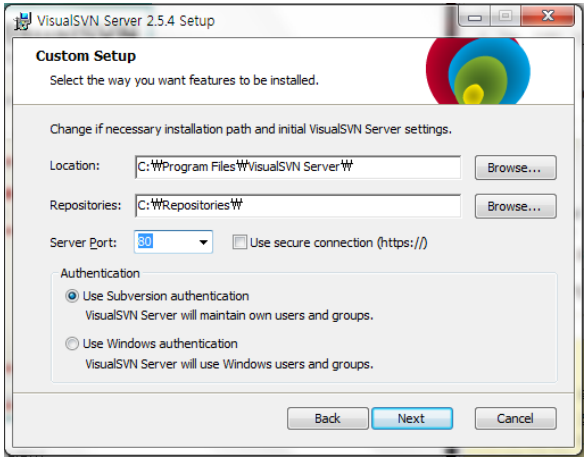


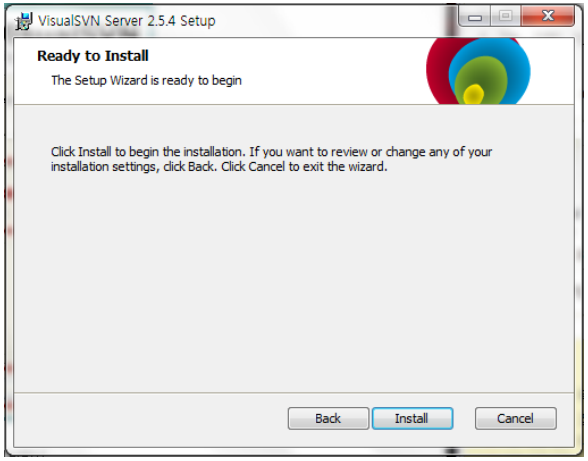
VisualSVN

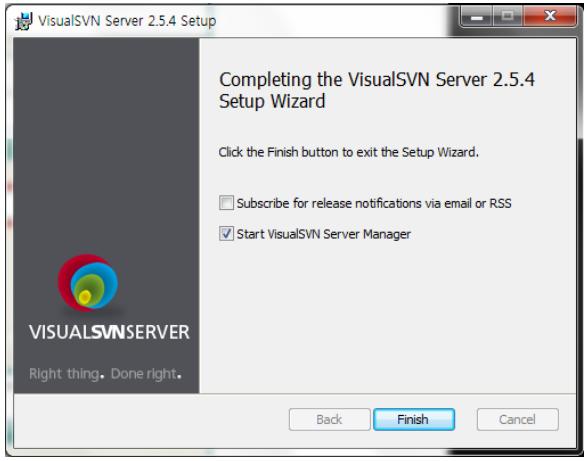

VisualSVN Server 2.5.4 Setup
Welcome to the VisualSVN Server 2.5.4 Setup Wizard
The Setup Wizard will install VisualSVN Server 2.5.4 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
This product includes the following components:
Apache HTTP Server 2.2.22
Apache Subversion 1.7.4
VISUALSVNSERVER
Right thing. Done right.
Back
Next
Cancel


VisualSVN Server 2.5.4 Setup
End-User License Agreement
Please read the following license agreement carefully
License Agreement for VisualSVN Server
IMPORTANT! READ CAREFULLY: THIS IS A LEGAL AGREEMENT. BY DOWNLOADING, INSTALLING, COPYING, SAVING ON YOUR COMPUTER, OR OTHERWISE USING THIS SOFTWARE, YOU (LICENSEE, AS DEFINED BELOW) ARE BECOMING A PARTY TO THIS AGREEMENT AND YOU ARE CONSENTING TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU SHOULD NOT DOWNLOAD, INSTALL AND USE THE SOFTWARE.
I accept the terms in the License Agreement
Back
Next
Cancel

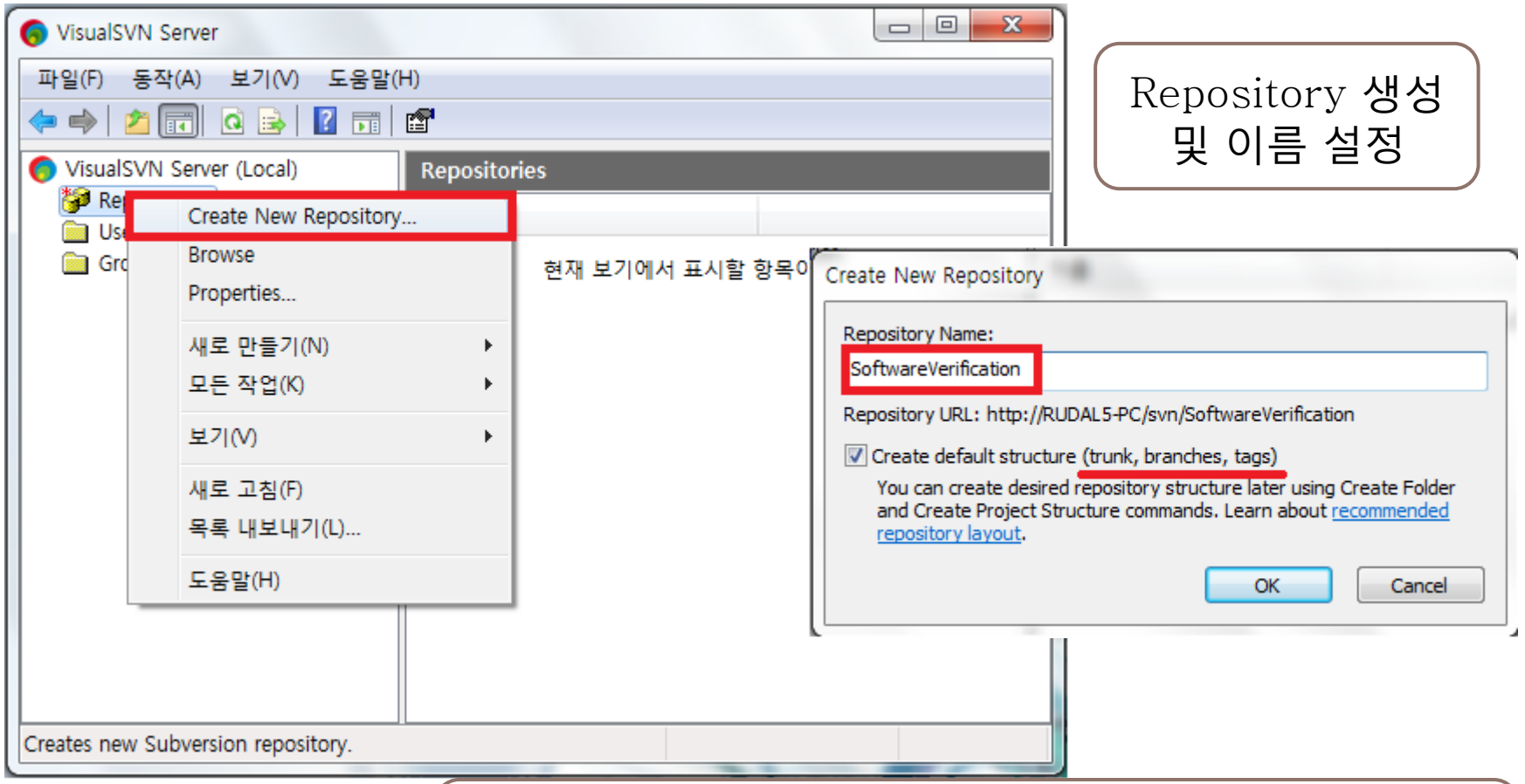

VisualSVN Server 2.5.4 Setup
Select Components
Please select components you would like to install.
VisualSVN Server and Management Console can be installed together or separately.
Please select components you want to be installed on this machine.
VisualSVN Server and Management Console
Install VisualSVN Server and management console to administer it.
Management Console Only
Install only Microsoft management console snap-in to administer VisualSVN Server installed on another computer.
Back
Next
Cancel


VisualSVN Server 2.5.4 Setup
Custom Setup
Select the way you want features to be installed.
Change if necessary installation path and initial VisualSVN Server settings.
Location:
C:₩Program Files₩VisualSVN Server₩
Browse...
Repositories:
C:₩Repositories₩
Browse...
Server Port:
80
Use secure connection (https://)
Authentication
Use Subversion authentication
VisualSVN Server will maintain own users and groups.
Use Windows authentication
VisualSVN Server will use Windows users and groups.
Back
Next
Cancel


VisualSVN Server 2.5.4 Setup
Ready to Install
The Setup Wizard is ready to begin
Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
Back
Install
Cancel


VisualSVN Server 2.5.4 Setup
Completing the VisualSVN Server 2.5.4 Setup Wizard
Click the Finish button to exit the Setup Wizard.
Subscribe for release notifications via email or RSS
Start VisualSVN Server Manager
VISUALSVNSERVER
Right thing. Done right.
Back
Finish
Cancel

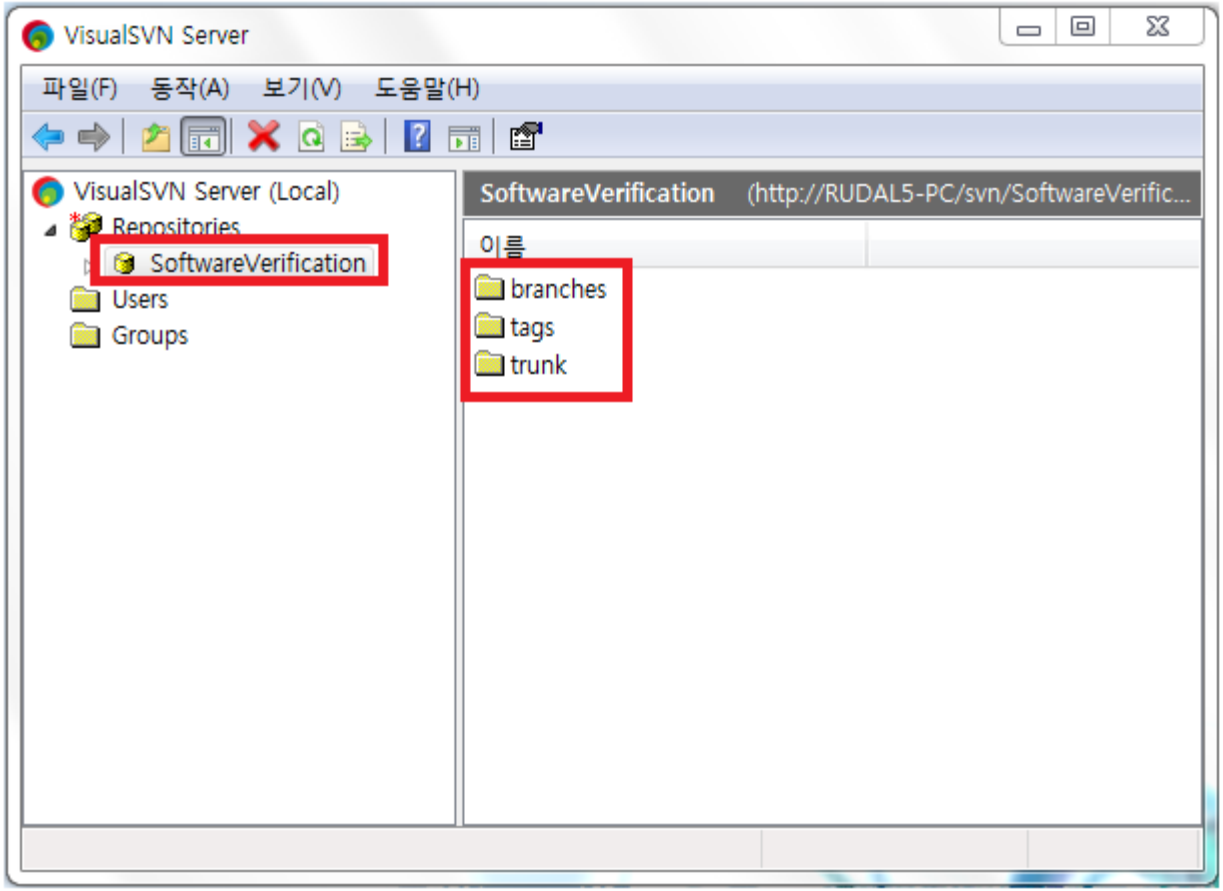
VisualSVN



Repository 생성
및 이름 설정

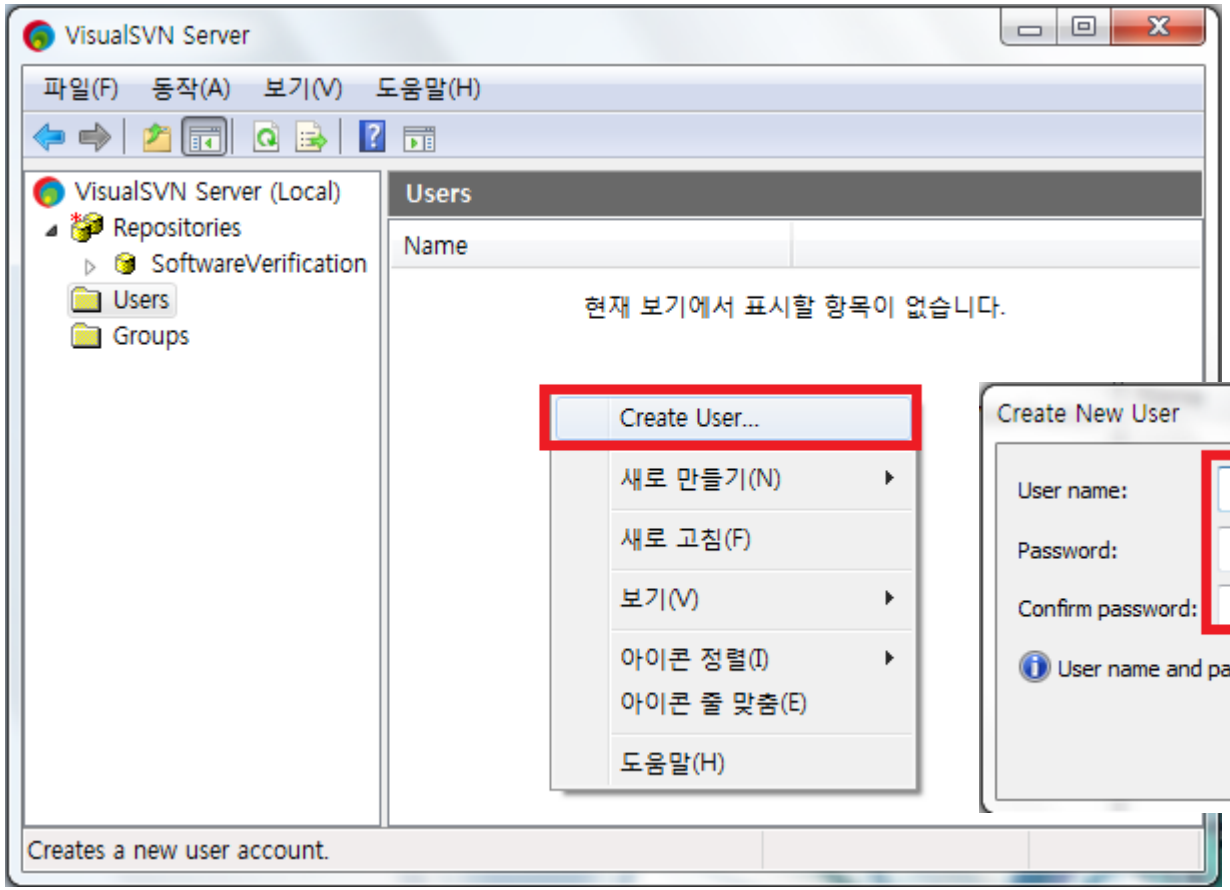
trunk : 현개발되는 가장 최신의 소스를 지정하는 것이 관례
branches : 현 개발되는 가운데 다른 방향으로 개발할 것에 대한 분기
tags : 일종의 버전관리, 구버전들을 모아 계속 업뎃을 해준다

VisualSVN



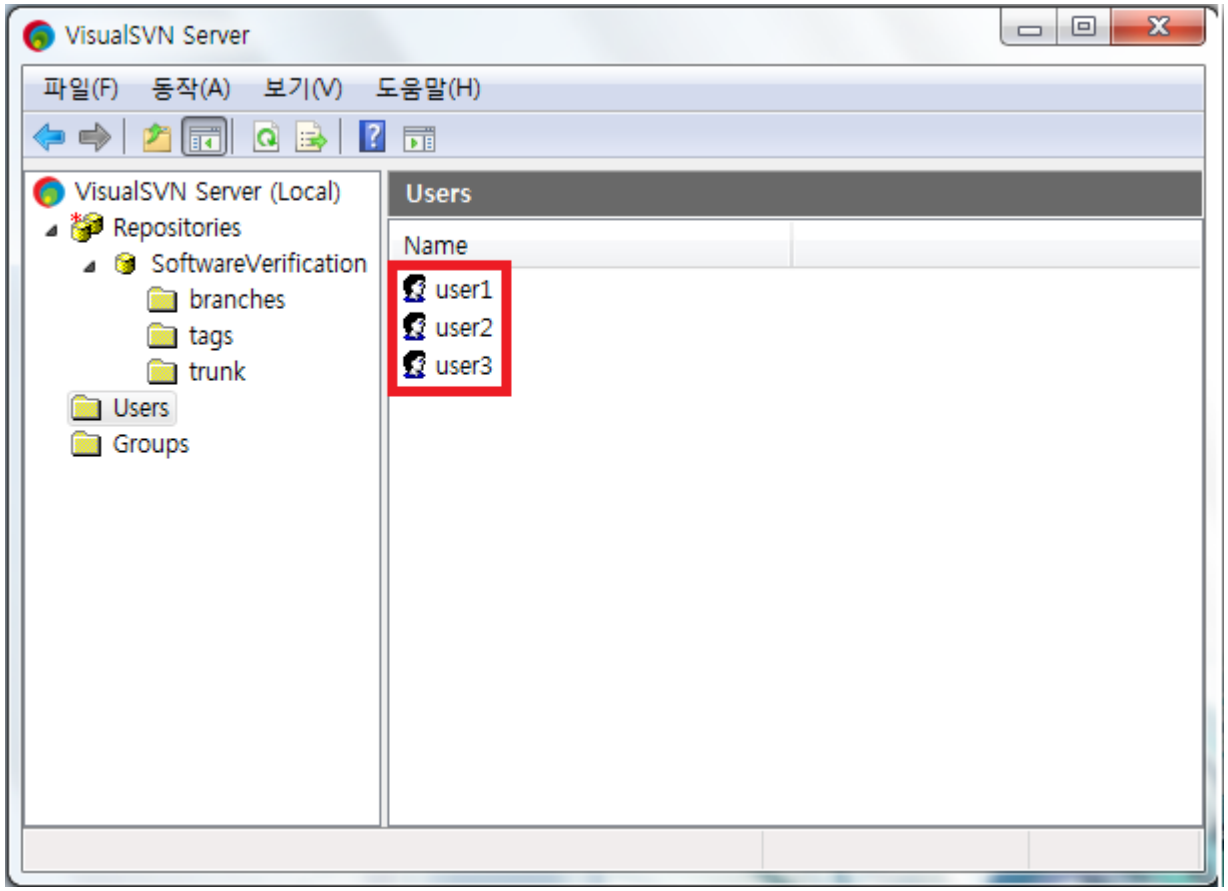
Repository
생성 확인

VisualSVN



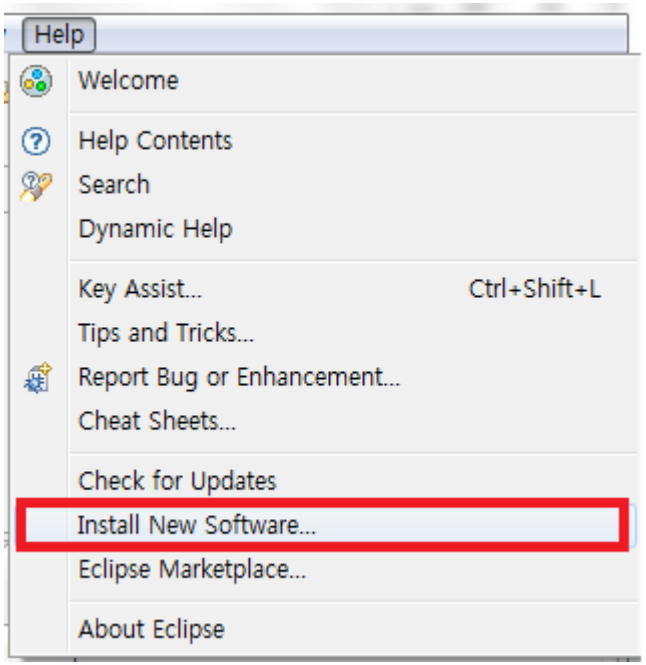
user 생성

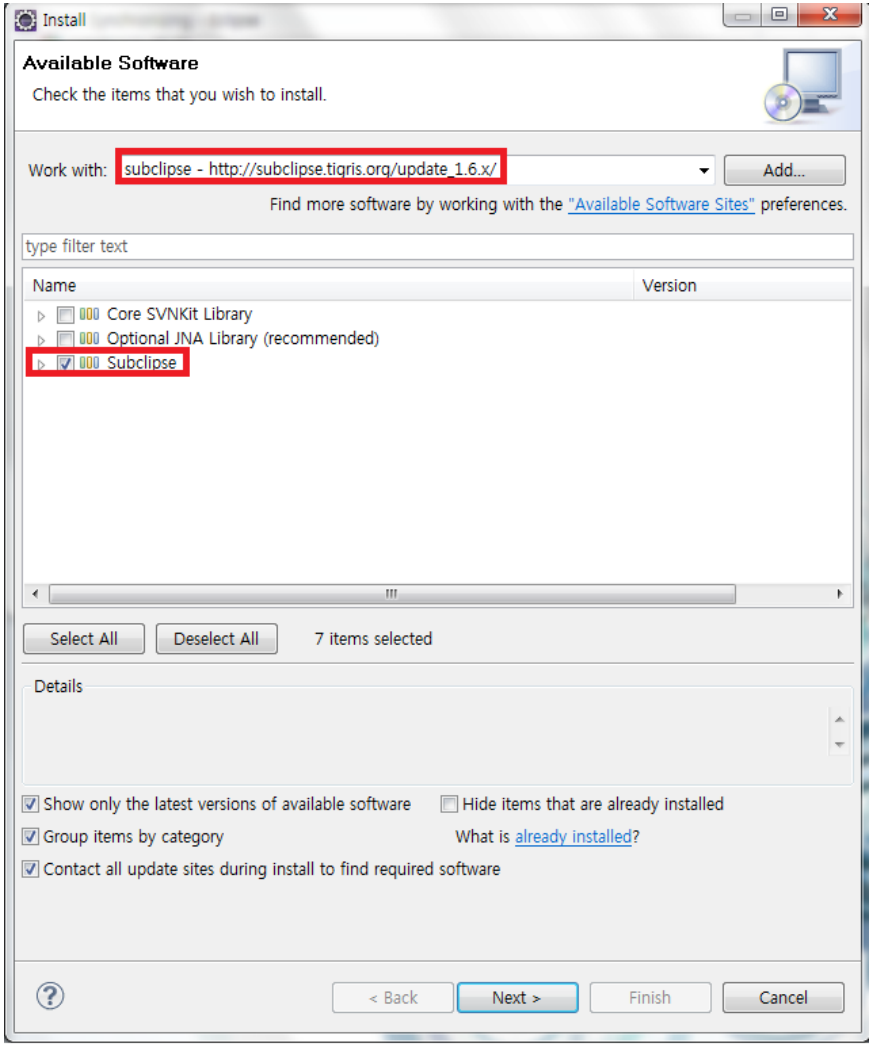
VisualSVN



user 생성 확인

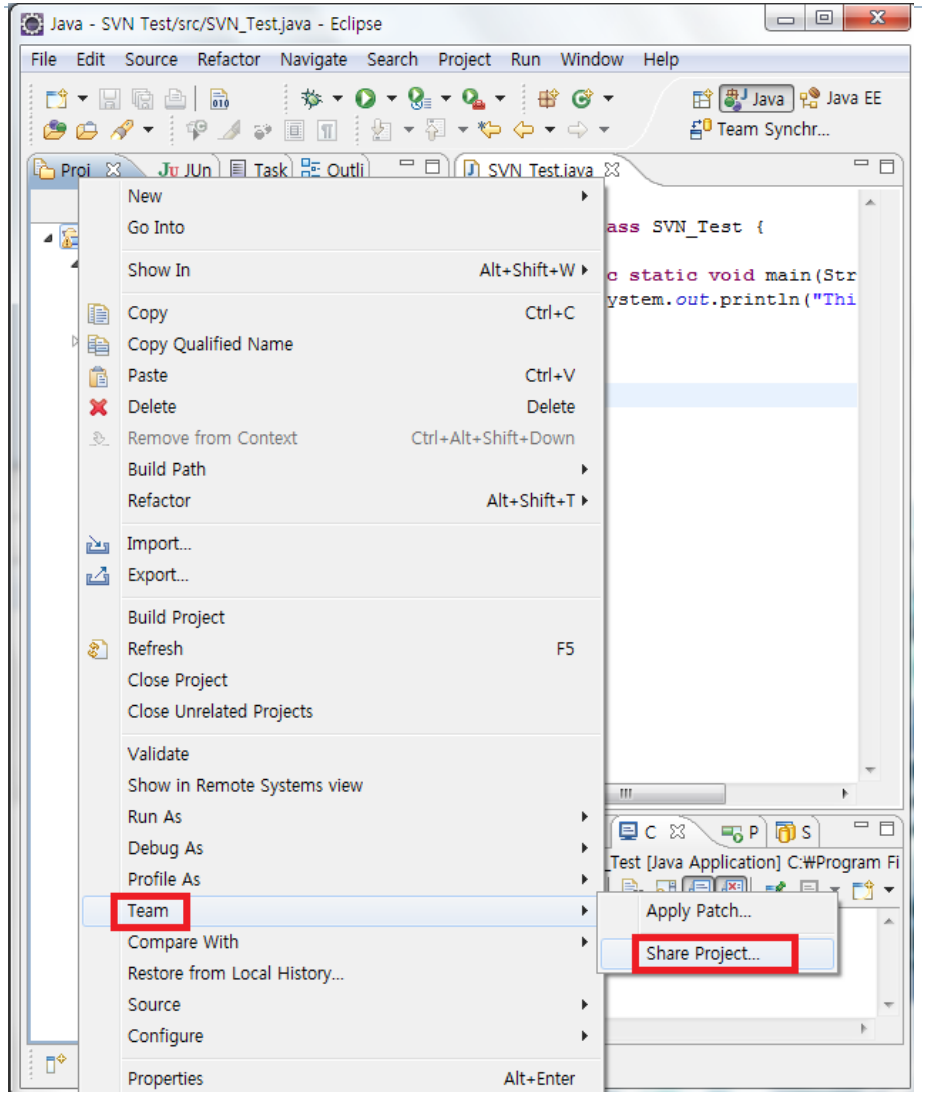
Subclipse

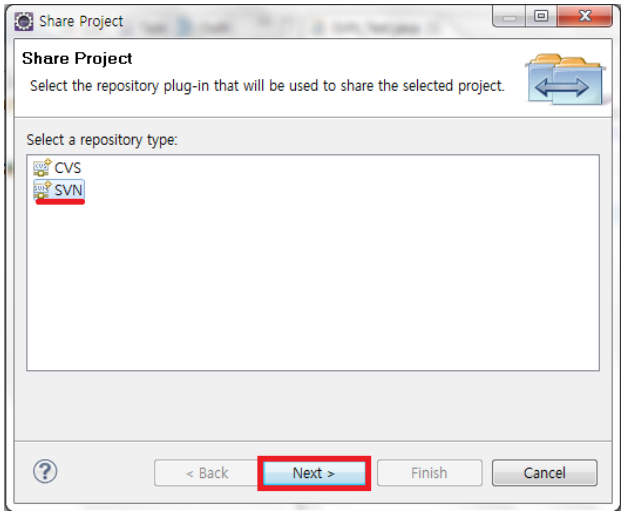


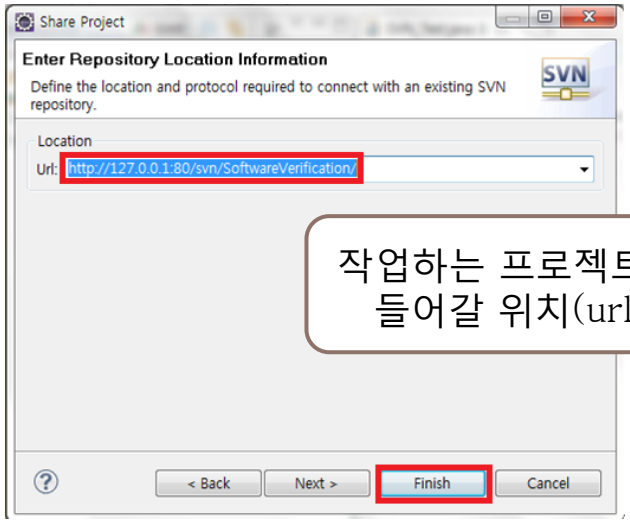


http://subclipse.tigris.org/update_1.6.x/

Subclipse

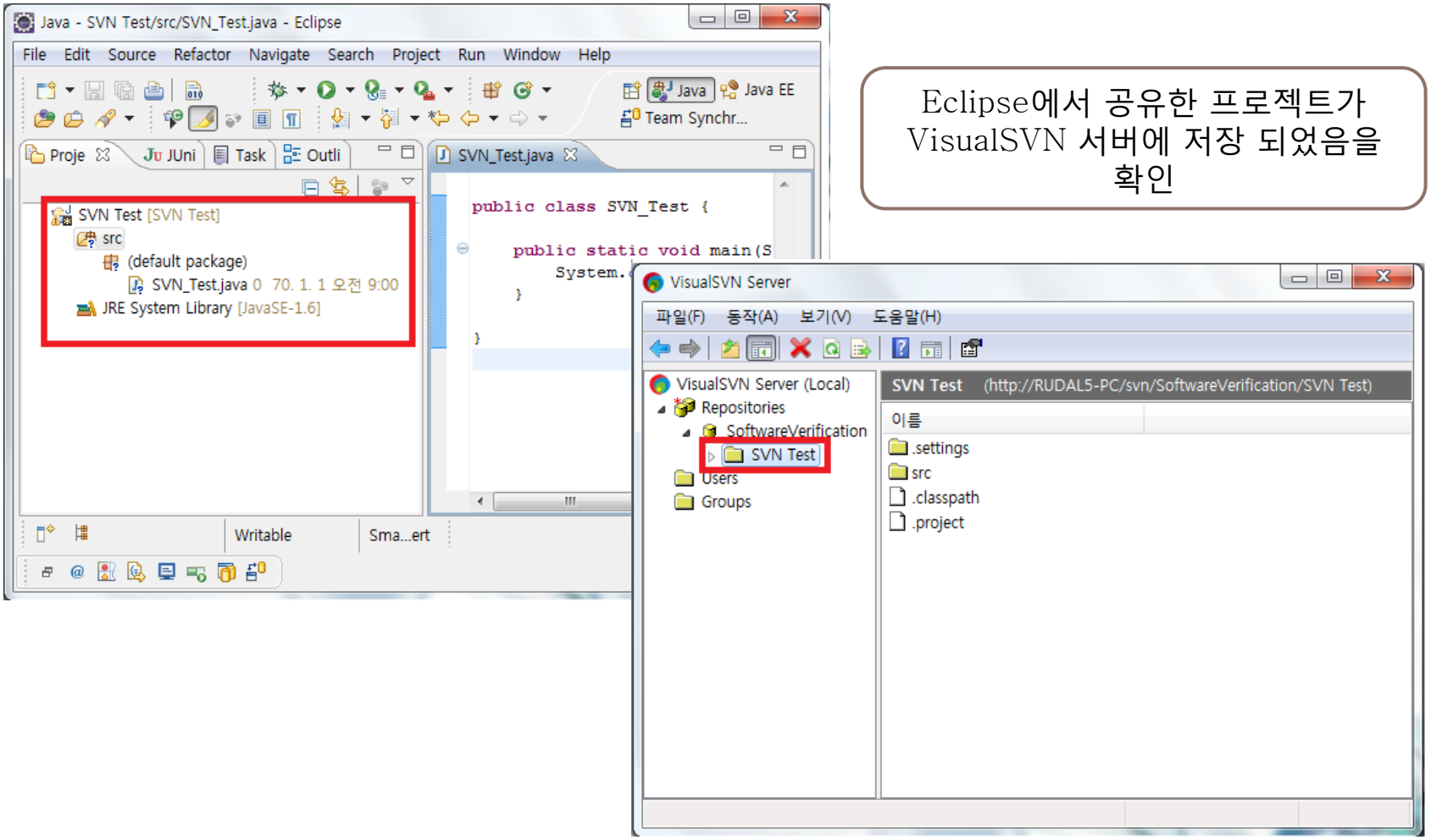






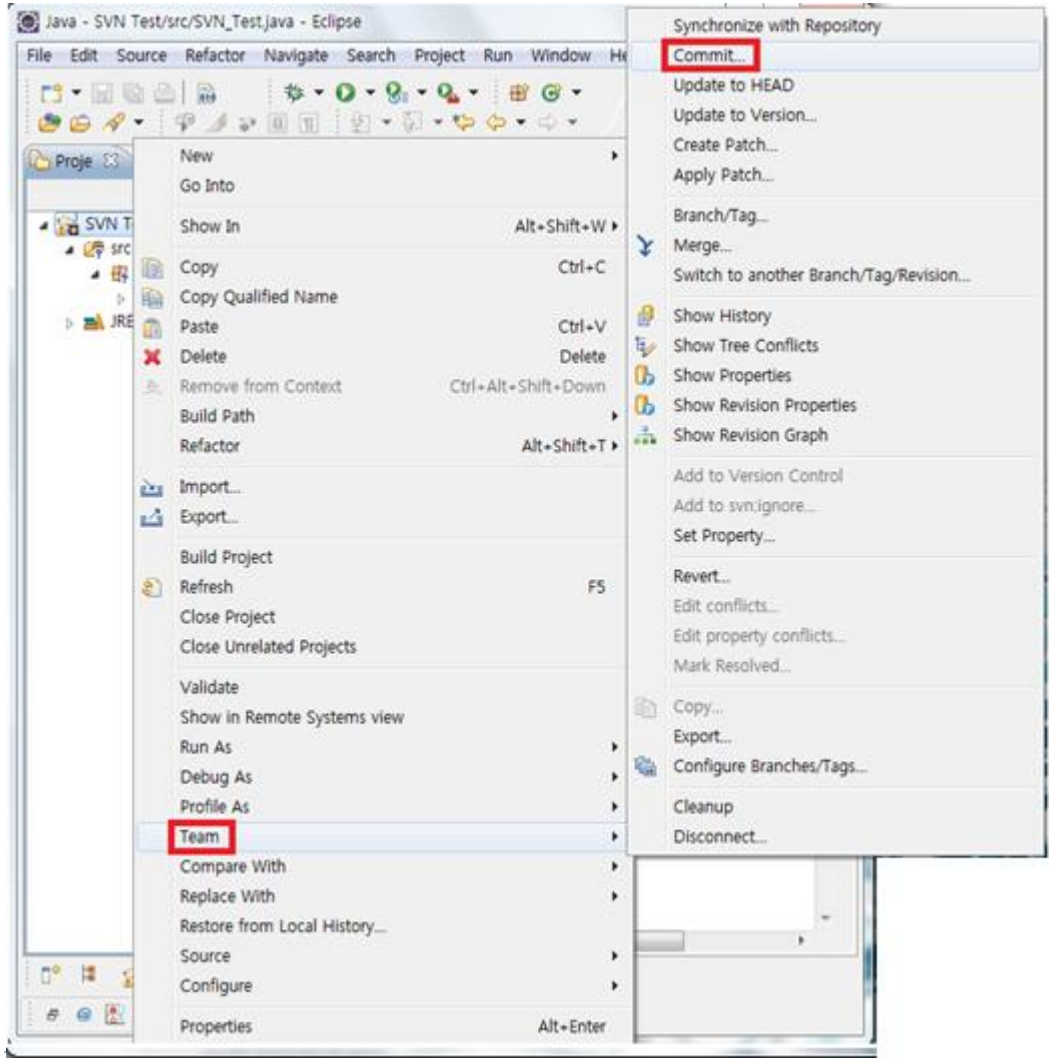
작업하는 프로젝트가 들어갈 위치(url)

Subclipse

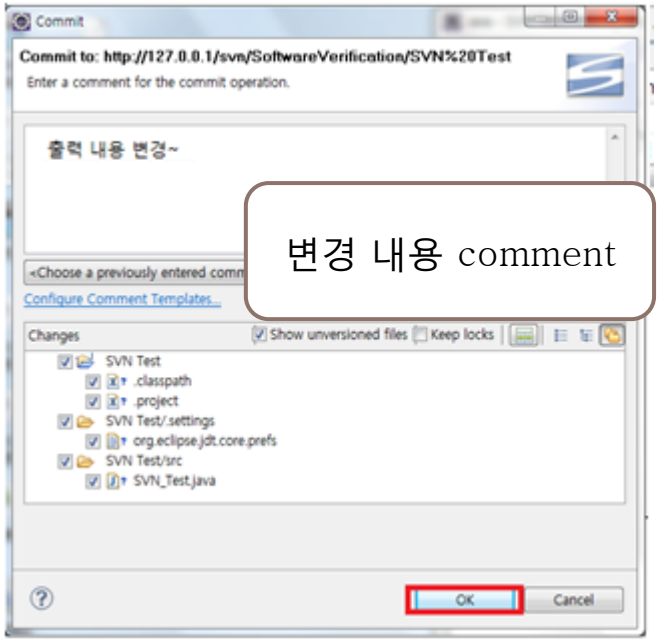


Eclipse에서 공유한 프로젝트가 VisualSVN 서버에 저장 되었음을 확인

Subclipse

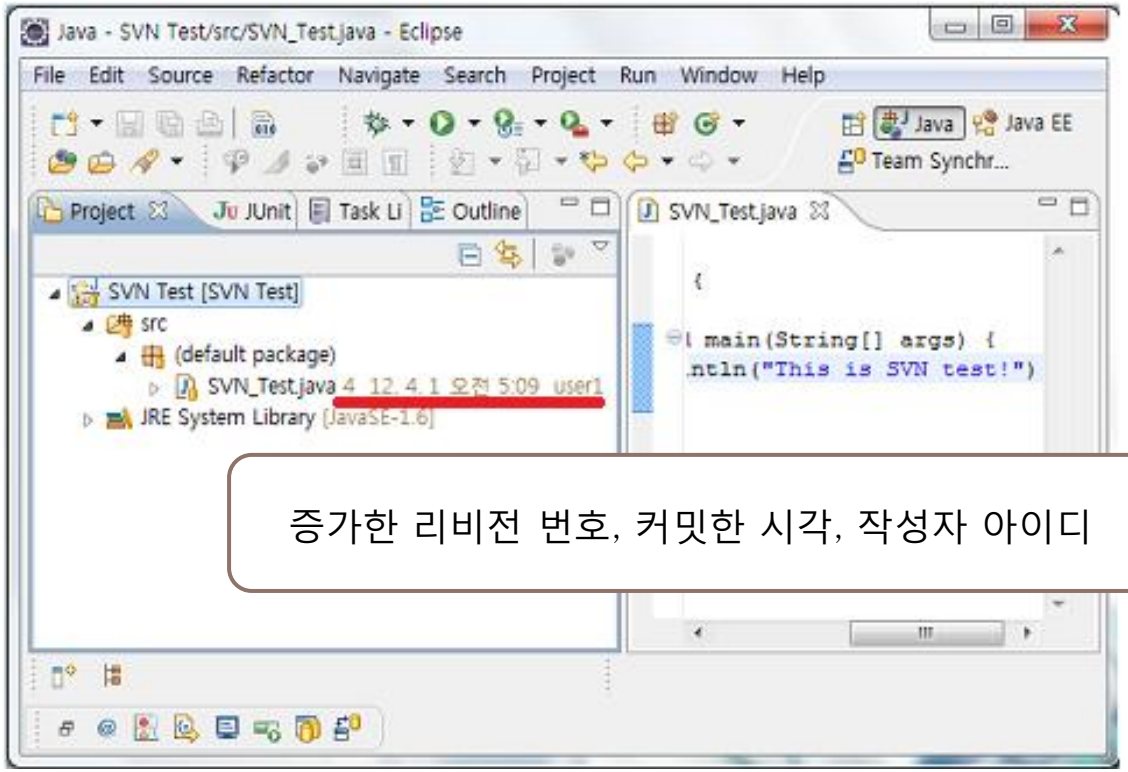


코드를 수정한 뒤 commit



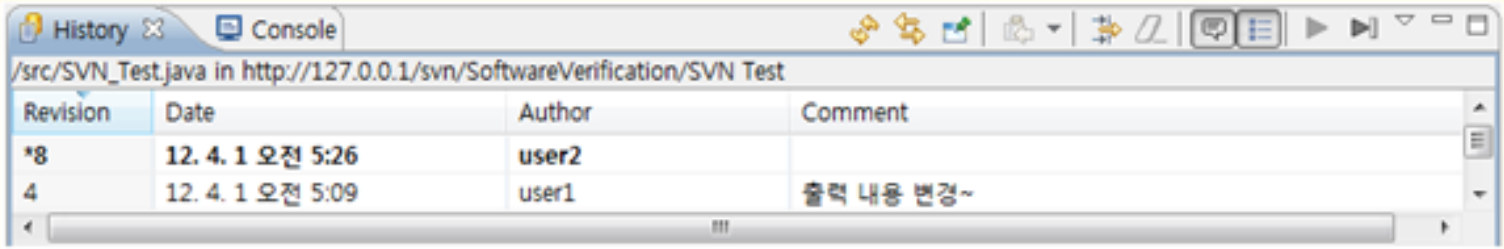
변경 내용 comment

Subclipse

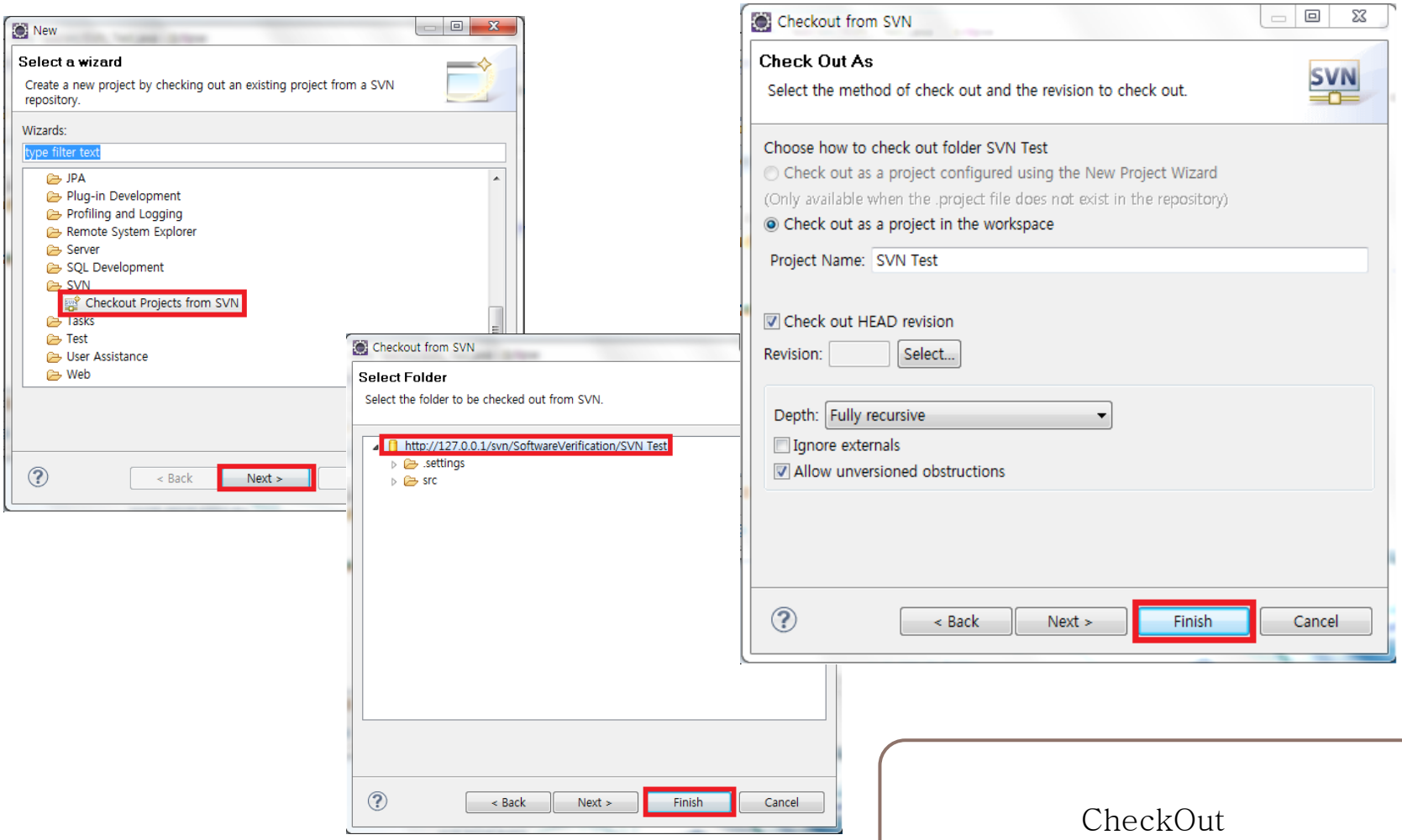


증가한 리비전 번호, 커밋한 시각, 작성자 아이디

history 뷰에서
확인 가능

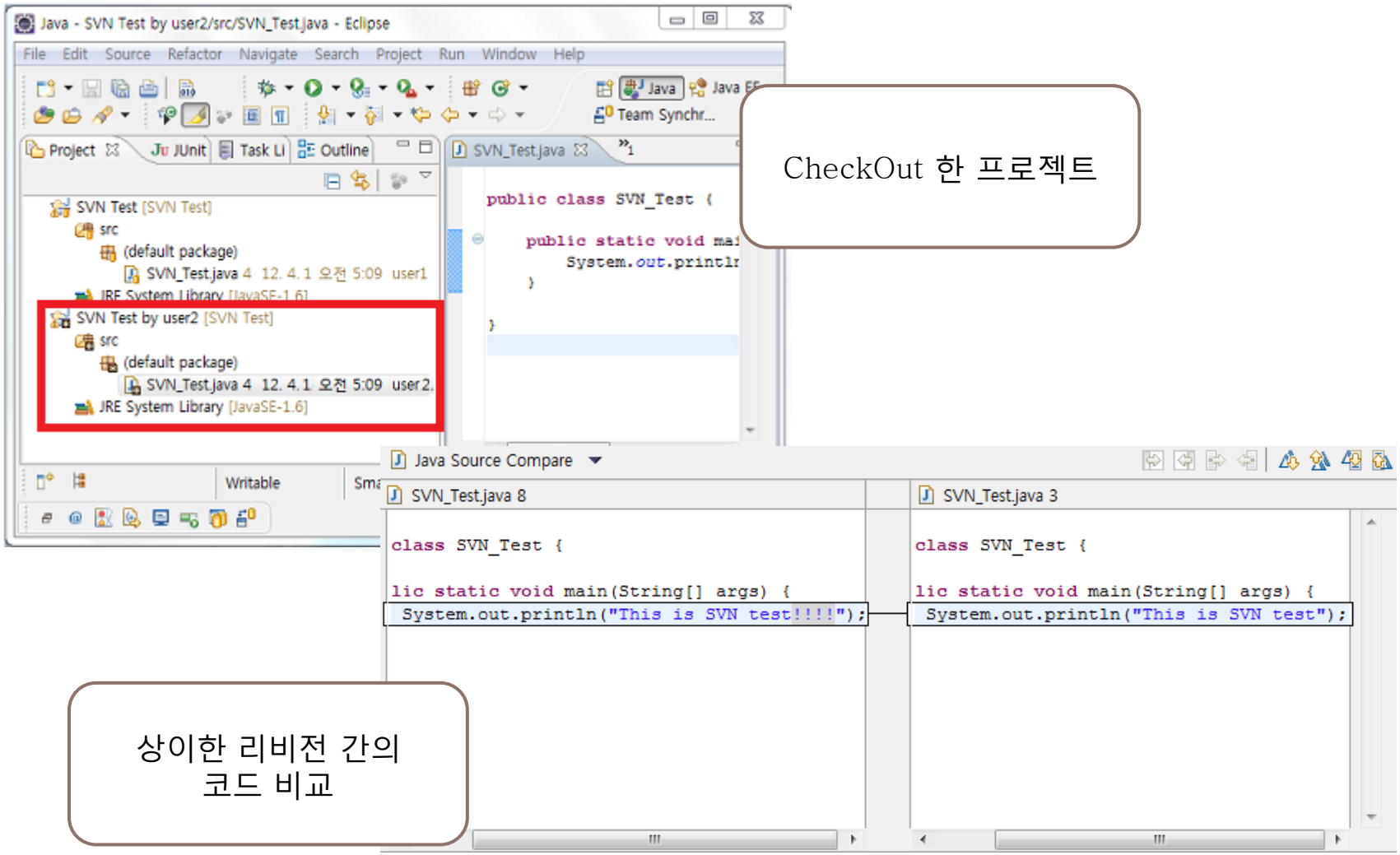


Subclipse



CheckOut

Subclipse


Java - SVN Test by user2/src/SVN_Test.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project JUnit Task Li Outline
SVN_Test.java
SVN Test [SVN Test]
src
(default package)
SVN_Test.java 4 12. 4. 1 오전 5:09 user1
JRE System Library [JavaSE-1.6]
SVN Test by user2 [SVN Test]
src
(default package)
SVN_Test.java 4 12. 4. 1 오전 5:09 user2
JRE System Library [JavaSE-1.6]
public class SVN_Test {
public static void mai
System.out.printlr
}
}
Writable
CheckOut 한 프로젝트
Java Source Compare
SVN_Test.java 8
SVN_Test.java 3
class SVN_Test {
lic static void main(String[] args) {
System.out.println("This is SVN test!!!!");
class SVN_Test {
lic static void main(String[] args) {
System.out.println("This is SVN test");
상이한 리비전 간의
코드 비교

Git

- 분산 버전 관리 시스템

- Git의 장점
  - 분산 아키텍처
    -> 완전히 연결이 끊어진 상태에서, 항상 인터넷에 연결 돼야하는 고통 없이 동작한다.
  - 쉬운 브랜치 생성과 merge
    -> 다른 버전 관리 시스템과는 달리 브랜치를 생성하기가 쉽고, 비용이 적으며, 속도도 빠르다. 또한 브랜치를 여러 번 나눈 경우라도 간단히 합칠 수 있다.
  - 서브 버전과 통신
    -> 자신만이 회사에서 Git로 바꿀 준비가 된 유일한 사람이라면? 모두가 서브버전을 계속 사용하고 있어도 걱정할 필요가 없다. Git는 서브버전 저장소의 모든 이력을 가져올 수 있으며, 변경 사항을 다시 보낼 수도 있다.

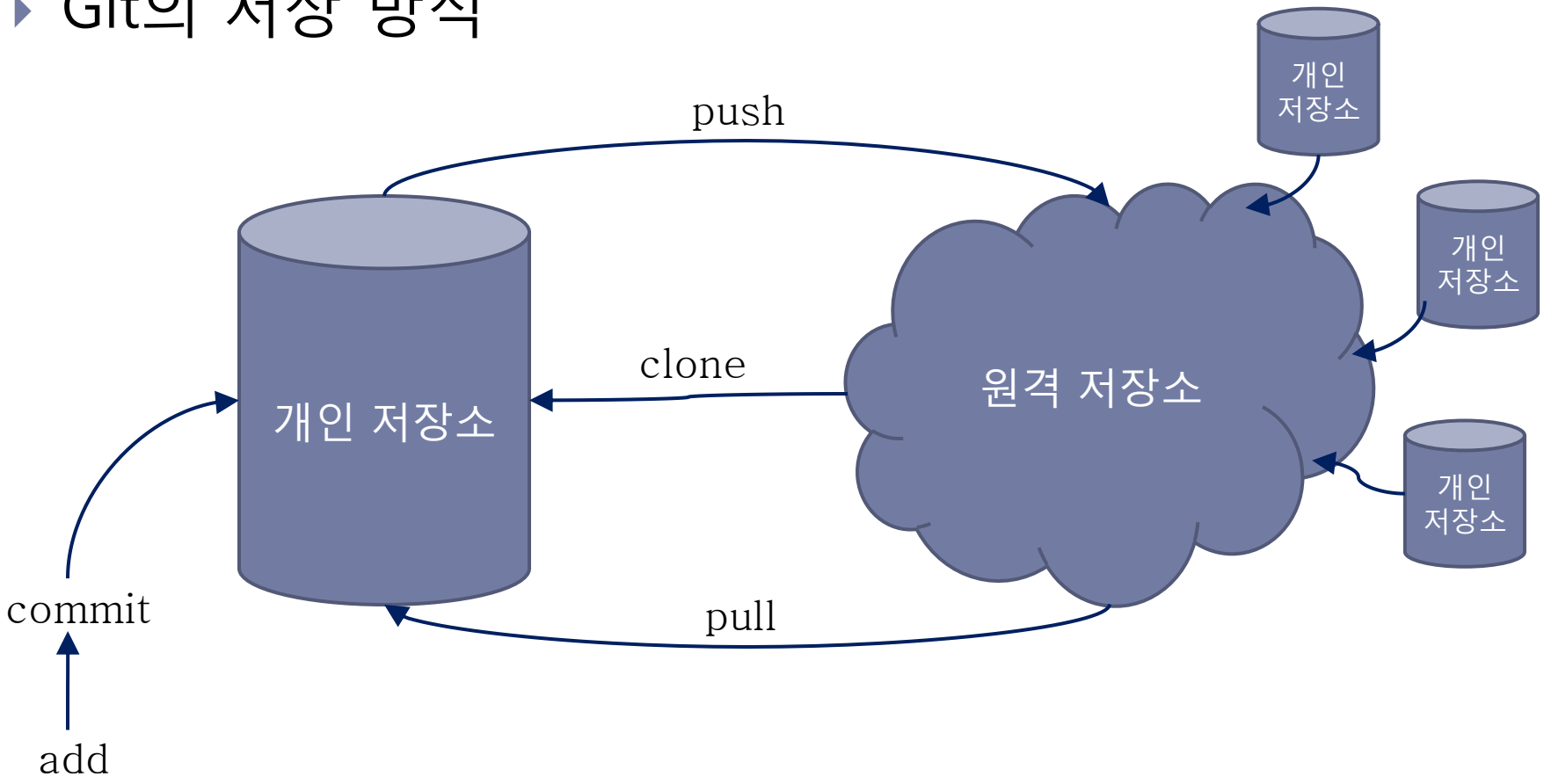
Git

- Terms of Git
  - Clone - 원격 저장소에 저장된 파일들을 개인의 저장소로 복제
  - Commit - 파일을 저장소에 저장
  - Add - commit하기 위해 변경사항을 스테이징
  - 스테이징 - 관리목록을 갱신 스테이징 영역은 저장소에 넣기 전 일종의 버퍼에 저장하는 것이다.

    -> 중복 저장(commit, add) 하는 이유 : 변경사항을 commit하기 전에 파일을 정교하게 다듬을 수 있는 기회를 제공
  - Push - 개인의 저장소에 저장된 파일을 원격저장소에 저장
  - Pull - 원격 저장소에 저장된 파일을 개인의 저장소로 가져옴
  - Branch - 분기 이력을 만들어 관리
  - Merge - 분기된 파일을 통합

Git

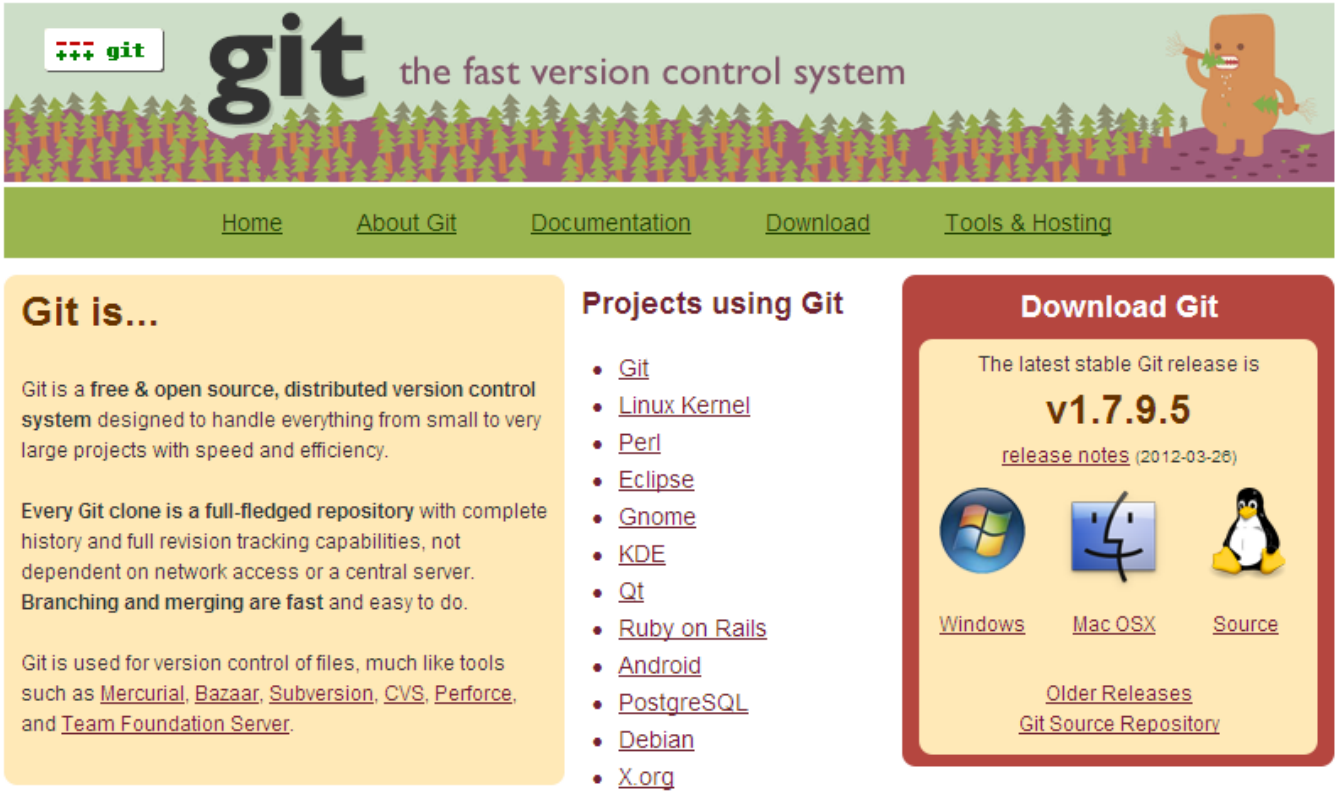
- Git의 저장 방식



Git

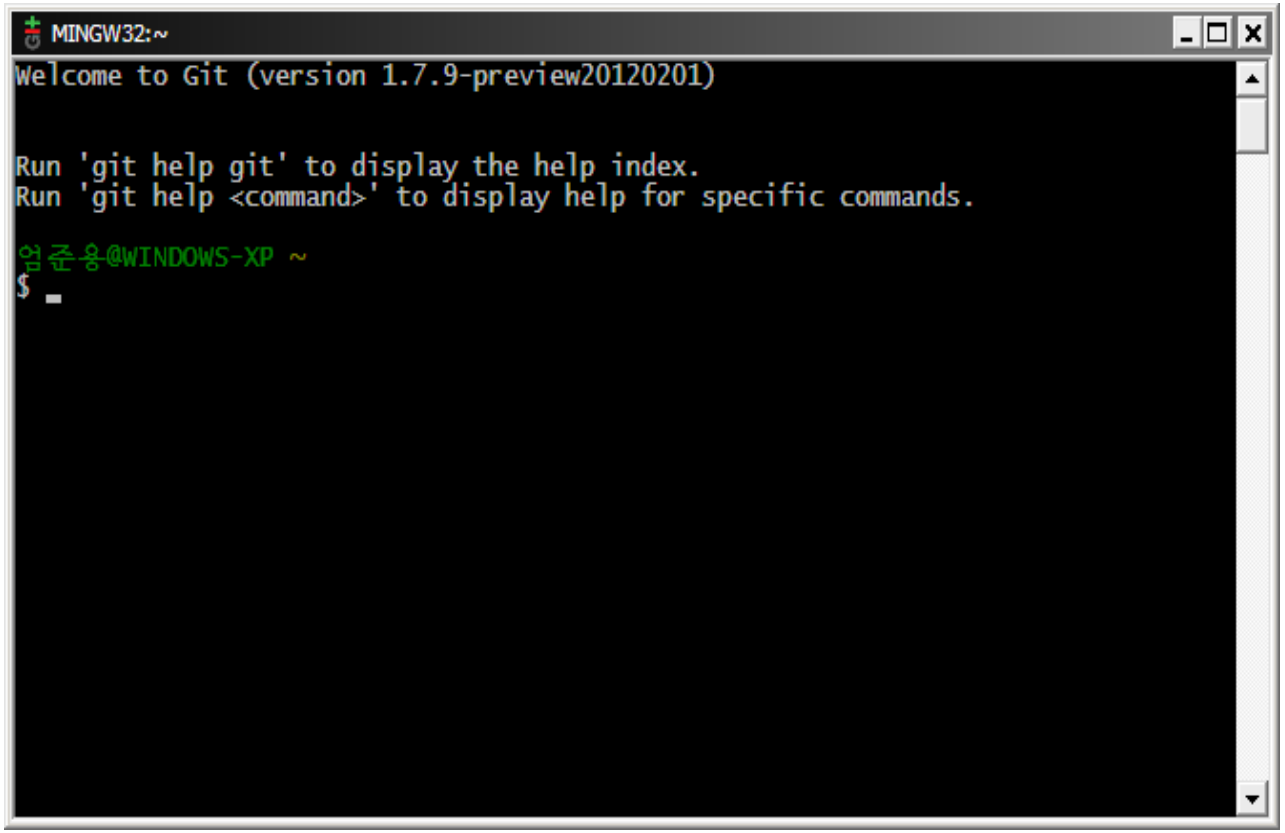
- Installation

  http://git-scm.com/



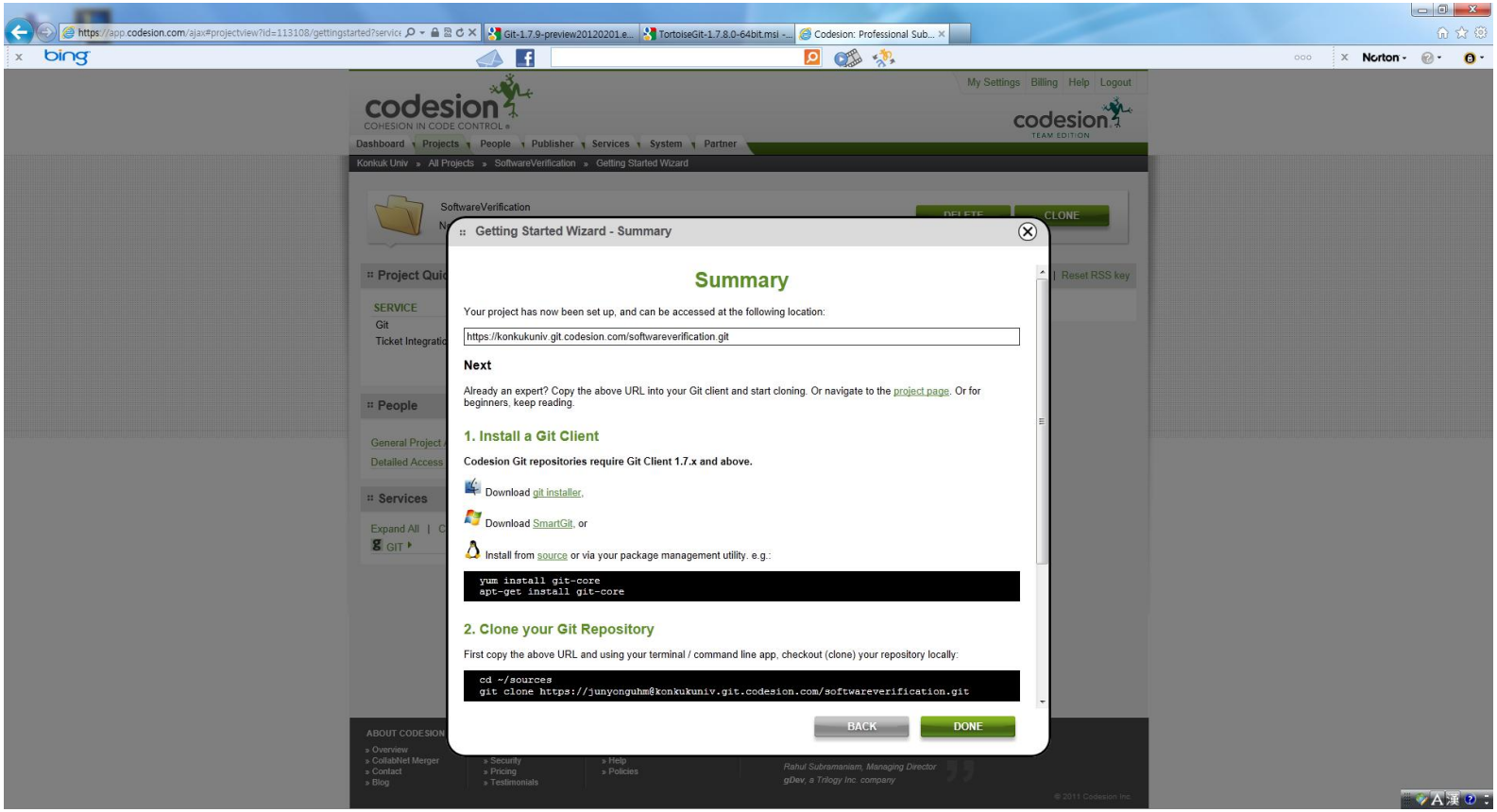
Git

- Installation



Git

- 원격 저장소 생성
  - http://blog.codesion.com/



Git

- Git 명령어 요약
  - 설정과 초기화
    - git config --global user.name [사용자명] : 전역 사용자명
    - git config --global user.email [이메일] : 이메일 구성하기

```
엄준용@WINDOWS-XP ~
$ git config --global user.name "UMJA"

엄준용@WINDOWS-XP ~
$ git config --global user.email "oursoon2@naver.com"
```

    - git init : 새로운 저장소 초기화하기

```
엄준용@WINDOWS-XP ~
$ mkdir gitprac

엄준용@WINDOWS-XP ~
$ cd gitprac

엄준용@WINDOWS-XP ~/gitprac
$ git init
Initialized empty Git repository in c:/Documents and Settings/?/gitprac/.git/
warning: Your console font probably doesn't support Unicode. If you experience s
trange characters in the output, consider switching to a TrueType font such as L
ucida Console!
```

Git

- Git 명령어 요약
  - 설정과 초기화
    - git clone [원격저장소 url] : 저장소 복제하기

```
엄준용@WINDOWS-XP ~/gitprac (RB_1.0.1)
$ git clone https://konkukuniv.git.codesion.com/softwareverification.git
Cloning into 'softwareverification'...
Username for 'https://konkukuniv.git.codesion.com':
Password for 'https://junyonguhm@konkukuniv.git.codesion.com':
warning: You appear to have cloned an empty repository.
```

    - git remote add [디렉토리] [원격저장소 url] : 새원격 저장소 추가

```
엄준용@WINDOWS-XP ~/gitprac (RB_1.0.1)
$ git remote add gitprac-remote https://konkukuniv.git.codesion.com/softwarever
ification.git
```

    - git config –global –list : 전역 사용자 리스트 확인

```
엄준용@WINDOWS-XP ~
$ git config --global --list
user.name=UMJA
user.email=oursoon2@naver.com
```

Git

- Git 명령어 요약
  - 기본적 사용법
    - git add [파일명] : 새로운 파일 추가 or 스테이징
    - git commit –m [이력 메시지] : commit하기

```
엄준용@WINDOWS-XP ~/gitprac (master)
$ git add index.html

엄준용@WINDOWS-XP ~/gitprac (master)
$ git commit -m "add in git practice HTML"
[master (root-commit) de5d2ed] add in git practice HTML
 1 files changed, 5 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

    - git add -i : Add명령에서 git 대화 모드를 사용하여 파일 추가하기

```
엄준용@WINDOWS-XP ~/gitprac (RB_1.0.1)
$ git add -i
           staged     unstaged path
  1:    unchanged        +1/-1 index.html

*** Commands ***
  1: status       2: update       3: revert       4: add untracked
  5: patch        6: diff         7: quit         8: help
What now> 2
           staged     unstaged path
  1:    unchanged        +1/-1 index.html
Update>> 1
           staged     unstaged path
* 1:    unchanged        +1/-1 index.html
Update>>
updated one path
```

Git

- Git 명령어 요약
  - 기본적 사용법
    - git branch [브랜치명] :새로운 브랜치 생성
    - git branch : 브랜치 리스트 보기

```
엄준용@WINDOWS-XP ~/gitprac (master)
$ git branch contaxt

엄준용@WINDOWS-XP ~/gitprac (master)
$ git branch
  RB_1.0.1
  alternate
  contact
  contaxt
* master
  new
```

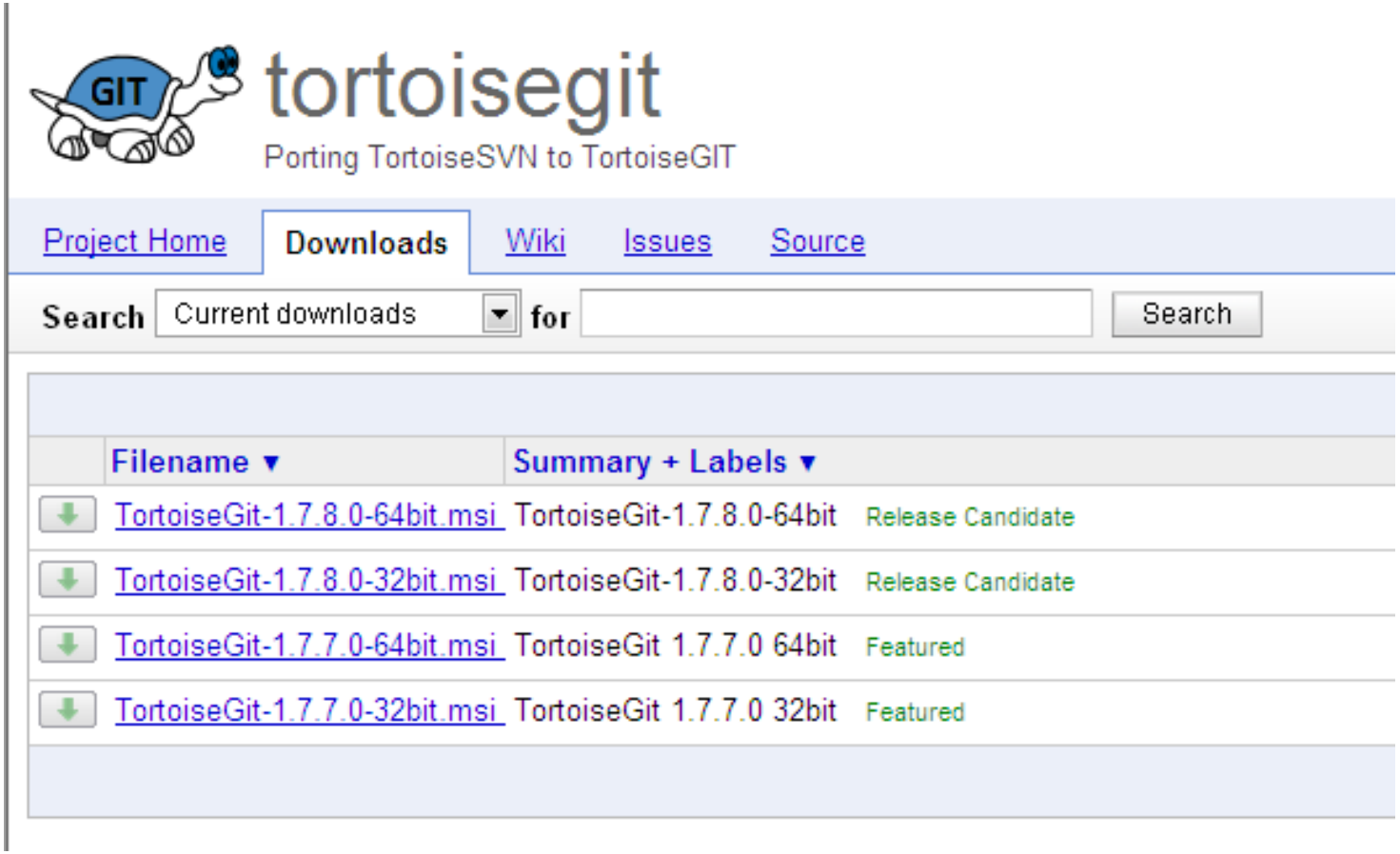
    - git chekout [브랜치명] : 다른 브랜치 체크아웃하기

```
엄준용@WINDOWS-XP ~/gitprac (master)
$ git branch
  RB_1.0.1
  alternate
  contact
  contaxt
* master
  new

엄준용@WINDOWS-XP ~/gitprac (master)
$ git checkout contact
Switched to branch 'contact'

엄준용@WINDOWS-XP ~/gitprac (contact)
$ git branch
  RB_1.0.1
  alternate
* contact
  contaxt
  master
  new
```

Tortoisegit

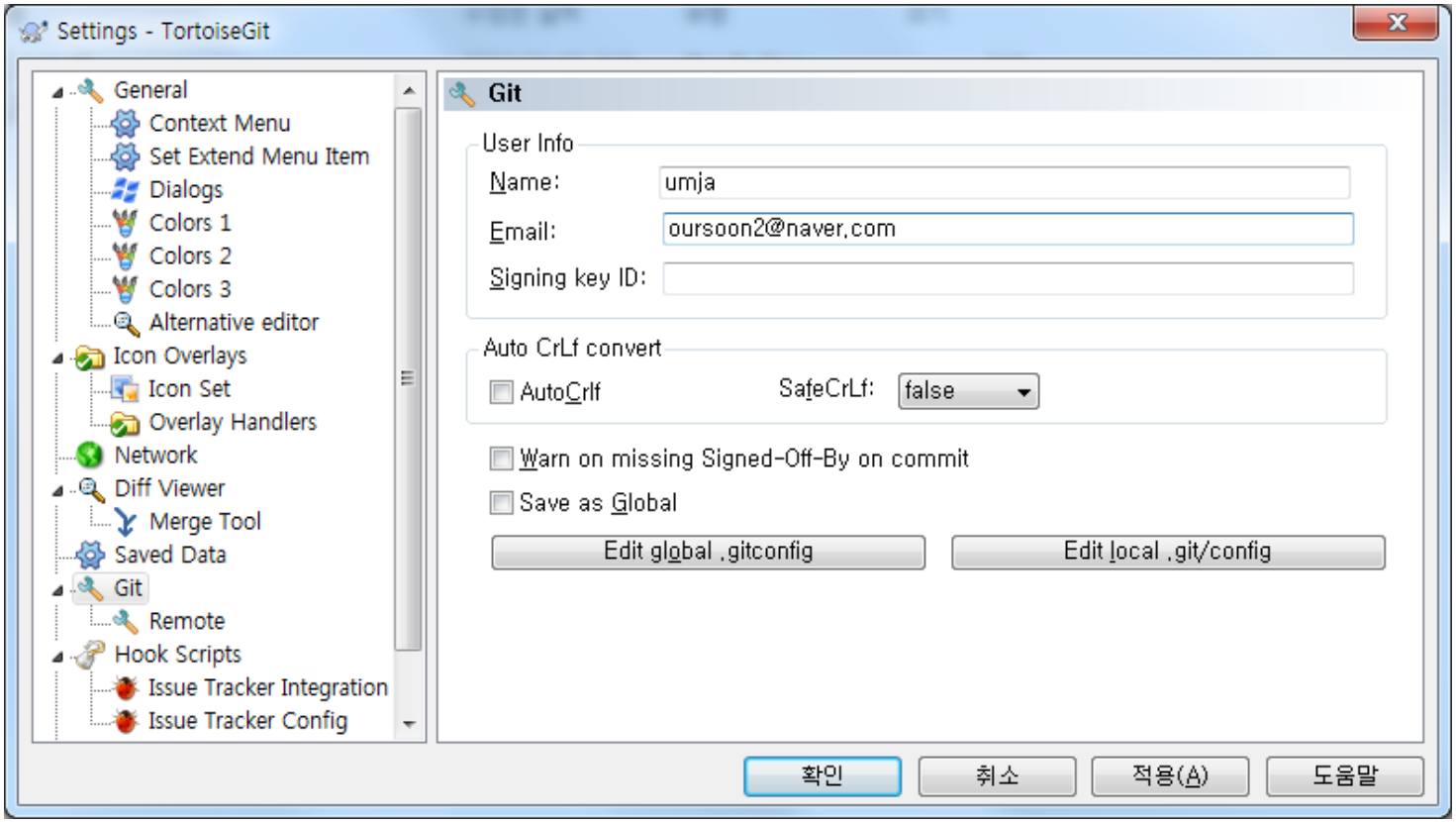
- Installation

  http://code.google.com/p/tortoisegit/



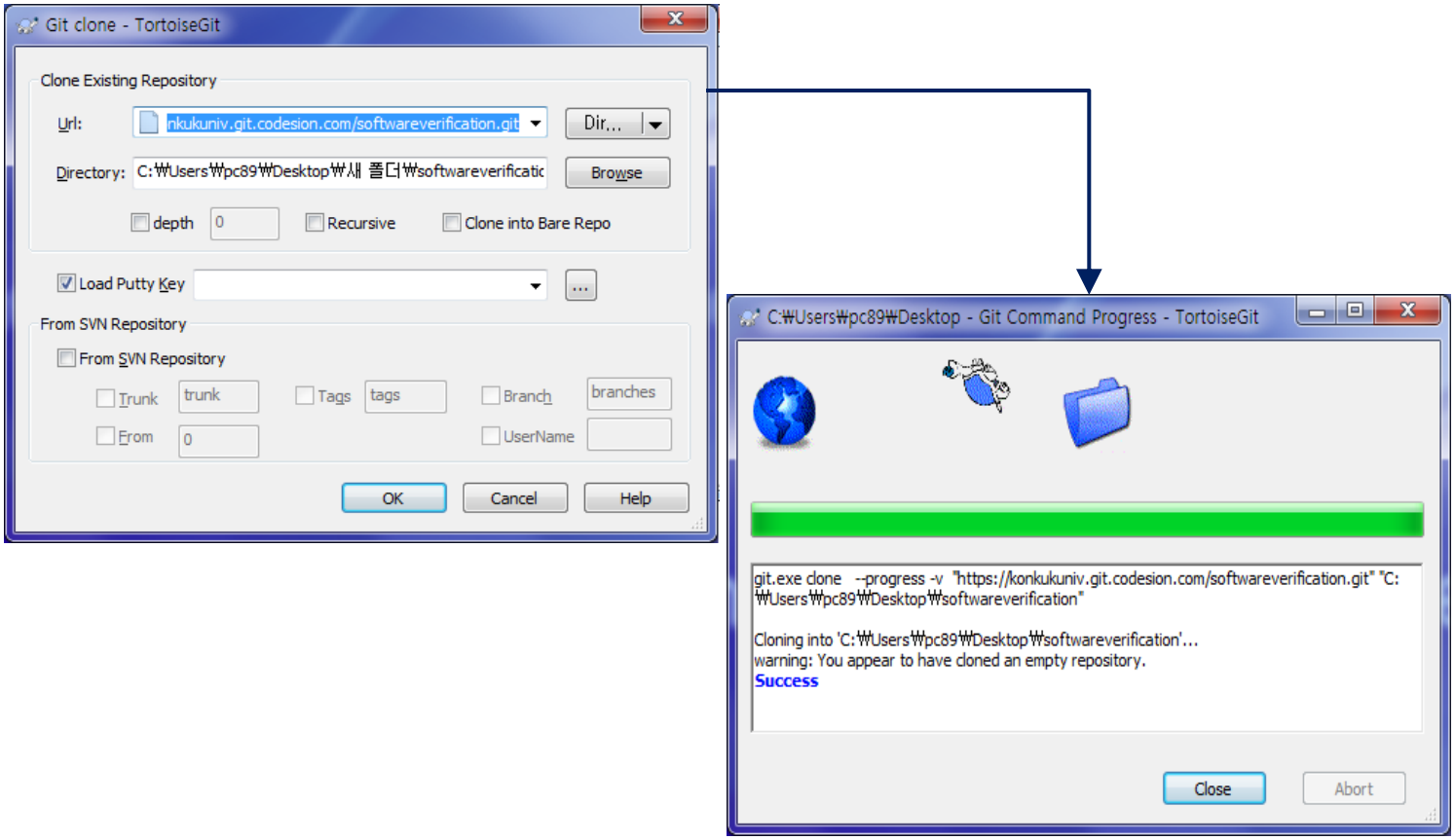
Tortoisegit

- Usage
  - 사용자명 / 이메일 설정



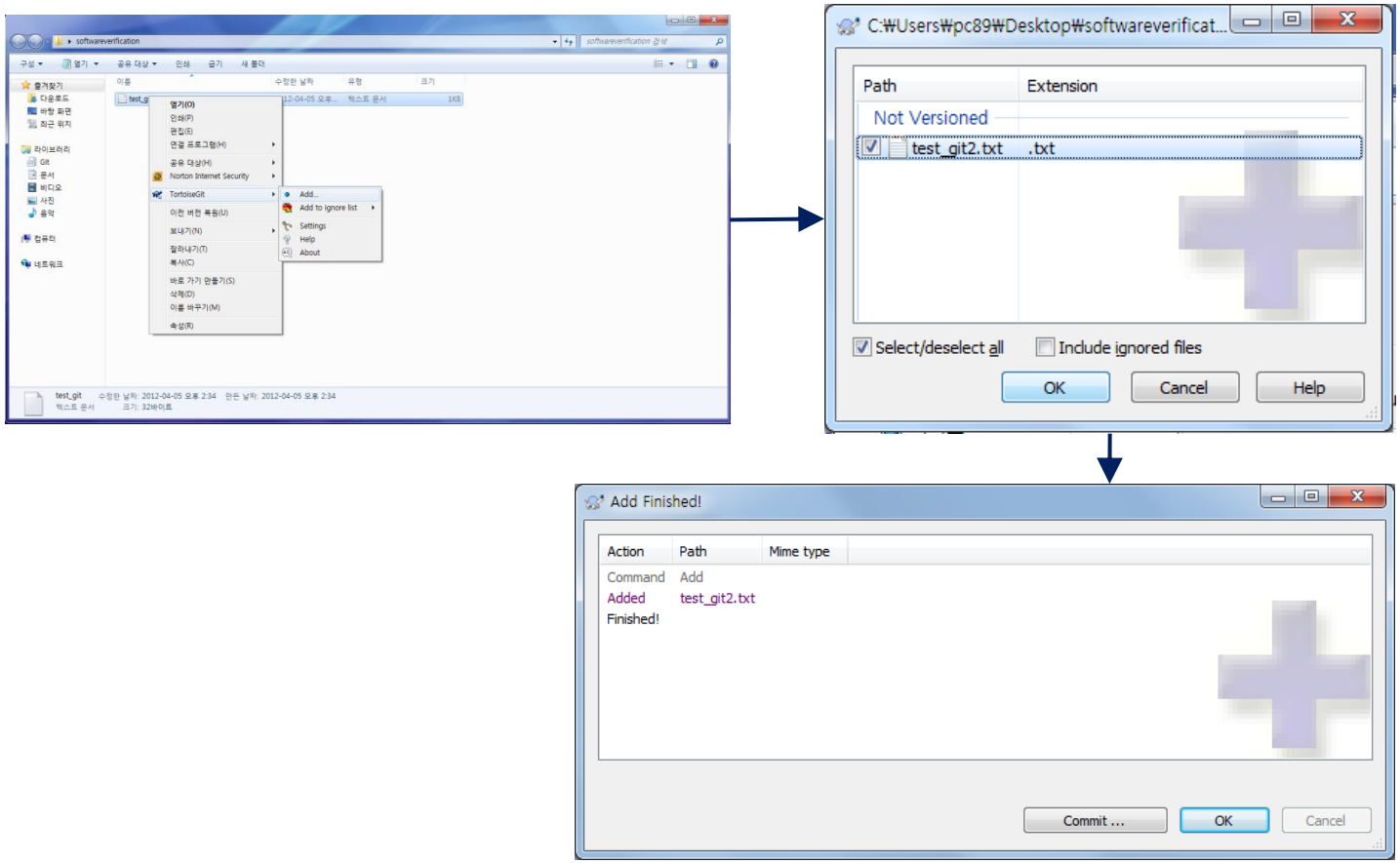
Tortoisegit

- Usage
  - Git clone



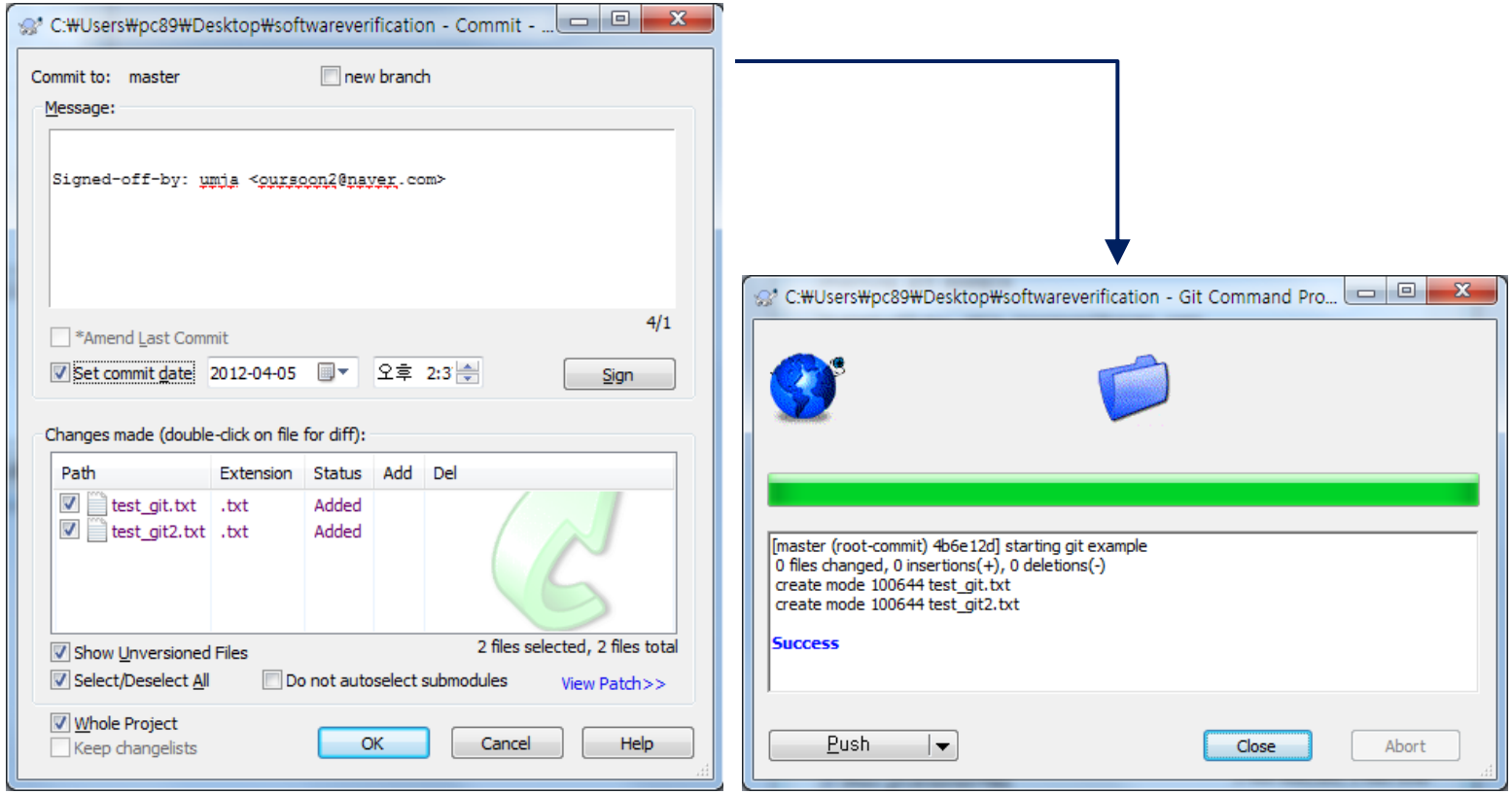
Tortoisegit

- Usage
  - Git add



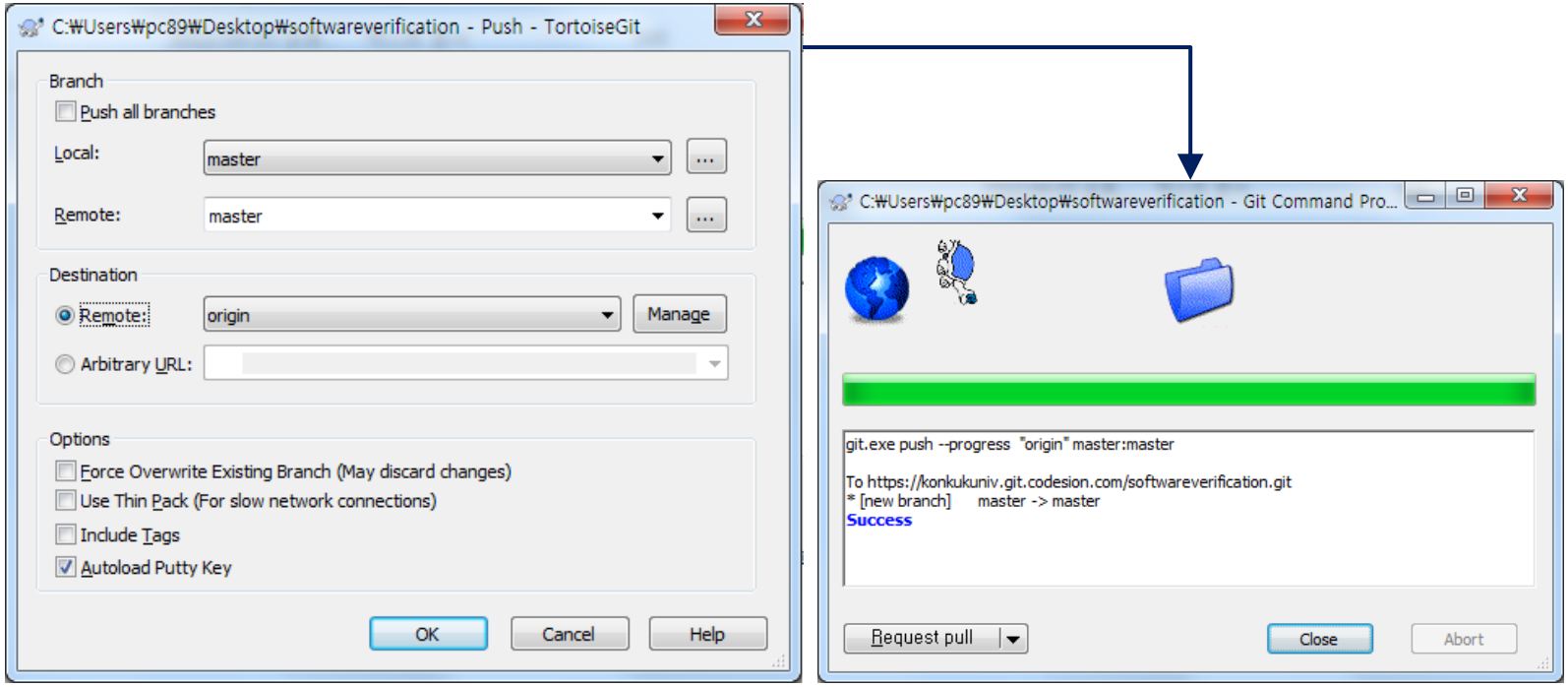
Tortoisegit

- Usage
  - Git commit



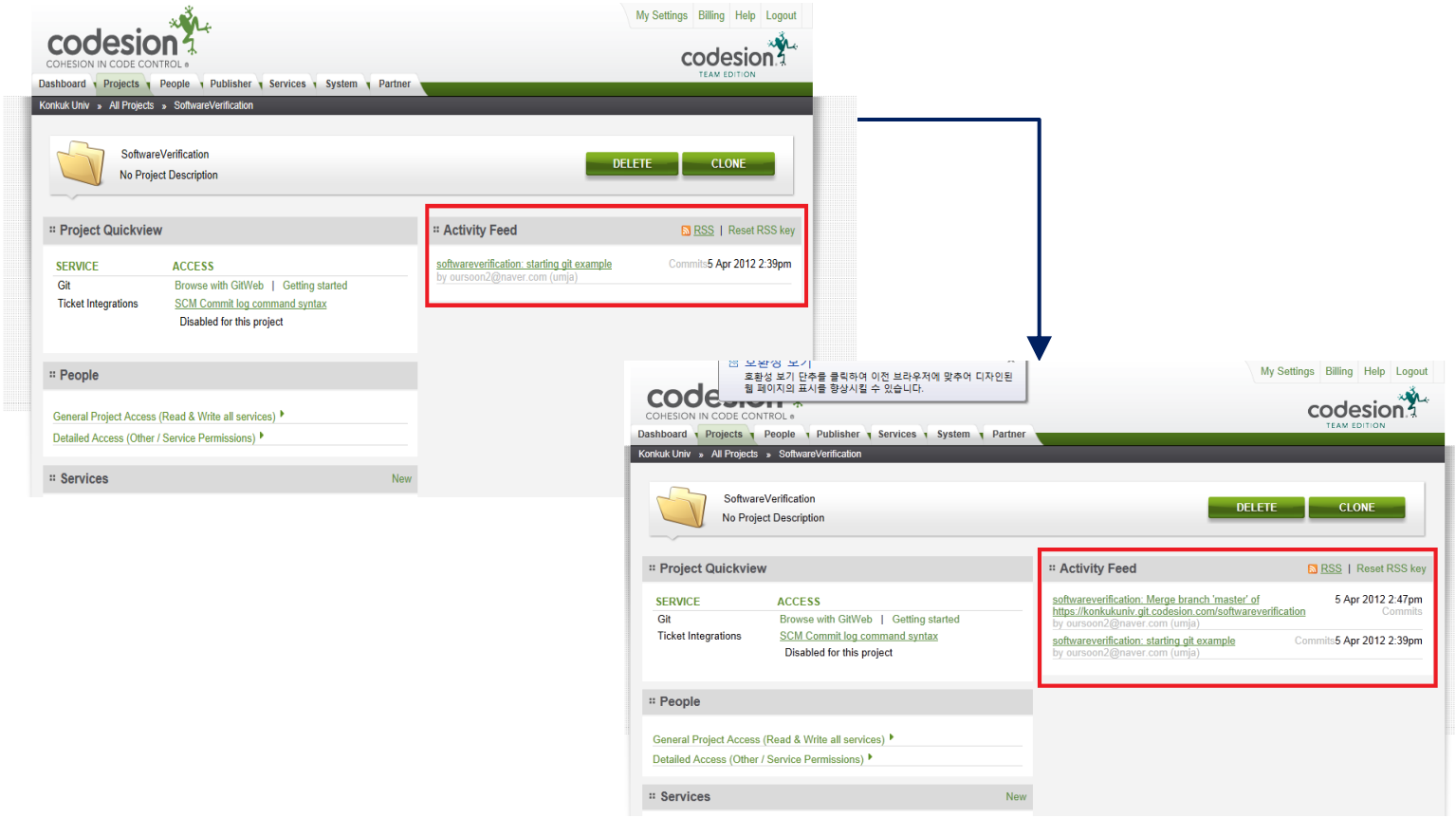
Tortoisegit

- Usage
  - Git push



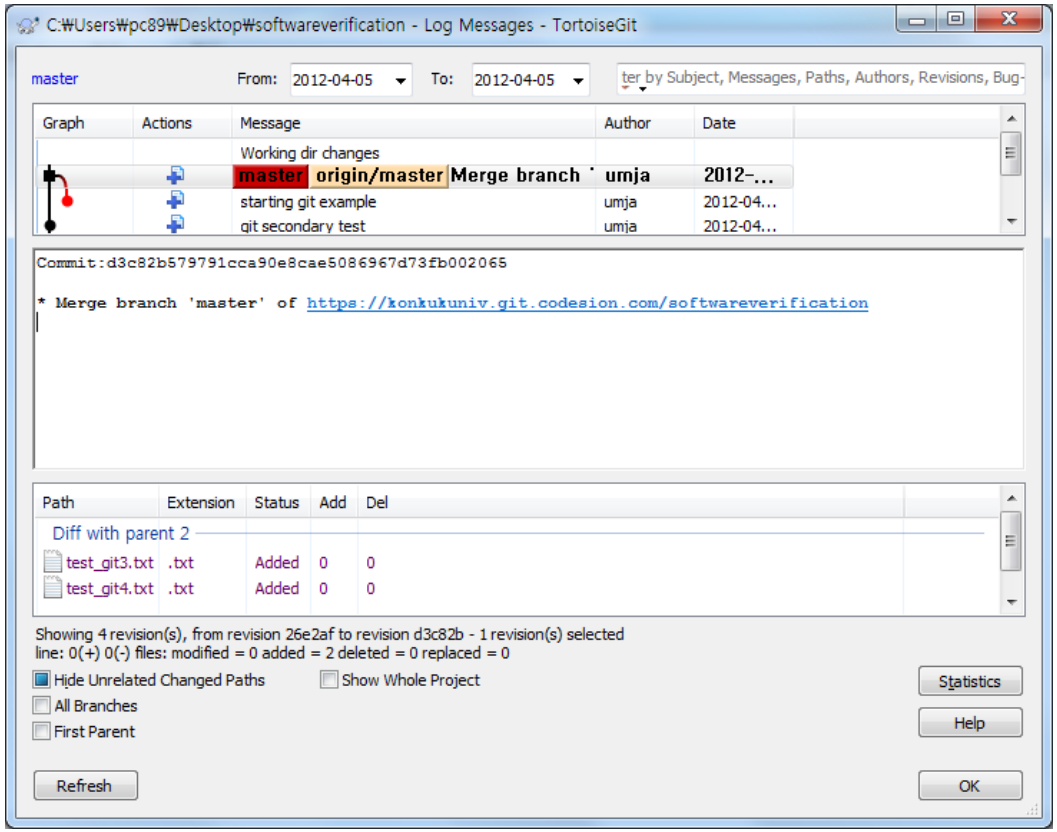
Tortoisegit

- Usage
  - Git push



Tortoisegit

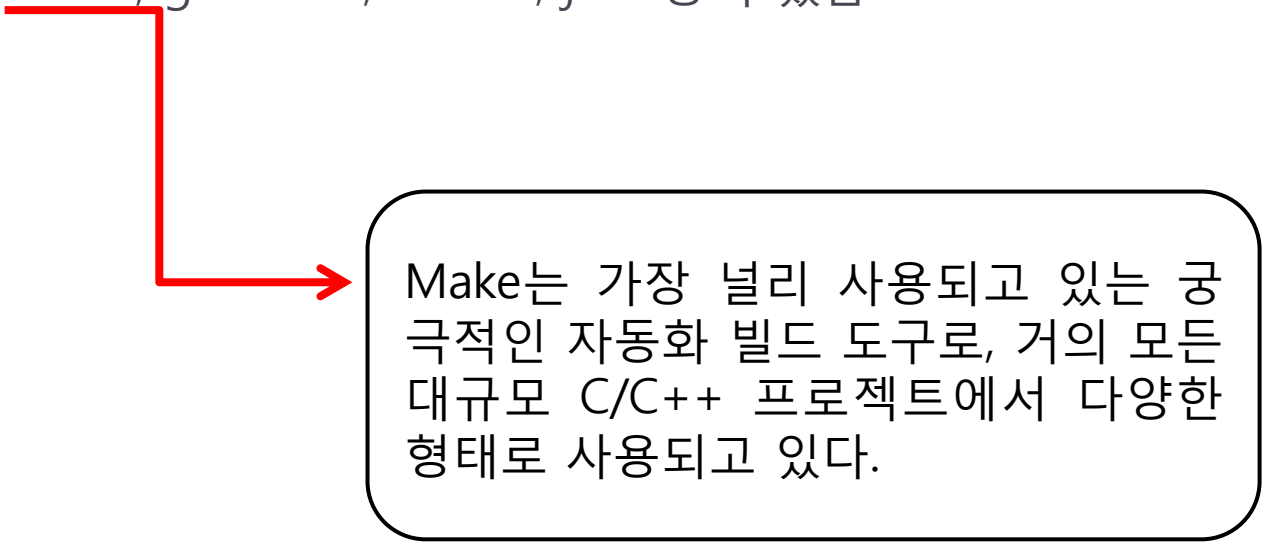
- Usage
  - Git show log



Build tool

- Build tool?
  - 일반적으로 빌드는 컴파일을 의미

    ex) 소스를 작성하고 빌드한다 = 소스를 작성하고 컴파일한다
  - 즉, 빌드 도구는 컴파일을 도와주는 유틸리티를 의미
  - 빌드 도구로는 make, gnumake, nmake, jam 등이 있음



Ant

- Ant
  - Another Neat Tool
  - Java 기반의 빌드 도구로써 멀티 플랫폼과 사용의 용이성, 확장성을 고려하여 설계

- Ant가 make보다 좋은이유?
  - Make 종류의 빌드 도구는 쉘 기반

    -> 작업공간이 Unix같은 OS 에 국한됨
  - Ant는 Java와 XML기반

    -> 멀티 플랫폼에서의 소프트웨어 개발환경 지원 가능

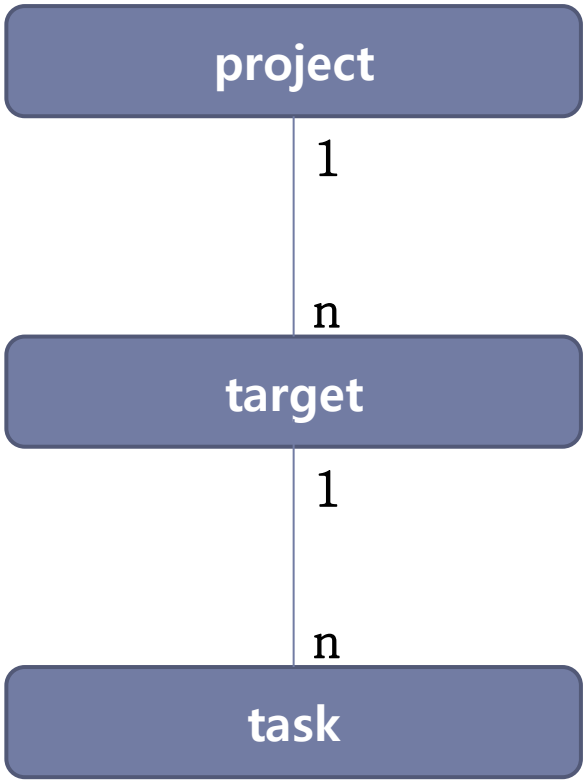
    -> 복잡할 쉘 명령어 대신에 XML 기반의 설정 파일을 사용하기 때문에 쉬움

    -> 미리 정의된 태스크를 사용하여 매우 쉽고 빠르게 배치 작업을 설정 가능

    -> 새로운 태스크의 추가를 통해서 처리 가능한 배치 작업의 확장이 가능

Architecture of Build.xml

- 빌드 파일은 project, target, task들의 집합체



About Project

- 빌드 파일에는 반드시 하나의 프로젝트가 있음
- 프로젝트는 Target의 집합체

속성	설명	타일	필수
Name	프로젝트 이름	String	no
Default	지정한 타겟이 없을 때 자동으로 실행할 디폴트 타겟 이름	String	yes
Basedir	빌드 파일 내에서 경로 지정의 기본이 되는 디렉토리	Path	no

About Target

- 다른 타겟과 의존성을 가질 수 있음

속성	설명	타입	필수
Name	타겟 이름	String	yes
Depend	의존성 있는 타겟 목록을 콤마로 구분	String	no
If	타켓이 실행되기 위해 설정해야 할 프로퍼티 이름	Property	no
Unless	타겟이 실행되기 위해 설정되지 말아야 할 프로퍼티 이름	Property	no
Description	타겟 설명	String	yes

About Task

- Ant의 기능은 task로 정의

  -> 즉, task는 특정한 일을 수행하는 컴포넌트 or 코드 단위

- Task 의 구조

< task_name attribute1="value1" attribute2="vaule2" ... >

- Task의 종류

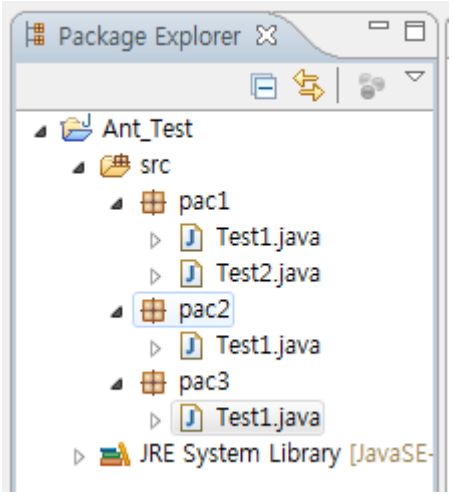
  -> 핵심 태스크 : 설치 후 즉시 사용 가능한 태스크
  -> 옵셔널 태스크 : 외부 라이브러리나 프로그램을 필요로 함
  -> 서드파티 태스크 : 프로젝트 외부에서 개발하여 ant와 통합 가능한 태스크
  -> 커스텀 태스크 : 사용자가 작성하고 컴파일 하여 사용

About Task

- Java - Ant의 기본 VM에서 자바 클래스를 실행시킴
- Javac - 자바 소스를 컴파일하는 태스크
- Jar - 클래스나 파일 및 디렉토리 들의 리소스를 하나의 jar파일로 묶어주는 태스크
- Javadoc - Javadoc 도구를 사용하여 javadoc 문서를 생성하는 태스크
- Mkdir - 디렉토리를 생성하는 태스크
- Copy - 파일이나 fileset에서 지정한 파일, 디렉토리 등을 복사하는 태스크
- War - jar 태스크의 확장형으로 웹 애플리케이션의 아카이브 파일인 war파일을 작성

Ant Example

Pac 1,2,3 을 compile 후 jar로 묶고 zip으로 압축



Build 디렉터리에서 pac3 패키지의 파일을 제외한 파일을 jar로 묶어서 dist에 pac1_2.날짜.jar 형태로 저장
(Dstamp -> yyyyMMdd)

Lib 디렉터리를 dist/lib에 복사하고 dist디렉터리의 파일들을 zip으로 묶음
Zip 파일 이름은 test.날짜.zip 형태가 됨

```
<target name="pac1_2" depends="compile"
description="pac1 and pac2 packaging">
<mkdir dir="${dist}"/>
<jar jarfile="${dist}/pac1_2.${DSTAMP}.jar">
<fileset dir="${build}">
<exclude name="pac3/*.*"/>
</fileset>
</jar>
</target>

...

<target name="doc" depends="pac1_2, pac2_3">
<mkdir dir="${doc}"/>
<javadoc destdir="${doc}">
<fileset dir="${src}"/>
</javadoc>
</target>

<target name="zip" depends="pac1_2, pac2_3">
<copy todir="${dist}/lib">
<fileset dir="${lib}"/>
</copy>
<zip destfile="test_${DSTAMP}.zip">
<fileset dir="${dist}"/>
</zip>
</target>
```

Summary about Ant

- Ant는 아주 작은 규모에서부터 매우 큰 대규모의 Java 프로젝트에 이르기까지 빌드, 테스트, 배치를 수행해 줄 수 있는 Java 기반의 도구
- Ant는 무엇을 빌드 할 것인지를 기술하는 XML 형식의 빌드 파일들을 사용
- 각각의 빌드 파일은 하나의 Ant 프로젝트를 다루게 되며 하나의 프로젝트는 여러 타겟들로 나뉘어질 수 있음
- 또한 각 타겟들은 하나 이상의 태스크들을 포함 가능
- 이러한 태스크들은 실제로는 구축작업을 수행하는 Java의 클래스들
- 타겟은 다른 타겟에 종속적일 수 있음
- Ant는 타겟을 수행할 때 이러한 타겟들 간의 종속관계를 파악하여 실행 시 그 순서에 따라 처리파일 단위의 종속성이나 규칙을 따르는 Make와는 달리, 파일 단위의 종속성에 대한 것은 태스크 내에서 처리하도록 함

Maven

- 애플리케이션을 개발하기 위해 반복적으로 진행해 왔던 작업들을 지원하기 위해 등장한 툴

- Benefit of Maven
  - 다양한 기능 지원 : Build, Documentation, Reporting, Dependency, SCM, Release, Distribution
  - 모든 프로젝트의 빌드 프로세스를 일관되게 가져갈 수 있음
  - Maven이 제공하는 많은 플러그인의 활용이 가능
  - Maven 프로젝트를 Eclipse 기반 프로젝트로 쉽게 변환이 가능
  - 신규 프로젝트 세팅을 쉽고 빠르게 진행할 수 있음

Maven

- Ant 와 비교
  - Ant 를 사용하여 Builds, Reporting 등의 작업을 진행하지만 일관된 가이드라인이 없는 상태이기 때문에 대부분의 작업을 반복해야 함

    ->그러나 Maven의 경우에는 프로젝트 관리를 위하여 필요한 모든 작업을 추상화하여 툴이 지원하도록 구현

  - Ant만큼 자유도가 높지는 않지만 Ant를 사용하면서 반복해야 했던 많은 작업들의 양을 줄여줌

    ->모든 프로젝트를 일관된 구조로 관리, 배포, 운영하는 것이 가능하기 때문에 프로젝트의 복잡도가 큰 프로젝트에 적합함

The Concept of RE

- Requirement Engineering
  - 요구 공학
  - 소프트웨어 개발에 필요한 제반 요구 사항들을 체계적으로 수립하기 위한 소프트웨어 공학의 한 분야

- Requirement Engineering Process
  - 시스템에 대한 요구사항 추출(Elicitation)
  - 추출된 요구사항 분석(Analysis) 및 검증(Validation)
  - 요구사항 명세화(Specification)
  - 저장 및 관리(Management)
  - 요구사항 변경에 대한 체계적인 대처 및 관리 방안

The Concept of RE

- Functional Requirements

  사용자가 필요로 하는 시스템의 기능 관련 요구 사항

  - 시스템의 개발 범위
  - 시스템의 목적
  - 시스템이 제공해야 하는 기능들
  - 시스템의 사용 방법

- Non-Functional Requirements

  시스템의 속성 및 제약 사항과 같이 시스템의 품질 속성에 대한 요구 사항으로 시스템의 설계, 인수 조건의 기준이 됨

  따라서 정량적이고 검증 가능하도록 기술되어야 함

  - 성능 : 응답 속도, 시간당 데이터 처리량 등
  - 신뢰성 : 데이터의 무결성, 정보 처리의 정확성 등
  - 보안성 : 시스템에 대한 비 권한자의 접근과 자료의 유출 방지 기능
  - 운영 편의성 : 장비, 유지 보수 방법 등 운영 및 유지 보수 환경

The Concept of RE

- 요구 사항 관리의 필요성
  - 요구 사항을 기반으로 소프트웨어의 구조와 기능을 정의

    -> 소프트웨어 요구 사항 명세서 : 고객과의 계약 내용을 포함. 설계, 구현, 시험 등 다음 개발 단계의 기준이 되는 문서. 요구 공학의 최종 결과물

  - 요구 사항 변경 시 필요한 여러 가지 재작업에 용이

    -> 뒤늦은 고객의 요청이나 불충분한 요구 사항 분석 등으로 인해 요구 사항은 지속적으로 변경 됨

    -> 관련된 다른 요구 사항 파악 필요

    -> 관련된 테스트 케이스 수정 파악 필요

  - 요구 사항은 고객과의 계약 사항 이행 여부 확인 할 수 있는 근거 자료

    -> SRS를 통해 요구 사항이 어떻게 소프트웨어 시스템에 구현되어 있는지 확인

    -> 시스템의 기능과 코드의 해당 모듈 매칭

    -> 테스트 케이스에 의해 성공적으로 시험되었다는 사실 추적 가능

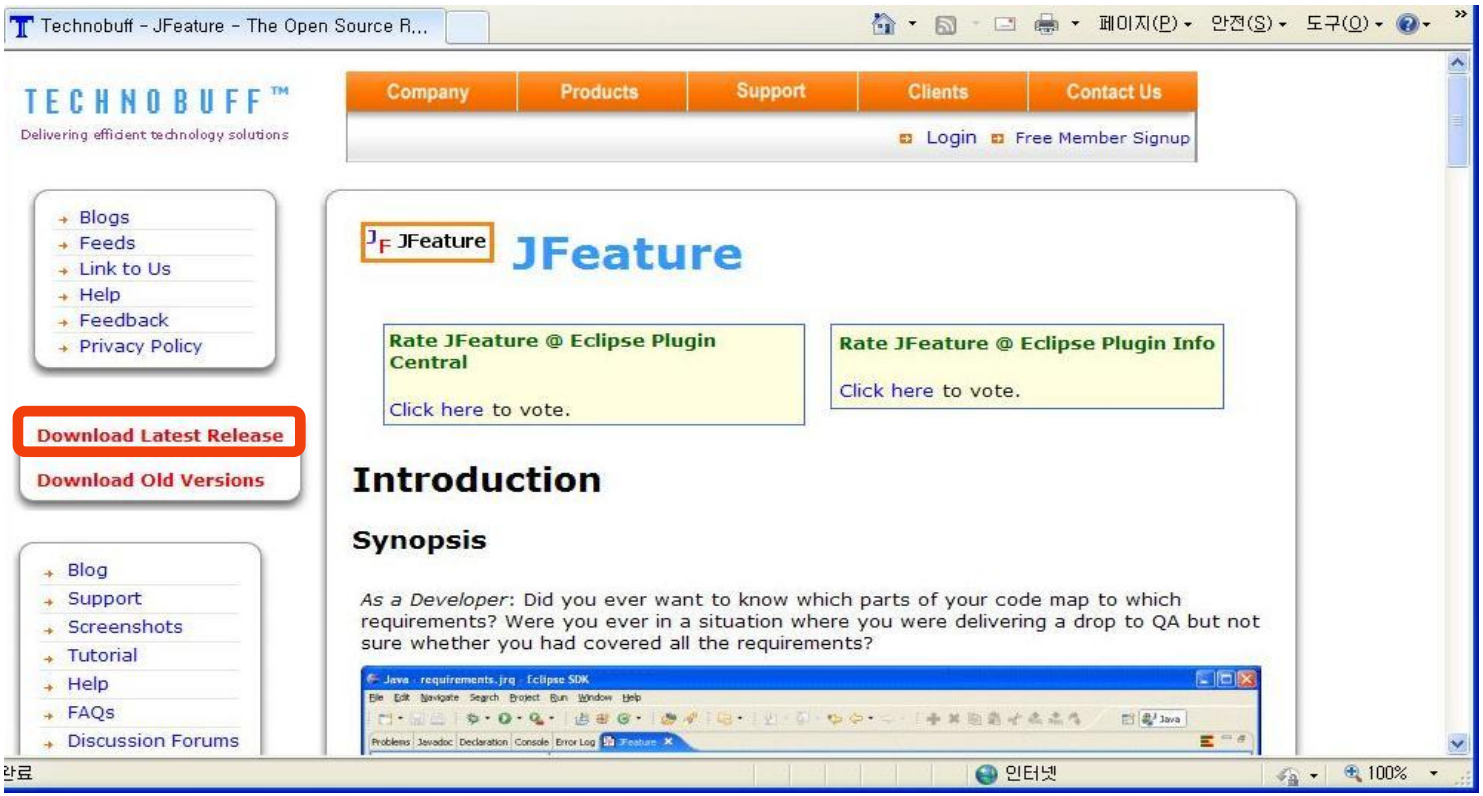
JFeature

- 요구 사항을 보고 Unit Test를 진행하여 좀 더 직관적인 Unit Test 가능
- 요구 사항을 기록하고 이들을 JUnitTestCase와 Mapping 시킴으로서 프로세스를 단순화
- 요구 사항의 수정이 일어났을 때, 즉시 View를 제공하므로 빠른 편집이 가능
- 요구 사항 반영 여부를 Coverage 형태로 쉽게 확인 가능
- 요구 사항을 직접 입력하거나, CSV나 XML 파일로 저장된 요구사항을 불러올 수 있음
- JUnitTestCase 외의 다른 Plug-in과는 연동되지 않음

JFeature

- Installation

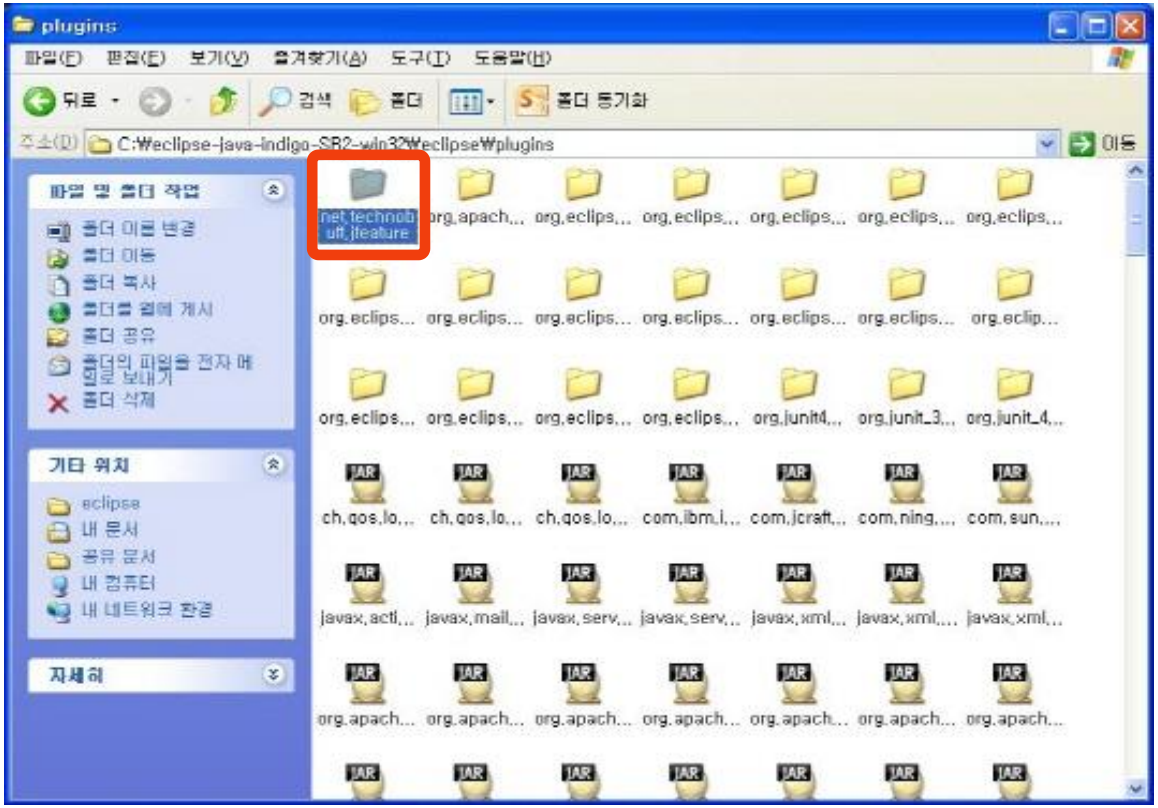
http://www.technobuff.net
에서 JFeature 다운



JFeature

- Installation

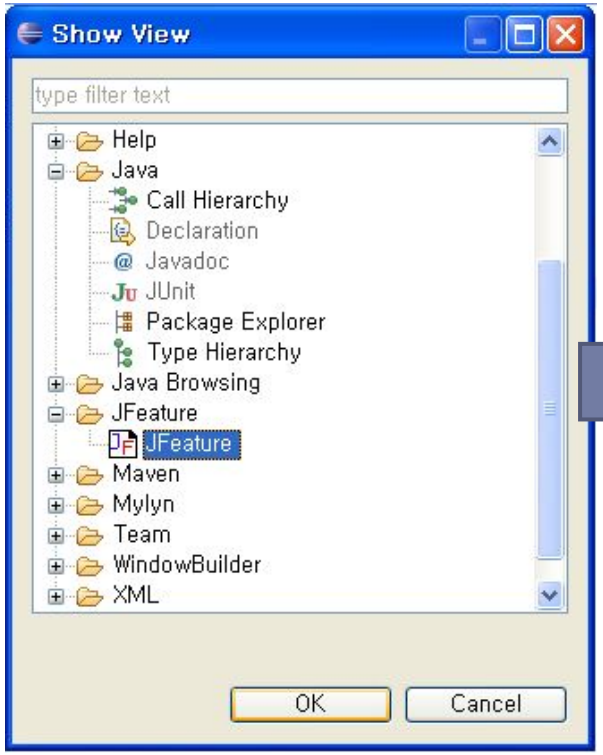
Eclipse가 설치된
Plugins 폴더에 압축 해제

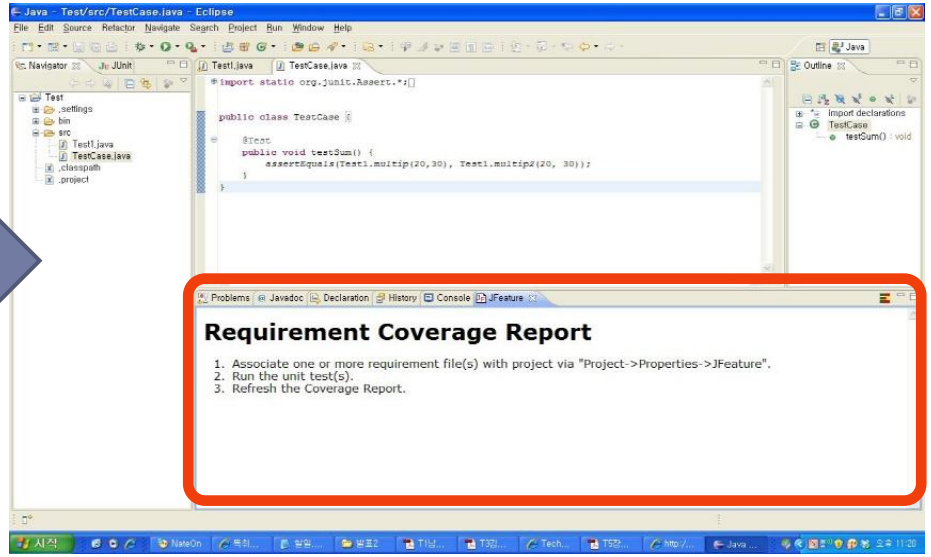


JFeature

- Installation

Windows -> Show View -> Other...에서
jFeature가 생성된 것 확인
Ok누르면 다음과 같은 jFeature View 생성

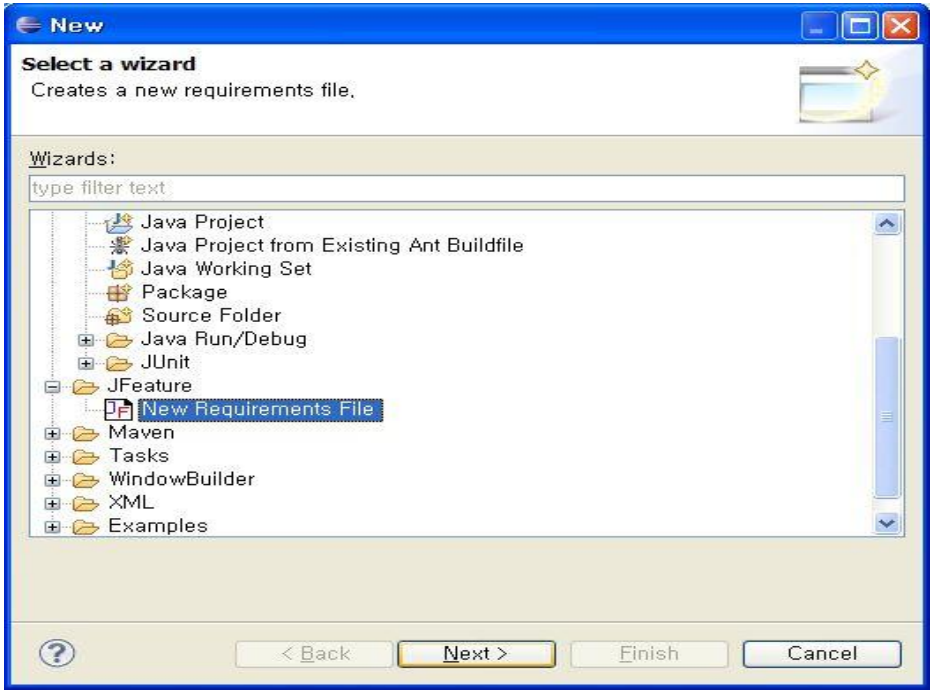


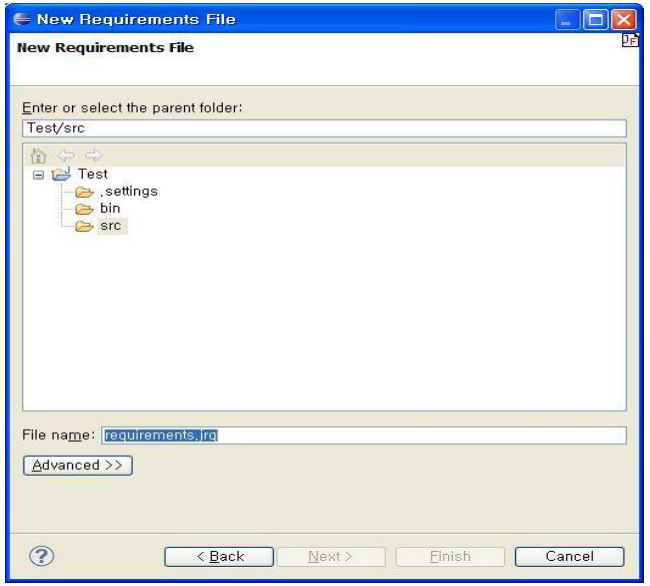


JFeature

- Usage

- 프로젝트 에서 New->Other -> Jfeatrue -> Jfeature Rquirements File을 통해 새 요구사항 파일을 생성
- 요구사항이 적용될 프로젝트를 선택하고 요구사항 파일 명을 입력

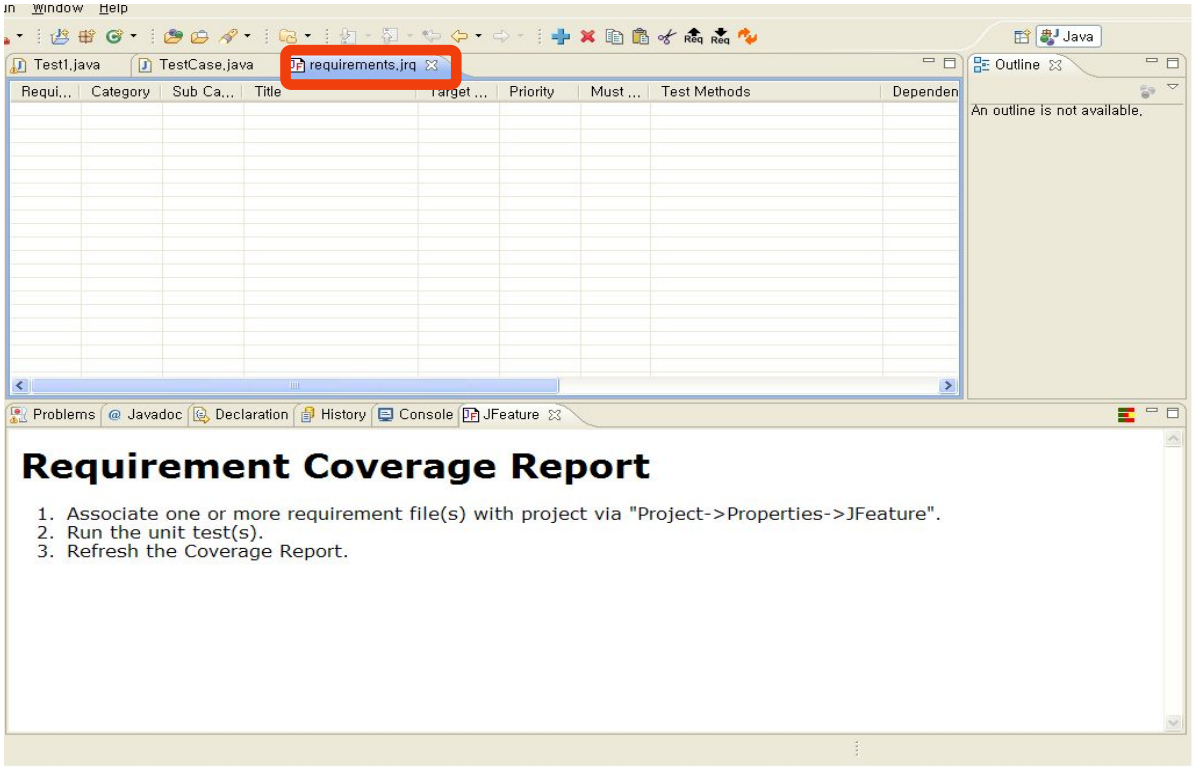




JFeature

- Usage

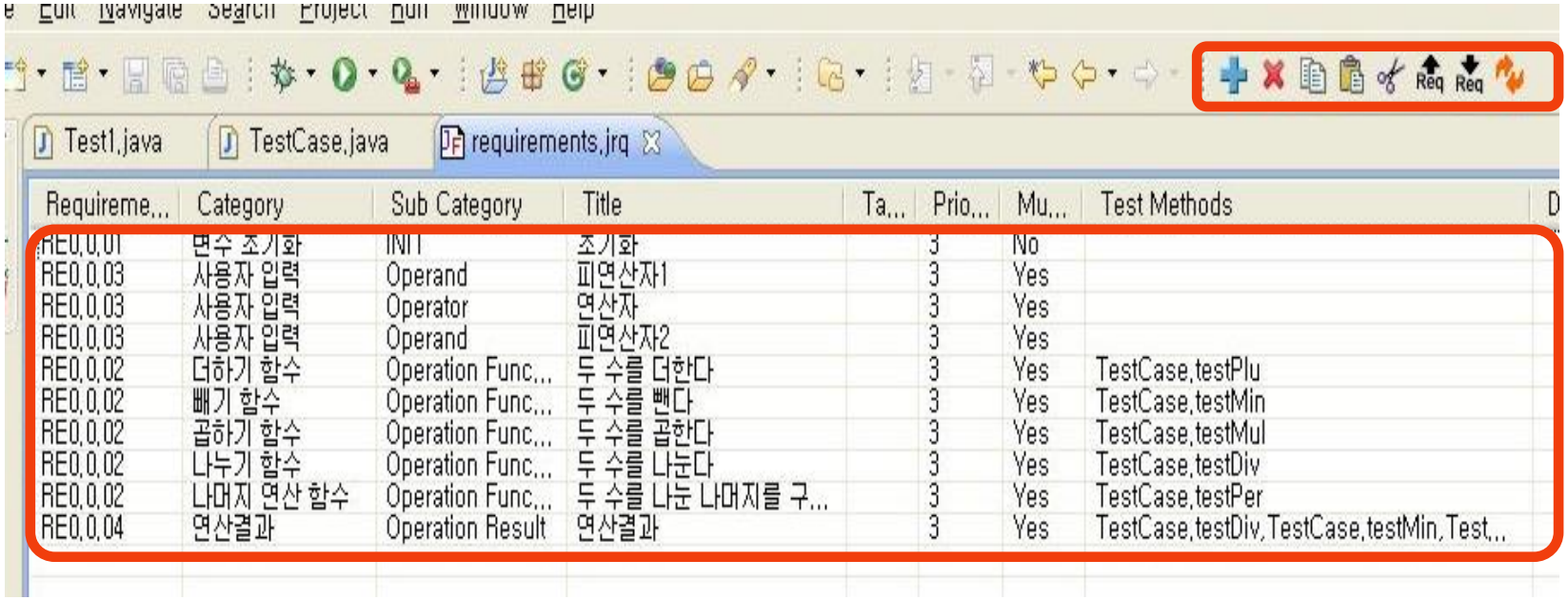
Requirments File 생성 확인



JFeature

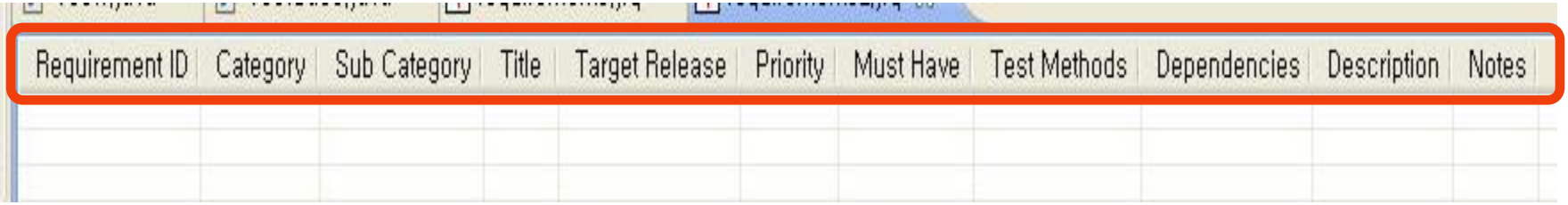
- Usage

- 상단에 추가된 메뉴를 이용해 요구 사항을 입력한다.
- Add Requirement, Delete Requirement, Copy Requirement, Paste Requirement, Cut Requirement, Move Requirement Up, Move Requirement Down, Round Trip Engineering의 기능



Requireme...	Category	Sub Category	Title	Ta...	Prio...	Mu...	Test Methods	D
RE0.0.01	변수 초기화	INIT	초기화		3	No		
RE0.0.03	사용자 입력	Operand	피연산자1		3	Yes		
RE0.0.03	사용자 입력	Operator	연산자		3	Yes		
RE0.0.03	사용자 입력	Operand	피연산자2		3	Yes		
RE0.0.02	더하기 함수	Operation Func...	두 수를 더한다		3	Yes	TestCase.testPlu	
RE0.0.02	빼기 함수	Operation Func...	두 수를 뺀다		3	Yes	TestCase.testMin	
RE0.0.02	곱하기 함수	Operation Func...	두 수를 곱한다		3	Yes	TestCase.testMul	
RE0.0.02	나누기 함수	Operation Func...	두 수를 나눈다		3	Yes	TestCase.testDiv	
RE0.0.02	나머지 연산 함수	Operation Func...	두 수를 나눈 나머지를 구...		3	Yes	TestCase.testPer	
RE0.0.04	연산결과	Operation Result	연산결과		3	Yes	TestCase.testDiv,TestCase.testMin,Test...	

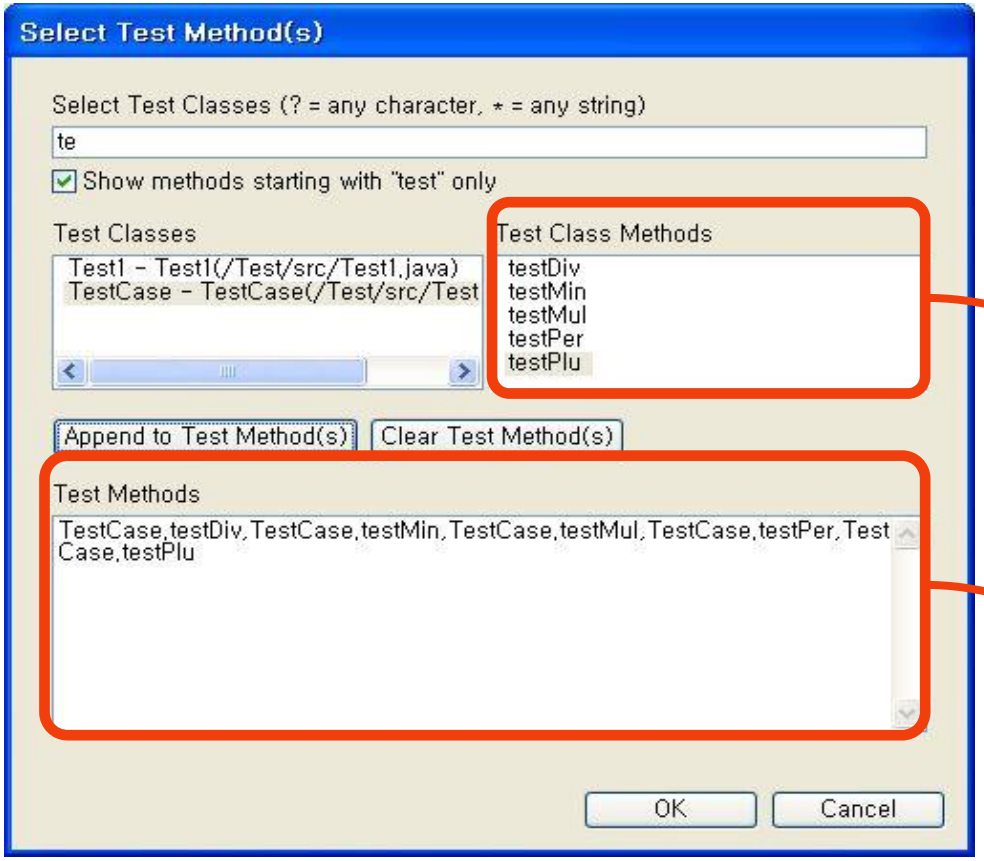
JFeature



- Requirement ID : 요구사항 고유의 식별번호
- Category : 요구사항의 상위분류
- Subcategory : 요구사항의 하위분류
- Title : 요구사항 명칭
- Target release : 요구사항의 타겟 배포 버전
- Priority : 요구사항의 구현 우선순위
- Must Have : 필수 요구사항인지 아닌지에 대한 판별 여부
- Test methods :요구사항과 매핑될 테스트 케이스
- Dependencies : 요구사항과 연관된 다른 요구사항(ID)
- Description : 요구사항의 상세한 설명
- Notes : 기타사항

JFeature

- 요구 사항과 TestCase 연결



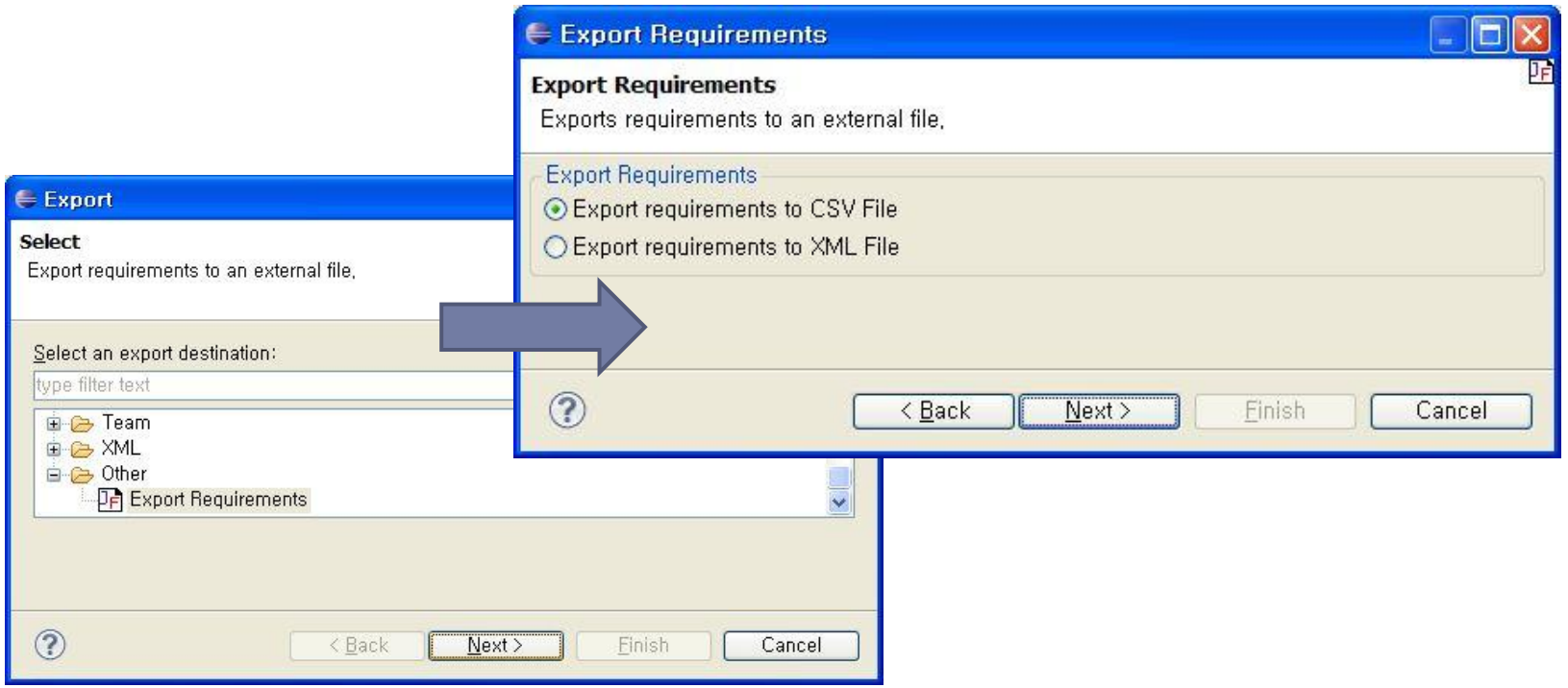
연결시킬
TestCase 목록

여러 TestCase
추가 가능

JFeature

- Export

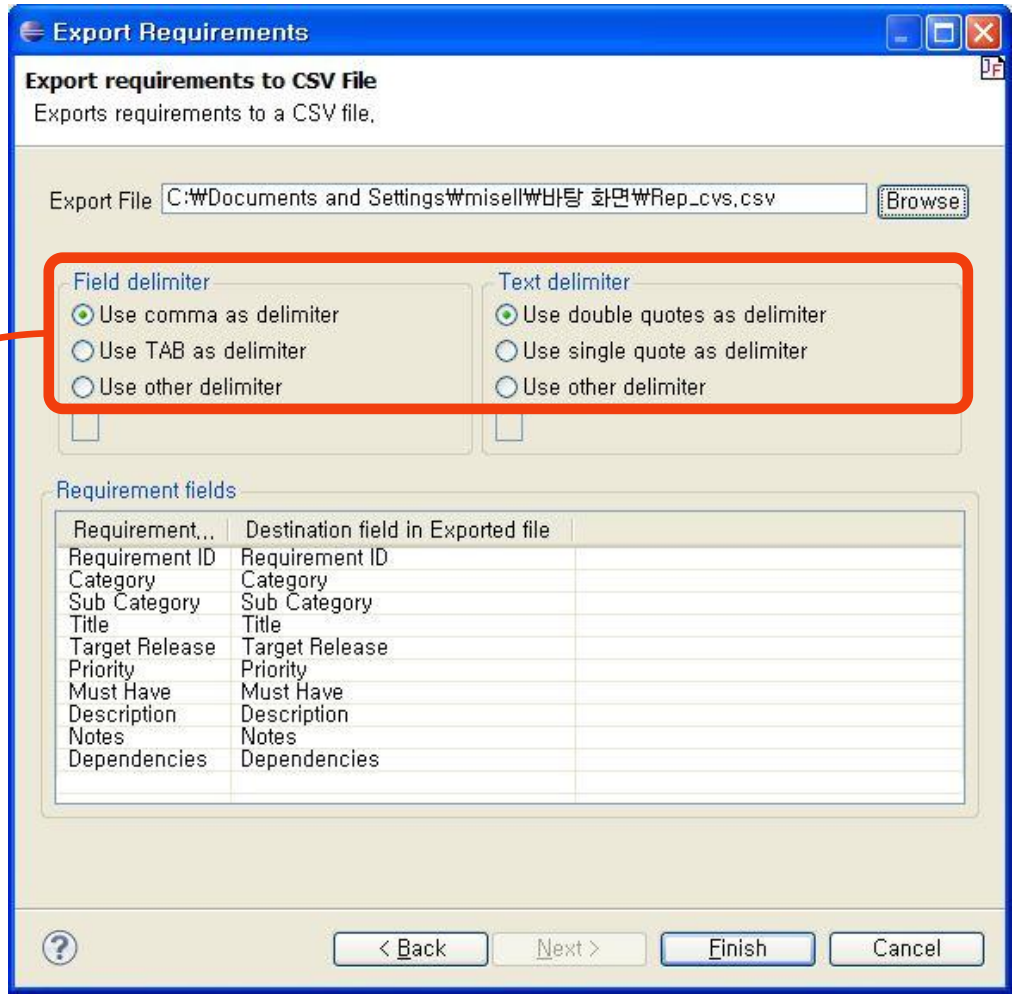
  Requirement 파일을 CSV, XML 파일로 저장
  File -> Export



JFeature

- Export

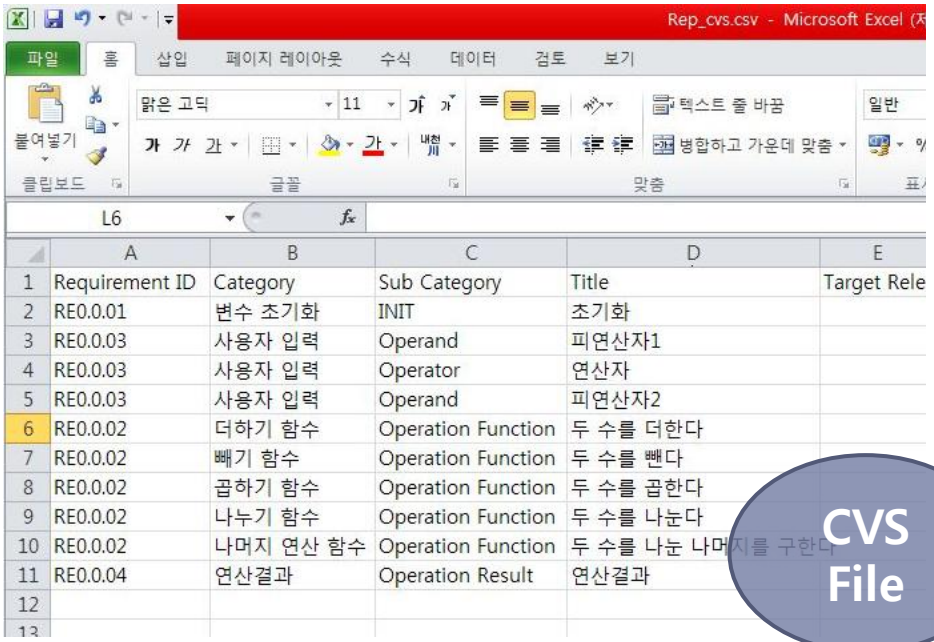
구분자 지정



JFeature

- Export

Encoding="UTF-8"이 기본
한글로 작성된 Requirement에는 부적합



	A	B	C	D	E
1	Requirement ID	Category	Sub Category	Title	Target Rele
2	RE0.0.01	변수 초기화	INIT	초기화	
3	RE0.0.03	사용자 입력	Operand	피연산자1	
4	RE0.0.03	사용자 입력	Operator	연산자	
5	RE0.0.03	사용자 입력	Operand	피연산자2	
6	RE0.0.02	더하기 함수	Operation Function	두 수를 더한다	
7	RE0.0.02	빼기 함수	Operation Function	두 수를 뺀다	
8	RE0.0.02	곱하기 함수	Operation Function	두 수를 곱한다	
9	RE0.0.02	나누기 함수	Operation Function	두 수를 나눈다	
10	RE0.0.02	나머지 연산 함수	Operation Function	두 수를 나눈 나머지를 구한다	
11	RE0.0.04	연산결과	Operation Result	연산결과	
12					

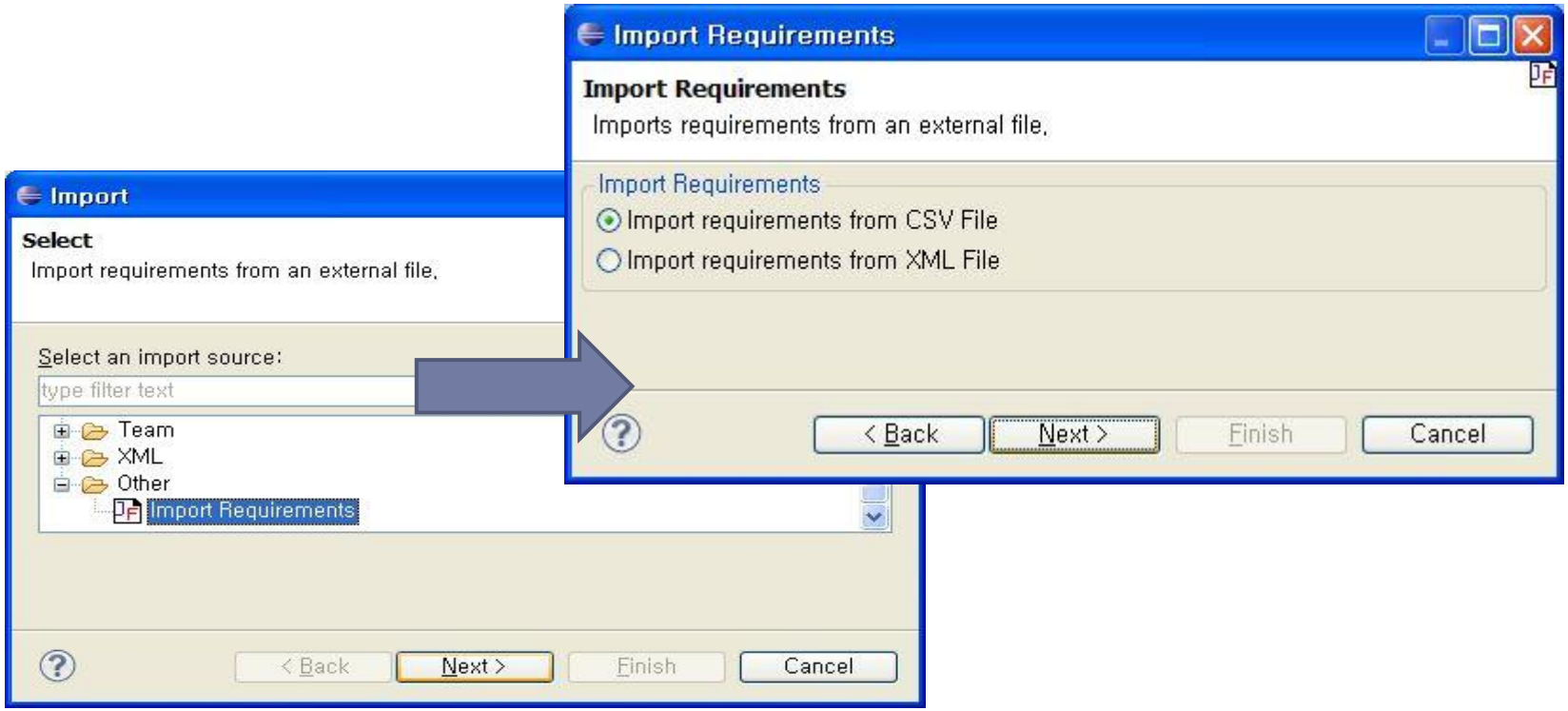
CVS File

```
<?xml version="1.0" encoding="euc-kr" ?>
- <requirements>
- <requirement>
  - <data>
      <![CDATA[ Requirement ID ]]>
    </data>
  - <data>
      <![CDATA[ Category ]]>
    </data>
  - <data>
      <![CDATA[ Sub Category ]]>
    </data>
  - <data>
      <![CDATA[ Title ]]>
    </data>
  - <data>
      <![CDATA[ Target Release ]]>
    </data>
  - <data>
      <![CDATA[ Priority ]]>
    </data>
  - <data>
      <![CDATA[ Must Have ]]>
    </data>
  - <data>
      <![CDATA[ Description ]]>
    </data>
  - <data>
      <![CDATA[ Notes ]]>
    </data>
  - <data>
      <![CDATA[ Dependencies ]]>
```

XML File

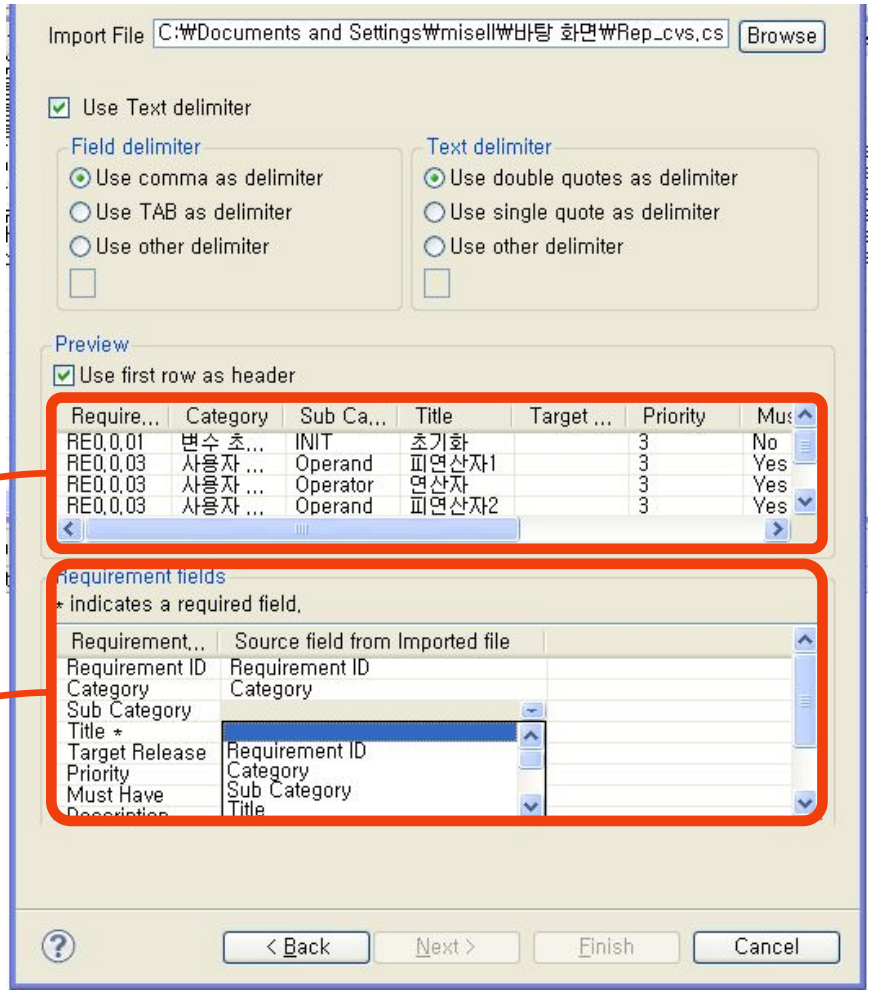
JFeature

- Import
  - CSV, XML 파일에서 Requirement 파일 생성
  - File -> Import



JFeature

- Import

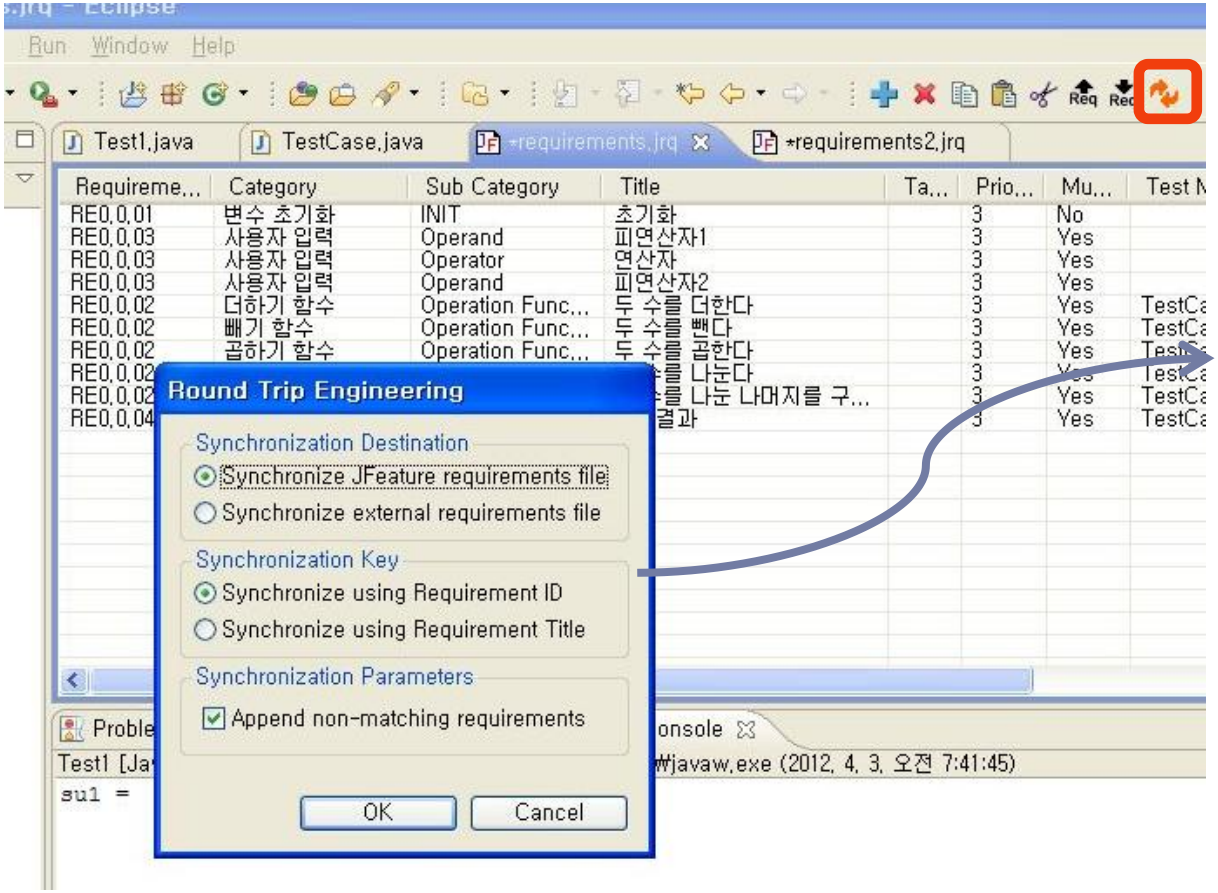


미리보기

JFeature 파일의 필드와 CSV 파일의 필드 값의 매치

JFeature

- Round Trip Engineering



-Synchronize JFeature requirements file : JFeature 요구 사항 파일의 변경사항을 CSV, XML 파일에 변경

-Synchronize external requirements file : CSV, XML 파일을 JFeature 요구사항 파일에 적용

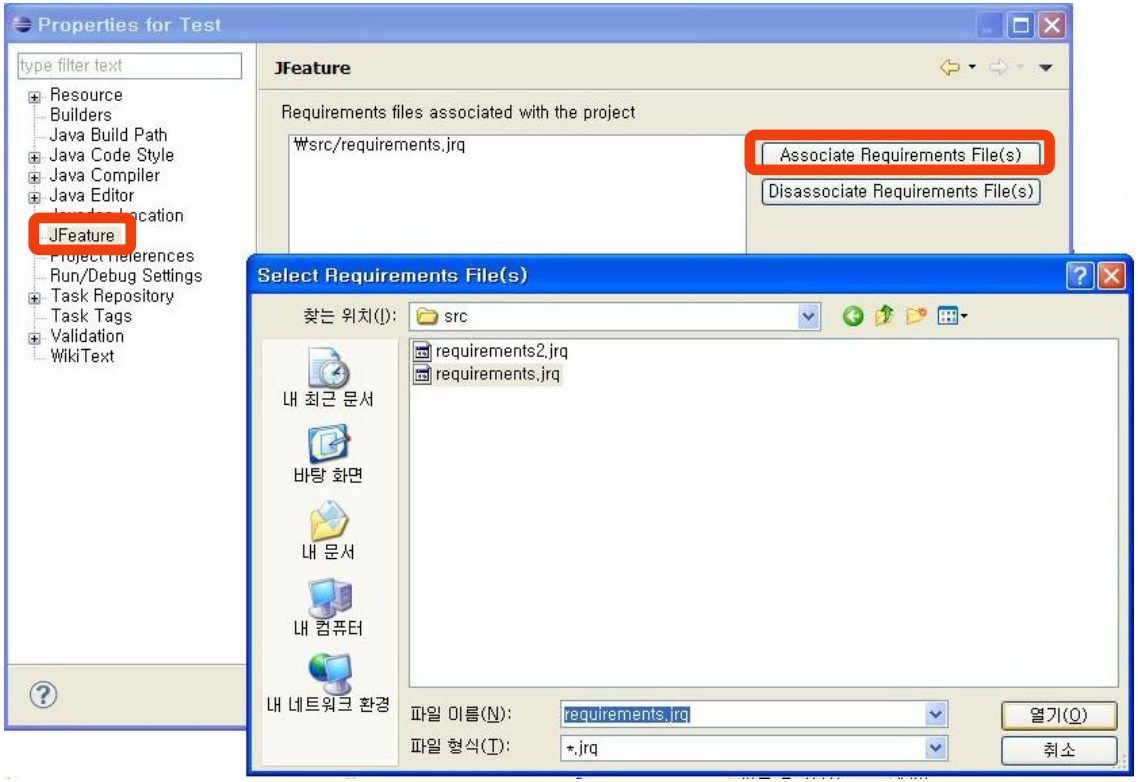
-Synchronize using Requirement ID : 요구사항 ID를 비교 한 후 적용

-Synchronize using Requirement Title : 요구사항 제목을 비교 한 후 적용

- Append non-matching requirements : 일치하지 않는 요구사항도 추가

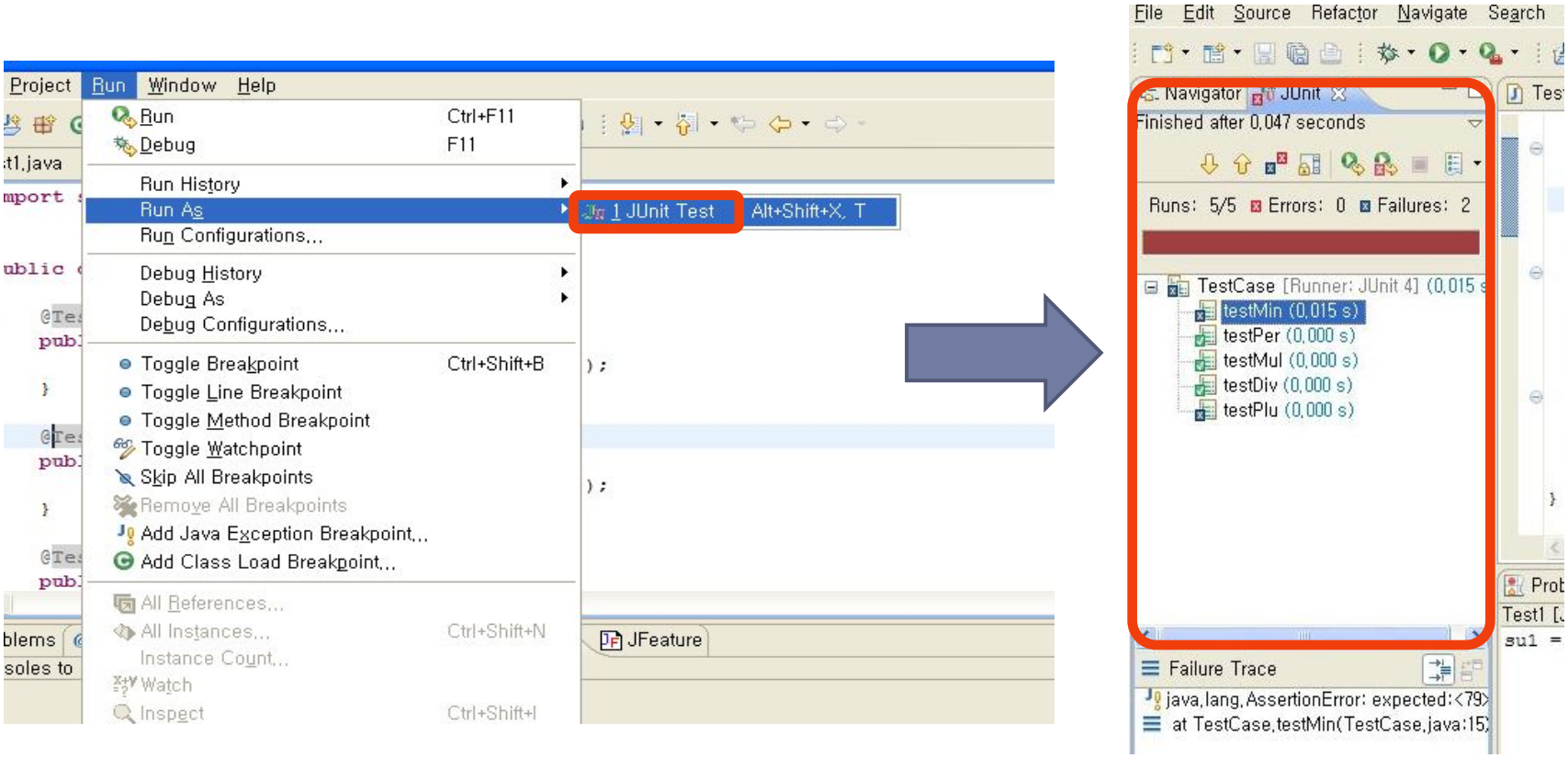
JFeature

- Report와 프로젝트 연결
  - Project -> Properties -> Jfeature -> Associate Requirements File -> 요구사항 파일 선택



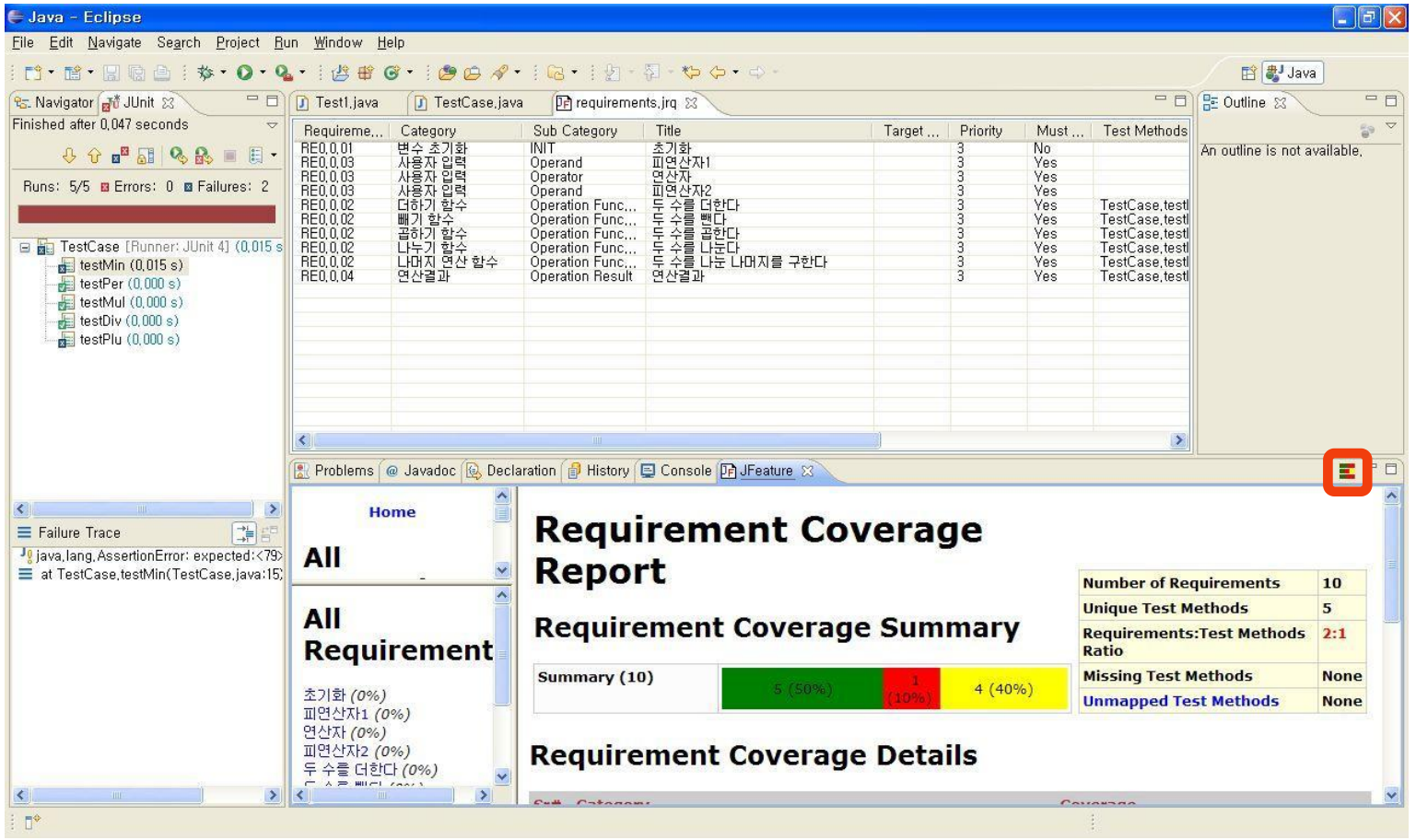
JFeature

- JUnit Test 수행



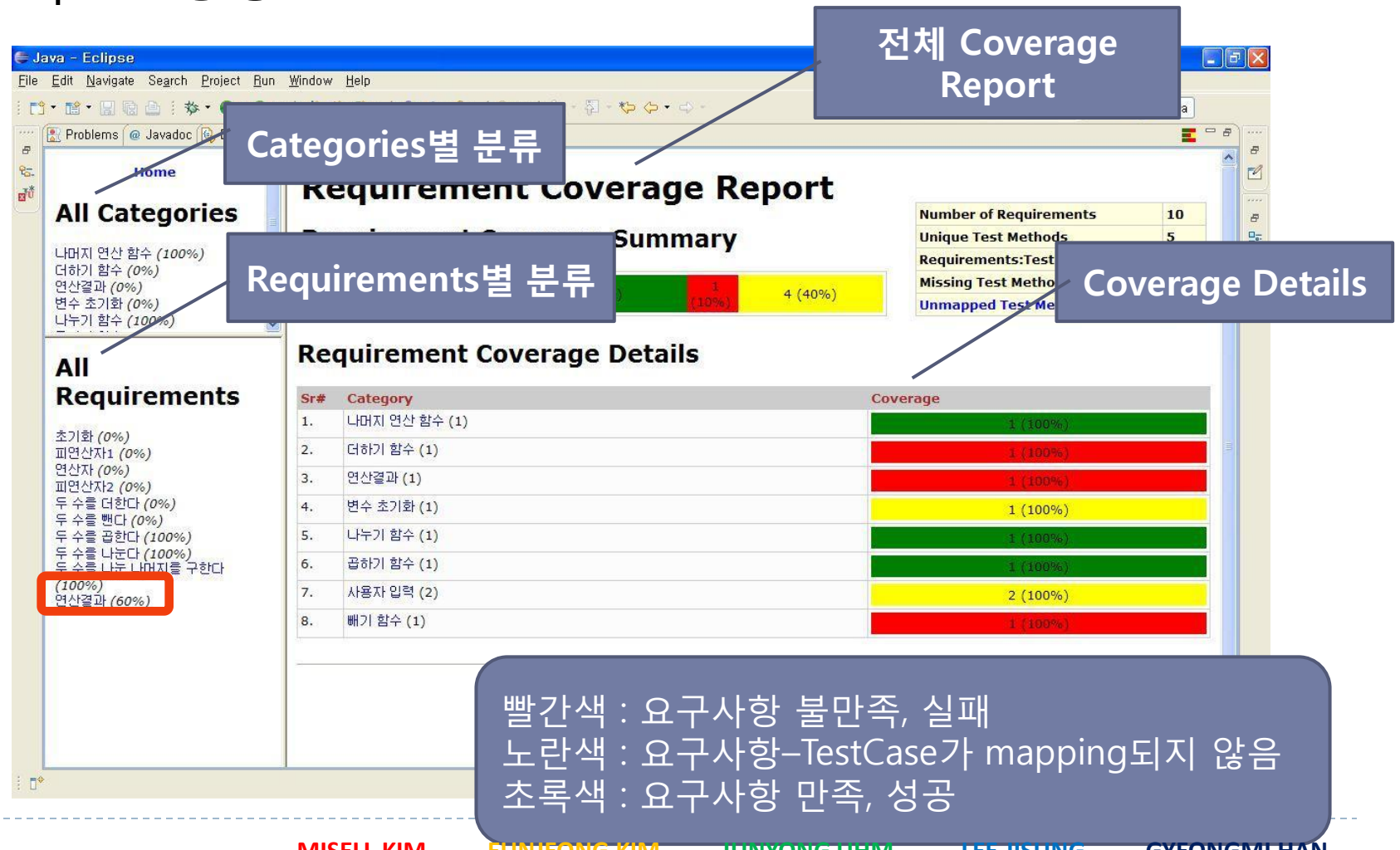
JFeature

- Report 생성



JFeature

- Report 생성



OSRMT

- Open Source Requirements Management Tool
- 특징, 요구사항, 설계, 구현, 테스팅 등의 SDLC에 대한 요구사항을 관리하고 추적할 수 있도록 디자인
- Client/Server System
  다수의 개발자간, 혹은 개발자와 의뢰자의 빠르고 정확한 의견 교환 가능
  WEB을 통해 데이터베이스의 내용에 접근 가능
- 독립된 Platform, 간편하면서 강력한 Requirements
- 다수의 개발자가 개발할 때 편리
- 요구사항에 대한 내용을 HTML, PDF 등 다양한 형태의 파일로 출력하여 확인 가능

IBM Rational DOORS

- (Dynamic Object Oriented Requirements System)
- 광범위하고 협력적인 요구 사항 관리 환경을 제공
- 내장된 변경 제안 시스템(Change Proposal System)을 통해서 또는 IBM Rational Change나 IBM Rational ClearQuest를 통해서 요구 사항에 대한 변경 사항을 관리
- 공급업체와 개발 파트너들이 개발 프로세스에 직접적으로 참여할 수 있게 해주는 요구 사항 교환 형식을 지원
- 간단하면서도 강력한 추적성을 확보하기 위해서 요구 사항을 설계 항목, 테스트 계획, 테스트 사례 및 기타 요구 사항 사례와 기타 다른 요구 사항에 링크
- 테스터가 요구 사항을 테스트 사례에 링크할 수 있도록 수동 테스트 환경을 위한 테스트 추적 툴킷(Test Tracking Toolkit)이 포함

IBM Rational RequisitePro

- 요구사항의 포착을 용이하게 하고 데이터베이스를 이용하여 요구사항 문서를 데이터베이스와 연결하여 요구사항에 우선순위를 부여하고 정리하기 쉽게 최적화된 도구
- 사용하기 쉬운 요구사항 관리 솔루션
- Microsoft Word 문서의 친숙함과 사용 용이성을 강력한 데이터베이스 기능과 결합해 보다 효과적으로 요구 사항을 관리
- 변화에 따른 영향을 철저하게 파악할 수 있기 때문에 변화를 보다 효과적으로 관리
- 툴과 팀 전반의 요구사항을 통합해 최신 요구사항 정보를 모든 사람에게 알려줄 수 있다.