

전자정부 표준프레임워크 개발환경



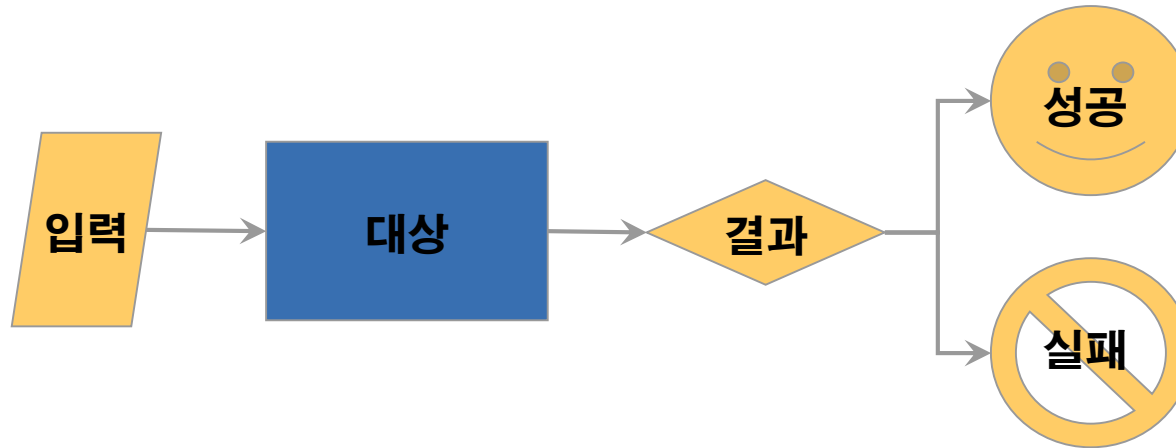


1.테스트 구조

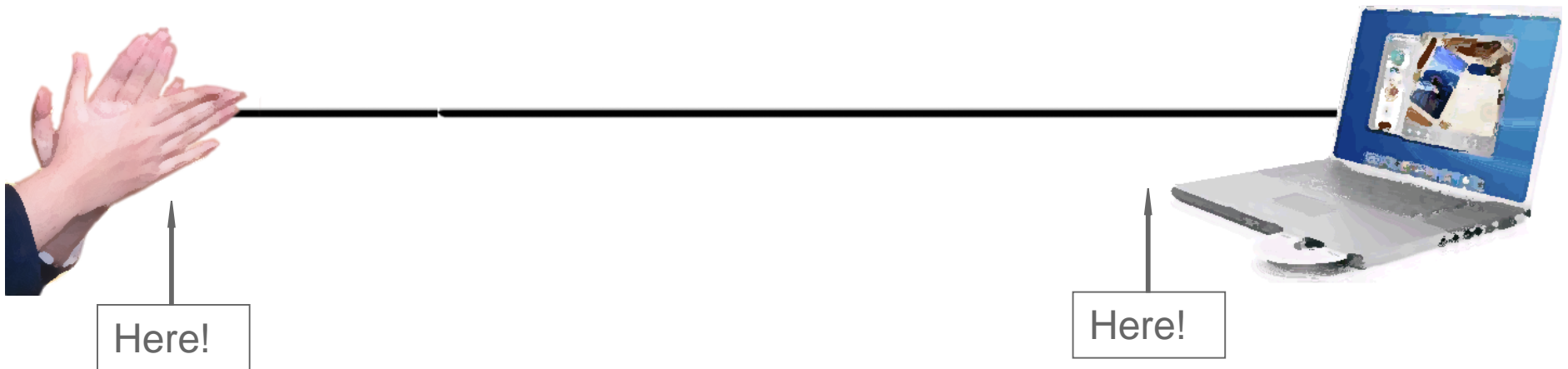
1. 테스트
2. 개요
3. 테스트 종류
4. 오픈소스
5. Unit Test
6. Mock
7. Batch Job Test
8. DB Test
9. Test Automation
10. Test Reporting
11. Test Coverage

□ 테스트(Test)

- 테스트 대상에 입력값을 넣었을 때 그 결과가 성공 혹은 실패의 결과를 내는 것이다.

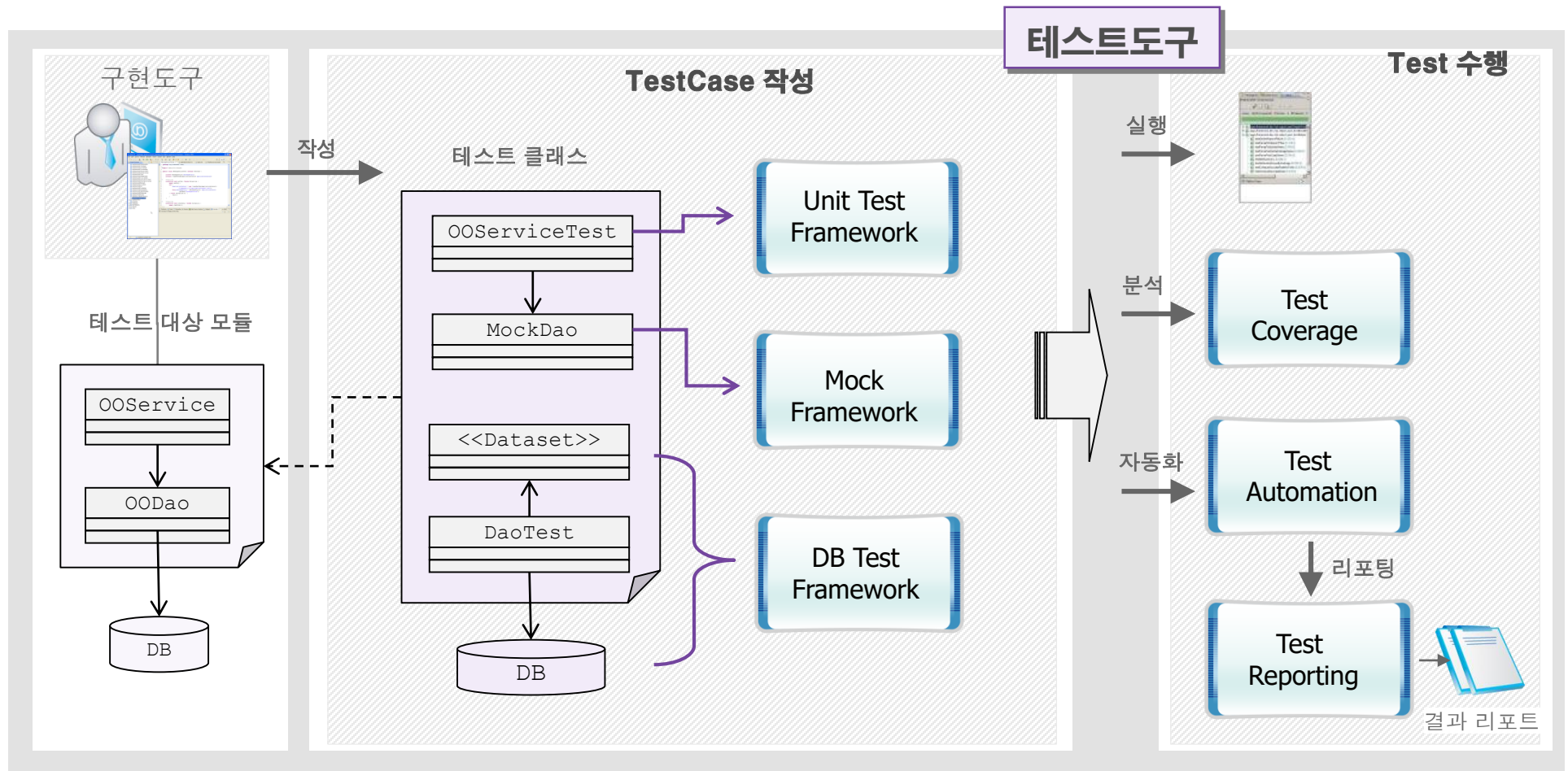


□ 수동 테스트 vs. 자동 테스트

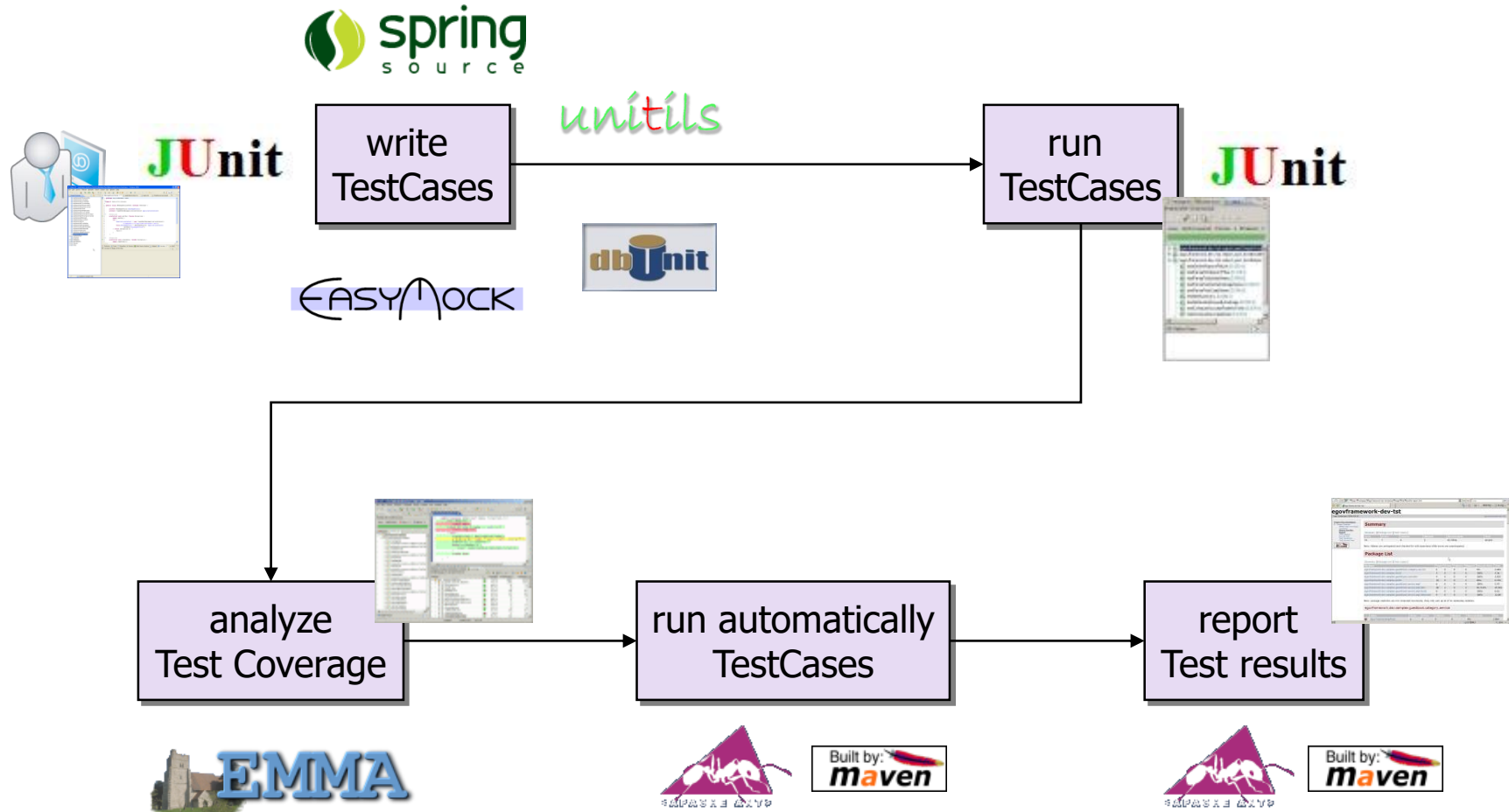


	수동 테스트	자동 테스트
장점	<div> <div>□ 쉽다. 간편하다.</div> <div>□ 테스트 불가능한 상황이 별로 없다.</div> </div>	<div> <div>□ 언제든지 같은 테스트를 여러 번 수행 가능</div> <div>□ 기존 테스트는 새 테스트를 작성하는 발판이 된다. (지식축적)</div> <div>□ 빠르므로 자주 돌려볼 수 있다.</div> <div>□ 개발자가 코드 개선 활동을 수행할 때에도 회귀테스트(Regression Test)를 자주 할 수 있다.</div> </div>
단점	<div> <div>□ 휘발성</div> <div>□ 테스트 항목이 늘어난다면?</div> <div>□ 다른 사람에게 테스트 내용을 설명하기 어려움</div> <div>□ 재현하기 어려움</div> </div>	<div> <div>□ 코드로 작성해야 한다. <div> -기술, 노하우가 필요하다</div> <div> -코드로 테스트를 작성할 수 없는 상황이 있다. (수동 테스트와 공조)</div> <div> -테스트도 관리 대상이다.</div> </div> </div> <div> <div>□ 모듈화가 잘 되어 있어야 한다. <div> -디자인의 개선 필요</div> <div> -Legacy Code의 어려움</div> </div> </div>

테스트 도구는 Unit Test, Mock, DB Test Framework을 통해 TestCase 작성을 지원하고, Test Automation, Test Coverage, Test Reporting 등의 기능을 제공함



□ 테스트 도구 사용 프로세스



Unit Test란 대상 코드에 대해 테스트하고자 개발자가 작성한 코드로서, 주로 특정 메소드를 실행해서 그 결과가 기대값과 일치하는지 확인하는 형태이며, **Unit Test**는 서로 독립적으로 수행되어야 함.

□ 효과

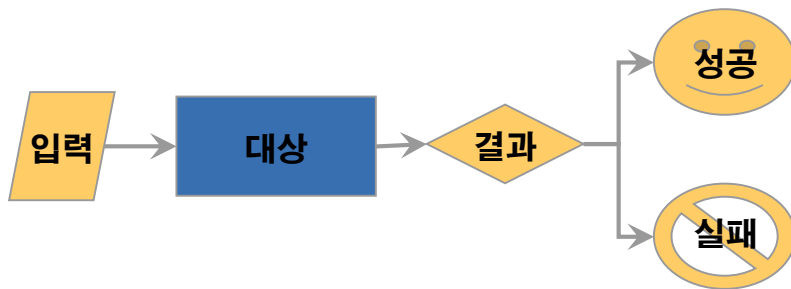
- 작성한 코드의 설계 개선 작업 시, 코드 품질에 대한 확신
- 코드 수정 시 버그를 쉽게 찾을 수 있게 해줌
- 자동화된 회귀 테스트 (Regression Test)를 가능하게 해주는 Source가 됨

□ 작성 범위

- 주요 흐름에 대한 테스트 (the happy path)
- 또 다른 주요 흐름에 대한 테스트 (the main alternative path)
- 경계 조건에 대한 테스트 (null 인자 체크 등)
- Exception 테스트 (Exception 발생하는 조건에 대한 테스트)

□ 구성

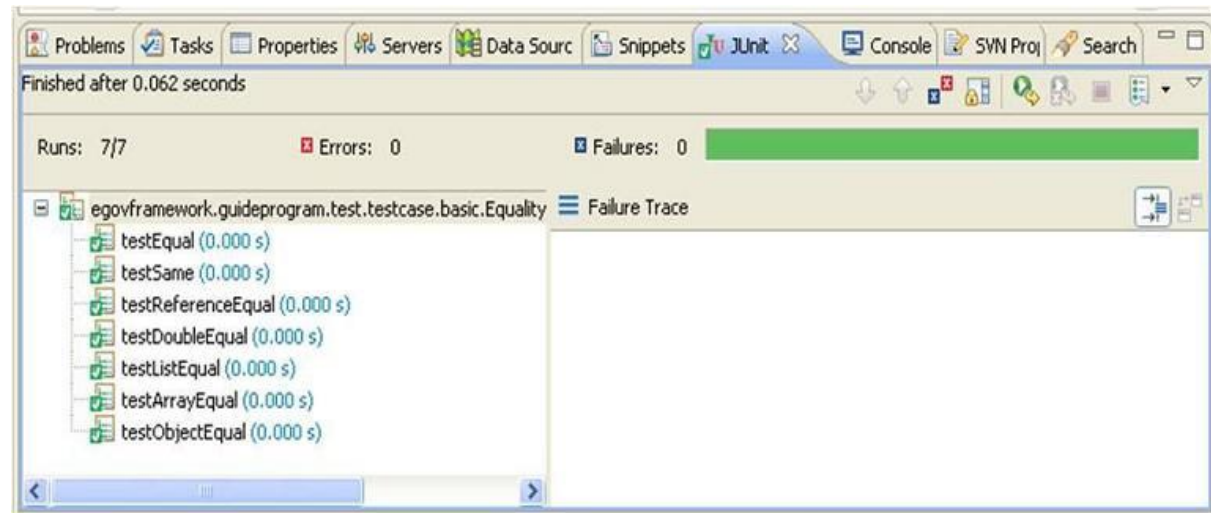
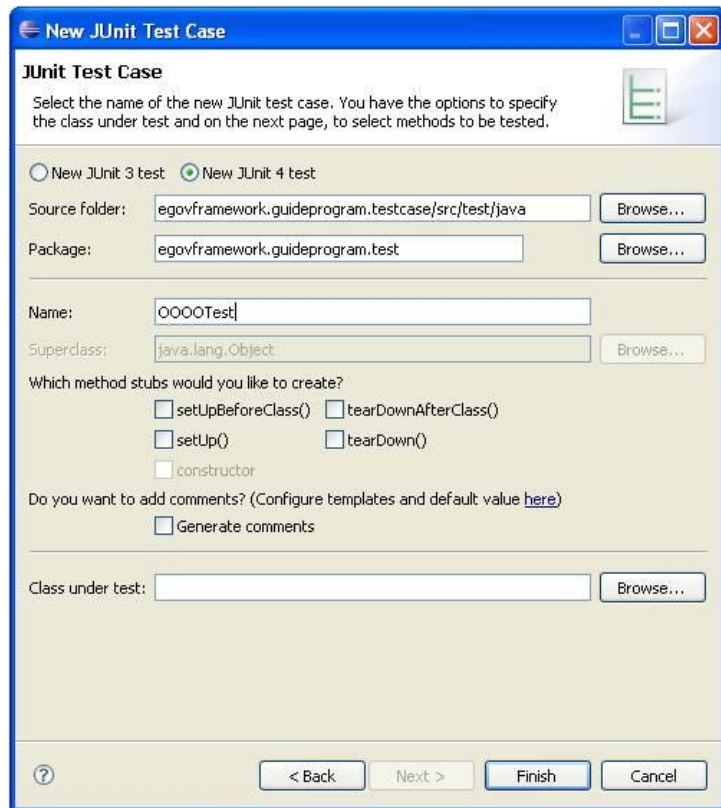
- 테스트 프레임워크를 사용하는 Class
- 공용으로 사용하는 테스트 데이터 (test fixture)
- 테스트 데이터 준비 (Setup of test data)
- 테스트 메소드 (testXXX)
 - (테스트 별 준비)
 - 테스트 대상 메소드 실행
 - assert 문을 이용한 결과 확인 (assertTrue, assertEquals etc.)
- (내부 메소드)



```

1 public class UserAdminTest {
2
3     /* Class under test */
4     private UserAdmin userAdmin;
5
6     /* A simple test user */
7     private User user; // Test Data (Fixture)
8     /* An administrator role */ //
9     private Role adminRole; //
10
11     /**
12      * Initializes the test fixture.
13      */
14     @Before
15     public void setUp() throws Exception {
16         userAdmin = new UserAdmin();
17         user = new User("John", "Doe"); // Test Data Setup
18         adminRole = new Role("Administrator"); //
19     }
20
21     /**
22      * Test for method with ...
23      */
24     @Test
25     public void testAddUser() {
26         user.setAge(18); // Extra Test Setup
27         userAdmin.addUser("jdoe", user, adminRole); // Use
28         Test Data
29
30         User result = userAdmin.getUser("jdoe");
31         assertEquals("John", result.getFirstName());
32         assertEquals("Doe", result.getLastName());
33     }
34 }
    
```


JUnit은 자바 프로그래밍 언어를 위한 Unit Test Framework로, Unit Test 코드를 작성하고 자동화된 테스트를 수행할 수 있는 기능을 제공함



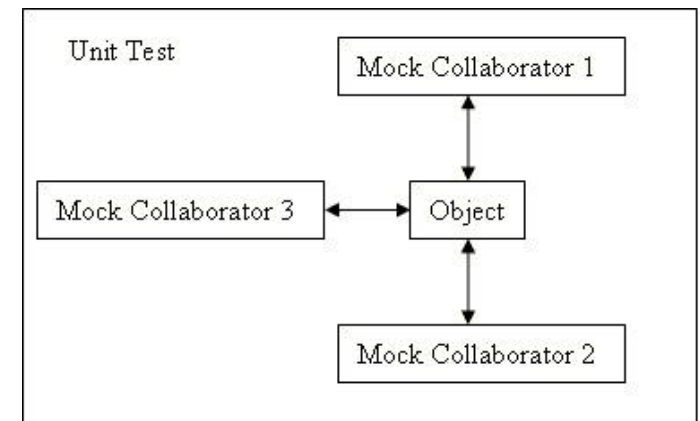
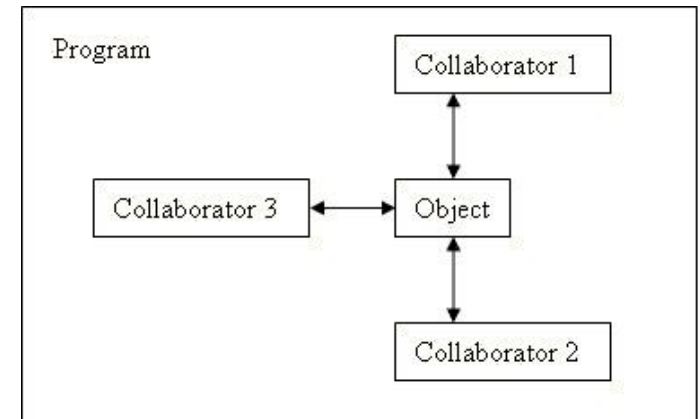
Mock 객체는 **Unit Test**의 독립성(isolation)을 높여주기 위해 사용하며, 테스트하고자 코드와 관련이 있는 객체(collaborator)를 흉내내어 **Unit Test**를 수행할 수 있도록 도와주는 객체임

□ Mock 객체를 사용하는 경우

- 진짜 객체를 준비, 설정하기 어렵다
- 진짜 객체가 느리다.
- 진짜 객체가 사용자 인터페이스를 갖거나 그 자체이다.
- 진짜 객체가 아직 없다.

□ Mock 사용 종류

- 이미 구현된 Mock 객체 사용 : Spring test (web, jndi), mockrunner
- EasyMock, JMock, Mockito 등의 Mock 라이브러리 사용
- Mock으로 사용할 가짜 클래스 직접 구현



□ Easymock + Unitils 샘플

```

1  @RunWith(UnitilsJUnit4TestClassRunner.class)
2  public class EgovControllerTest {
3
4      @Mock
5      @InjectIntoByType
6      private EgovService mockService;
7
8      @TestedObject
9      private EgovController egovController = new EgovController();
10
11     @Test
12     public void testSelectList() throws Exception {
13         assertNotNull(egovController);
14         expect(mockService.selectList()).andReturn(
15             Arrays.asList(new Message(101), new Message(102)));
16         EasyMockUnitils.replay();
17
18         List<Message> resultList = egovController.selectList();
19         assertNotNull(resultList);
20         assertEquals(2, resultList.size());
21     }
22
23     @Test
24     public void testInsert() throws Exception {
25         mockService.insert(makeVO());
26         EasyMockUnitils.replay();
27
28         SessionStatus status = new SimpleSessionStatus();
29         String view = egovController.add(makeVO(), status);
30     }
31 }

```

Mock으로 생성할 인터페이스에 대한 정의한 뒤, 테스트 대상 클래스에 Mock으로 정의한 클래스를 세팅한다.

리턴값을 갖는 메소드를 Mock 클래스에 정의하고 리턴값을 설정한 뒤, 진짜 객체처럼 사용한다.

리턴하지 않는 메소드를 Mock 객체에 정의 후 진짜 객체처럼 사용한다.

배치 테스트 도구를 통해 기 개발된 일괄처리 모듈에 대한 Junit 테스트 파일생성 및 테스트 수행과 함께, 기존에 생성한 Junit 테스트 파일을 재실행할 수 있는 기능을 제공

The screenshot shows the 'eGovFrame Batch Job Test' window. It has a title bar with standard Windows controls. The main area is titled 'Customize Batch Job Test Option' with a subtitle 'Batch Job Test를 수행할 조건을 설정하십시오.' Below this, the 'Project Name' is set to 'file.sche.egov.temp'. There are five red boxes with numbers and Korean text pointing to specific parts of the interface:

- ① Job 정보 선택: Points to a list of test jobs on the left.
- ② Job Launcher 정보 선택: Points to a table with 'File Name' and 'File Location' columns.
- ③ Job Parameter 정보 선택 (선택사항): Points to a 'Job Parameters' table.
- ④ 배치 테스트 파일 생성: Points to the 'Generate Batch Test File' button.
- ⑤ 배치 테스트 수행: Points to the 'Test' button.

At the bottom, there is a 'Result' section showing '<eGovFramework Batch Test>' and 'Batch Execution Status [OK]' with a green progress indicator. 'Finish' and 'Cancel' buttons are at the very bottom.

File Name	File Location
context-batch-job-launcher.x...	/src/main/resources/egovframework/batch/d
newJobLauncher.xml	/src/main/resources/newJobLaunc

Name	Value	Data Type
name	eGovFrame	String

□ 배치 테스트 절차

- ① 배치 Job 정보를 선택한다.
- ② Job Launcher 정보를 선택한다.
- ③ 필요 시 Job Parameter 정보를 선택한다.
- ④ Generate Batch Test File 버튼을 클릭하여 배치 테스트 파일을 생성한다.
- ⑤ Test 버튼을 눌러 배치 테스트를 수행한다.

DB Test는 **DAO**와 **DB** 모두를 통틀어 **Persistence layer**를 테스트하는 것을 말하며, 테스트DB와 테스트 데이터를 준비하여 실제로 **DB**를 이용한 단위테스트를 수행함

□ DB Test Framework

- DbUnit

□ DbUnit 기능

- DB 데이터를 XML 파일 형태로 import/export
- DB 연결, DB 초기화
- DB의 데이터가 기대값과 같은지 확인
- 빌드 도구를 통한 테스트 자동화 기능 제공

□ 효율적인 DB Test를 위해

- DB 구조만 갖고 있는 단위테스트용 DB 준비
- 각 개발자마다 별도의 단위테스트용 DB 공간
- Unitils, Spring Test 활용

작성된 다수의 **TestCase**를 자동으로 수행하여 소스 단위의 품질을 높이하고자 함. **CI** 서버와 연계될 때, 주기적인 테스트 수행으로 인해 목표 시스템의 품질을 향상할 수 있음

□ Test Suite

- TestCase 모음

1. Test Suite Class 작성

```
@RunWith(Suite.class)
@SuiteClasses( { HttpRequestMockTest.class, SessionMockTest.class,
FileUploadMockTest.class, EmailMockTest.class, JDBCMockTest.class })
public class MockTestSuite {

}
```

□ Test Suite 설정 방법

- Test Suite Class 작성
- 빌드 도구의 batchtest 묶음

2. batchtest로 묶음

```
<junit . . .>
  <batchtest fork="yes" todir="${testreports.dir}">
    <fileset dir="${testbuild.dir}">
      <include name="**/*Test.class" />
      <exclude name="**/Abstract*Test.class" />
    </fileset>
  </batchtest>
</junit>
```

❑Ant (build.xml)

```
<!-- Junit Test Result Report -->
<target name="junitreport" depends="tests">
  <junitreport todir="${testhtml.dir}">
    <fileset dir="${testreports.dir}">
      <include name="TEST-*.xml" />
    </fileset>
    <report format="frames" todir="${testhtml.dir}" />
  </junitreport>
</target>
```

❑Maven (pom.xml)

```
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-report-plugin</artifactId>
      <version>2.4.2</version>
    </plugin>
  </plugins>
</reporting>
```

Unit Test Results

Designed for use with [JUnit](#) and [Ant](#).

Summary

Tests	Failures	Errors	Success rate	Time
60	0	26	56.67%	65.561

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Packages

Name	Tests	Errors	Failures	Time(s)	Time Stamp	Host
org.springframework.samples.petclinic	1	0	0	0.672	2009-03-11T02:22:19	lgns-juroh
org.springframework.samples.petclinic.hibernate	10	0	0	8.453	2009-03-11T02:22:23	lgns-juroh
org.springframework.samples.petclinic.jdbc	10	0	0	5.109	2009-03-11T02:22:35	lgns-juroh
org.springframework.samples.petclinic.ipa	39	26	0	51.327	2009-03-11T02:22:44	lgns-juroh

egovframework-dev-tst

Last Published: 2009-03-12

Summary

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
74	7	0	5	83.784%	60.845

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Package List

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
egovframework.dev.samples.guestbook.category.service	6	6	0	0	0%	2.984
egovframework.dev.samples.mock	3	0	0	0	100%	4.36
egovframework.dev.samples.guestbook.controller	4	0	0	0	100%	2.625
egovframework.dev.samples.junit4	10	0	0	2	80%	0.375
egovframework.dev.samples.guestbook.service.impl	6	0	0	0	100%	5.47
egovframework.dev.samples.guestbook.service.impl.jdbc	28	1	0	3	85.714%	25.561
egovframework.dev.samples.guestbook.service.impl.ibatis	8	0	0	0	100%	6.61
egovframework.dev.samples.guestbook.service.impl.hibernate	9	0	0	0	100%	12.86

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

egovframework.dev.samples.guestbook.category.service

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
EgovTxServiceImplTest	6	6	0	0	0%	2.984

□Ant – Default로 사용한 예

```
<path id="egov.lib">
  <path refid="master-classpath" />
  <path location="${antlib.dir}/egovtest/egovframework-dev-tst-ant.jar" />
</path>

<taskdef resource="egovtest.properties" classpathref="egov.lib" />

<!-- JUnit Excel Report -->
<target name="egovtest" depends="tests">
  <egov-junitreport todir="${testxls.dir}">
    <fileset dir="${testreports.dir}" includes="**/TEST-*.xml" />
  </egov-junitreport>
</target>
```

Excel Reporting Ant Task 설정

Default 사용예

□Ant – 템플릿 엑셀 파일을 별도로 사용한 예

```
<target name="egovtest-full" depends="tests">
  <egov-junitreport todir="${testxls.dir}"
    outputname="egovtest-junit-full.xls"
    templatepath="${basedir}/build/template-kr.xls"
    summary="0,B,6"
    packages="0,B,11"
    lists="1,A,5">
    <fileset dir="${testreports.dir}"
      includes="**/TEST-*.xml" />
  </egov-junitreport>
</target>
```


❑ Maven (goal : egovtest:junit-xls)

```
<build>
  <plugins>
    <!-- Egovframework JUnit Excel Reporting -->
    <plugin>
      <groupId>egovframework.dev</groupId>
      <artifactId>egovtest-maven-plugin</artifactId>
      <version>1.0.0-SNAPSHOT</version>
    </plugin>
  </plugins>
</build>
```

Microsoft Excel - egovtest-junit.xls

파일(F) 편집(E) 보기(V) 삽입(I) 서식(O) 도구(T) 데이터(D) 창(W) 도움말(H) 질문을 입력하십시오.

119 fx 100% 도움말

	A	B	C	D	E	F	G
1	JUnit Test Results Summary						
2		2009-03-27					
3	Summary						
4	name	Tests	Errors	Failures	Skipped	Success Rate	Time (s)
5	all classes	44	6	0	4	77.27%	36.172
6							
7	Package						
8	name	Tests	Errors	Failures	Skipped	Success Rate	Time (s)
9	egovframework.dev.samples.guestbook.category.service	6	6	0	0	0.00%	2.984
10	egovframework.dev.samples.junit4	6	0	0	2	66.67%	0.156
11	egovframework.dev.samples.guestbook.service.impl	5	0	0	0	100.00%	3.313
12	egovframework.dev.samples.guestbook.service.impl.jdbc	18	0	0	2	88.89%	16.859
13	egovframework.dev.samples.guestbook.service.impl.hibernate	9	0	0	0	100.00%	12.860
14							

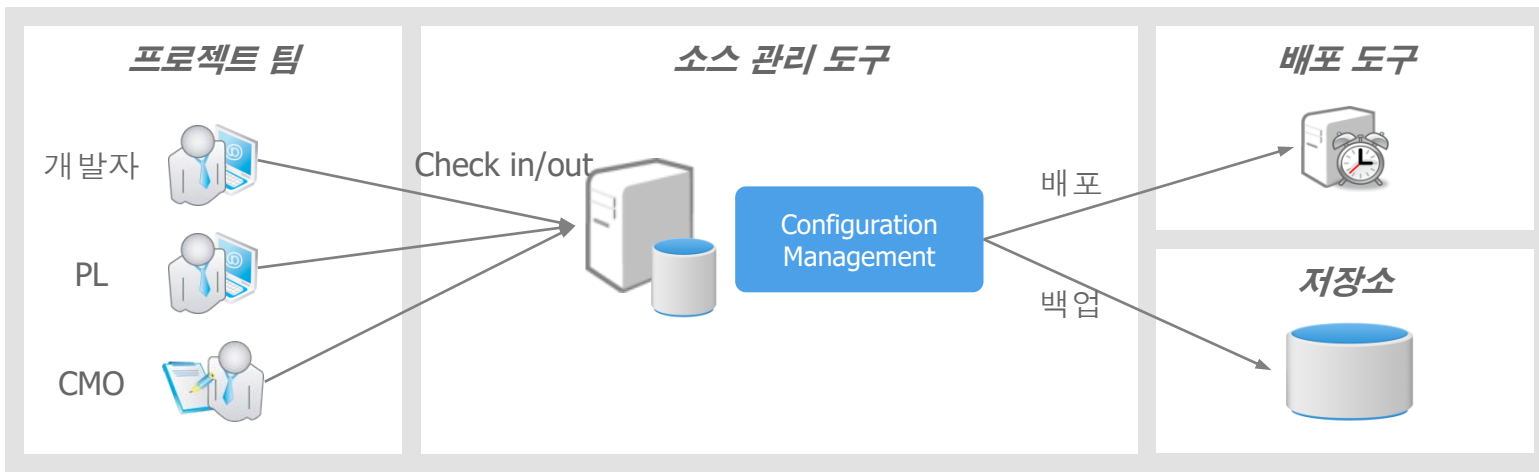
Summary / Lists / 준비 NUM

2. 형상관리



1. 개요
2. Subversion 개요
3. 소스 공유
4. Checkout
5. Update
6. Commit
7. History 조회
8. 충돌해결 (Merge)

- ❑ 소스 버전관리 도구는 시스템 형상 요소(소스 및 데이터)를 문서화하고 변경을 소스 버전관리 절차에 따라 관리하고, 효율적으로 처리 하기 위한 통제 환경을 의미
- ❑ 소스 버전관리 도구의 특징 및 기능
 - 형상 요소(소스 및 데이터)에 대한 변경 사항을 관리
 - 버전 관리를 수행하고, 배포도구를 통해 변경사항을 배포
 - 일정 주기에 백업 도구를 통해 백업을 수행
- ❑ 소스 버전관리 도구의 구성요소 및 관계도



❑ Subversion(SVN)의 배경

- CVS의 단점을 개선하고 CVS를 대체할 목적으로 개발되었음
- 오픈소스 기반의 소스 버전관리 도구 (<http://subversion.tigris.org>)

❑ CVS 대비 Subversion의 특징점

- CVS와 거의 동일한 사용법. CVS 사용자는 누구나 쉽게 사용할 수 있음
- Commit 단위: CVS는 개별 파일별로 리비전 관리, SVN은 변경이 발생하는 체인지셋(Change Set)별로 리비전 관리
- Atomic Commit: 여러 파일 커밋시, 어느 하나의 파일에서 커밋이 실패할 경우, CVS는 해당 파일 앞의 경우는 커밋되고, 뒷 파일은 그대로 남아있지만, SVN은 모두 이전 상태로 롤백
- 트리별, 파일별 접근 제어 리스트 지원
- 파일에 대한 이름변경/이동, 디렉터리 버전 관리도 지원
- CVS에 비해 빠른 동작 속도(Update/Branching/Tagging Time)

□ SVN 클라이언트 - Eclipse 플러그인

– Subversive

- Polarion 커뮤니티 그룹에서 개발
- Eclipse Ganymede 버전부터 SVN 클라이언트로 공식 채택
- 빠른 update, 안정적인 동작
- 공식 웹 페이지: <http://www.eclipse.org/subversive/>

– Subclipse

- Tigris 커뮤니티 그룹(Subversion 개발 커뮤니티)에서 개발
- Subversion 초창기 폭넓은 사용자 확보
- Subversive 대비 느린 update, 다수의 버그로 불안정
- 공식 웹 페이지: <http://subclipse.tigris.org/>

❑ 저장소(Repository)

- 모든 프로젝트의 프로그램 소스들과 소스에 대한 변경 사항이 저장 됨
- 네트워크를 통해서 여러 사람이 접근
- SVN에서는 Berkeley DB(Default) 또는 파일시스템(Optional)을 저장소로 사용

❑ 저장소 접근

① Eclipse 메뉴,
Window > Open Perspective
> Other... >
SVN Repository Exploring

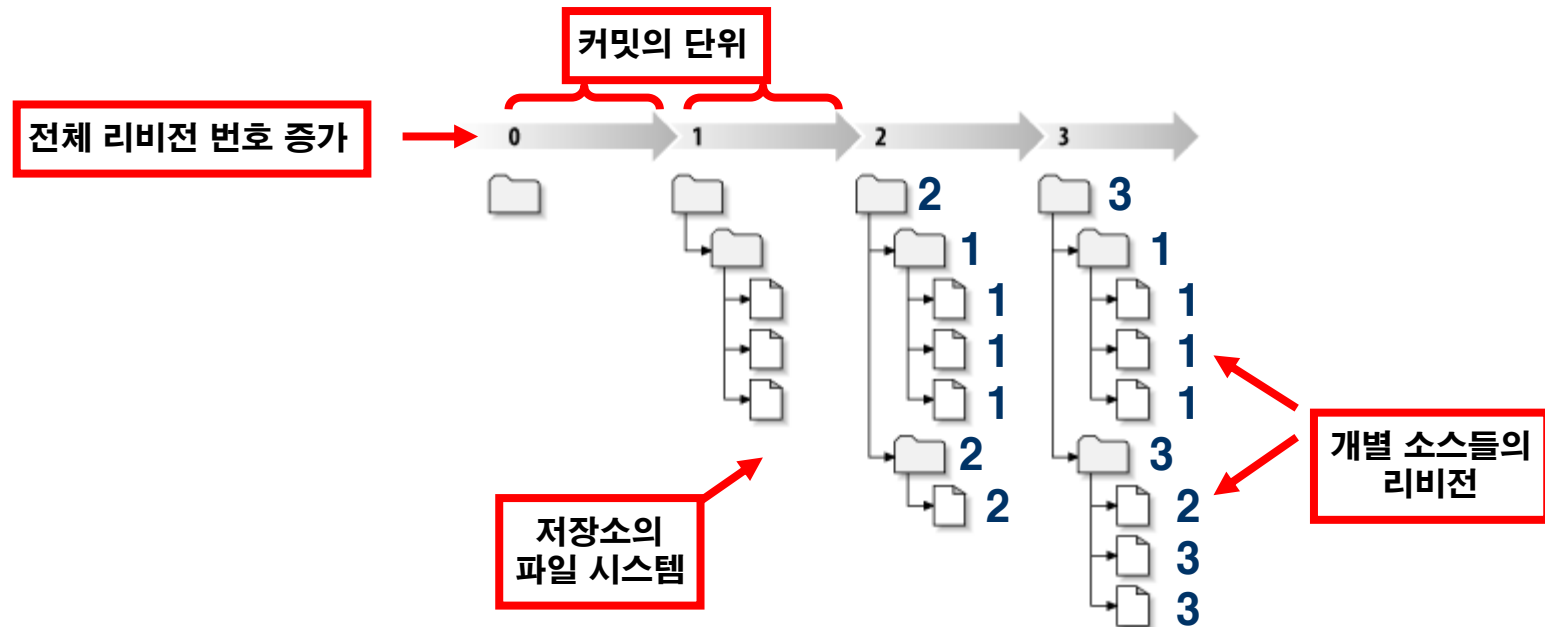
② Eclipse 메뉴,
File > New >
Repository Location

③ 마법사에서,
URL과 Authentication 항목을 입력

❑ Revisions

- 소스 파일 등을 수정하여 커밋하게 되면 일정한 규칙에 의해 숫자가 증가
- 저장소에 저장된 각각의 파일 버전이라 할 수 있음
- SVN의 경우 파일 별로 리비전이 매겨지지 않고 변경발생 단위로 전체 리비전이 매겨 짐
- 리비전을 보고 프로젝트 진행 상황을 알 수 있음

❑ 저장소를 통해 본 리비전 이해



❑ 프로젝트 공유

- 새로운 프로젝트를 저장소에 추가

① Eclipse 패키지/네비게이션 뷰, 프로젝트 오른쪽 클릭 > Team > Share Project...

Select a repository type:

- CVS
- File System Example (Non-Versioning)
- Pessimistic Simple Provider (Non-Versioning)
- SVN

② Share Project 마법사, SVN 선택

Share Project Wizard

Share Project with SVN repository

Select an existing repository location or create a new location.

This wizard will help you to share your files with the SVN repository for the first time. Your project will automatically be imported into the SVN repository.

☐ Create a new repository location

☒ Use existing repository location:

Label	URL
https://192.168.100.14:8443/svn/egovdev	https://192.168.100.14:8443/svn/egovdev
https://192.168.100.14:8443/svn/freework	https://192.168.100.14:8443/svn/freework

③ Share Project 마법사, 저장소 위치 선택 > Finish

□ 소스 단위 공유

- 공유 중인 프로젝트 내에서 해당 소스를 저장소에 추가

① Eclipse 패키지/네비게이션 뷰, 대상 소스 오른쪽 클릭 > Team > Add to Version Control...

② 팝업 창에서 내용 확인 > OK

Add to SVN Version Control

Selected resources will be added to the source control

There is 1 resource that is not under SVN version control. Are you sure you want to add it? The operation will be applied only to the resources which are selected in the list represented below.

Resource	Content	Properties
<input checked="" type="checkbox"/> egovframework.dev.tst.togenerator/src/egovfra...	New	

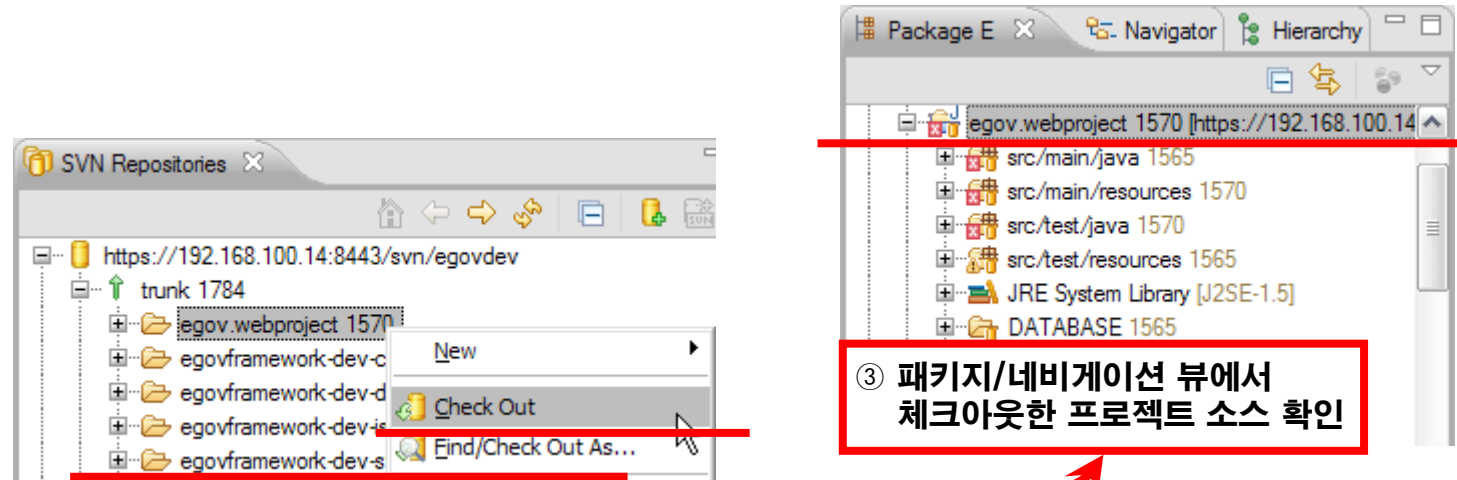
Select All Clear Selection

Selected: 1 of 1

OK Cancel

❑ 초기 checkout

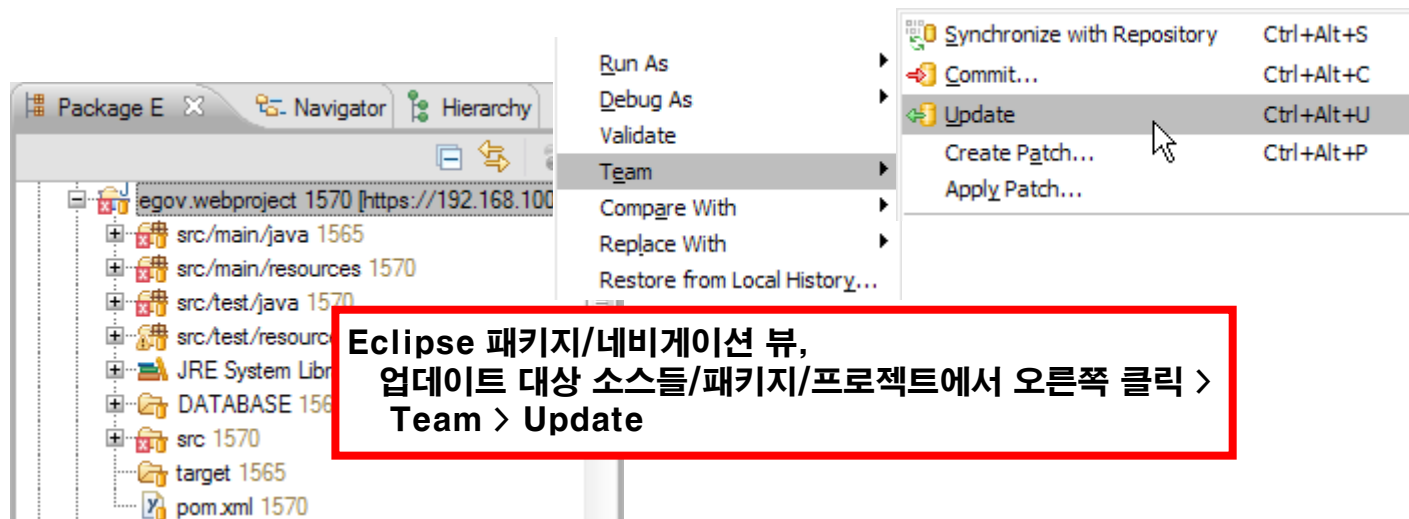
- 저장소에서 전체 소스의 최종 리비전을 받아오는 것
- 저장소 인증 정책에 따라 ID, Password 입력여부가 결정



- ① Eclipse 메뉴,
Window > Open Perspective
> Other... >
SVN Repository Exploring

❑ 저장소에 소스 업데이트

- 체크아웃을 한 이후의 타인에 의한 소스 변경사항을 확인
- 저장소에 있는 소스 중 로컬과 비교하여 변경된 항목의 최신 버전의 소스를 가져옴
- 변경된 항목의 최신 버전은 로컬 본에 통합됨



□ Commit

- 로컬에 체크아웃 한 소스를 수정, 파일 추가, 삭제 등을 한 뒤 저장소에 저장하여 갱신 하는 것
- 커밋을 하면 전체 리비전이 1 증가(CVS의 경우 수정한 각각 파일의 리비전이 증가)

② 팝업 창에서 반영 내용 확인 및 Comment 입력

① Eclipse 패키지/네비게이션 뷰, 커밋 대상 소스들/패키지/프로젝트에서 오른쪽 클릭 > Team > Commit...

Commit dialog box details:

- Comment: [Text area for commit message]
- Choose a previously entered comment or template: [Dropdown menu]
- Keep Locks: [Checkbox]
- Paste selected names: [Button]
- Resource list:

Resource	Content	Properties
egovframework.dev.tst.tcgenerator/lib/easysql.jar	Deleted	
egovframework.dev.tst.tcgenerator/.classpath	Modified	
egovframework.dev.tst.tcgenerator/build.properties	Modified	
egovframework.dev.tst.tcgenerator/META-INF/MANIFEST...	Modified	
egovframework.dev.tst.tcgenerator/src/egovframework/de...	Modified	

Select All Clear Selection Selected: 5 of 5

OK Cancel

□ History

- SVN에 의해 관리되는 특정 소스와 관련된 과거 모든 리비전의 내역

□ History 조회

- Eclipse의 기본 카테고리인 'Team'의 'History' 뷰를 통해 조회할 수 있음

① Eclipse 패키지/네비게이션 뷰, 대상 소스 오른쪽 클릭 > Team > Show History

Revision	Date	C...	Author	Comment
*634	09. 4. 29. 오후 5:11	7	suyol_yun	query section maximize added
628	09. 4. 29. 오전 10:25	5	hjee	[no comment]
625	09. 4. 29. 오전 10:12	7	hyungjoe	In/Out 의 Class, Map 항목의 배타적 표시
621	09. 4. 24. 오후 12:38	7	hyungjoe	[no comment]
618	09. 4. 23. 오후 2:30	7	hyungjoe	1.바인딩 변수타입 선택...
610	09. 4. 22. 오전 10:43	1	hjee	[no comment]
608	09. 4. 21. 오후 1:31	7	hyungjoe	[no comment]

플러그인 버전 업데이트

ROOT

Name	Path
artifacts.xml	
content.xml	
site.xml	
egovframework.dev.tst.tctemplates.feature_1.0.0.v200...	
egovframework.dev.help.feature_1.0.0.v20090310001...	

리비전 로그 목록

선택된 리비전의 Comment

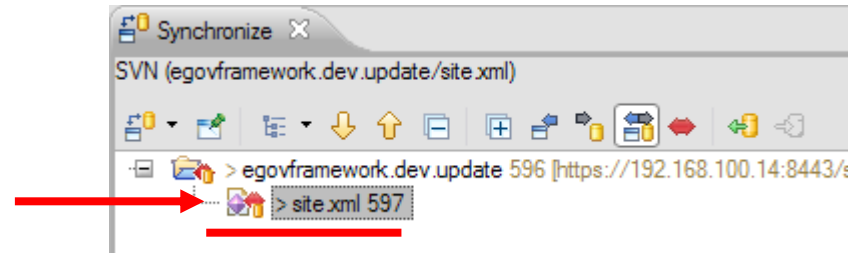
선택된 리비전의 구조와 변경 파일들

② History 뷰에서 대상 소스의 History 목록과 리비전 상세 내역 조회

❑ 충돌(Conflict)

- 로컬에 체크아웃 이후 수정한 소스를 저장소에 커밋할 때, 저장소의 리비전이 더 높을 경우 충돌 발생 (타인에 의해 수정되고 커밋된 상태)

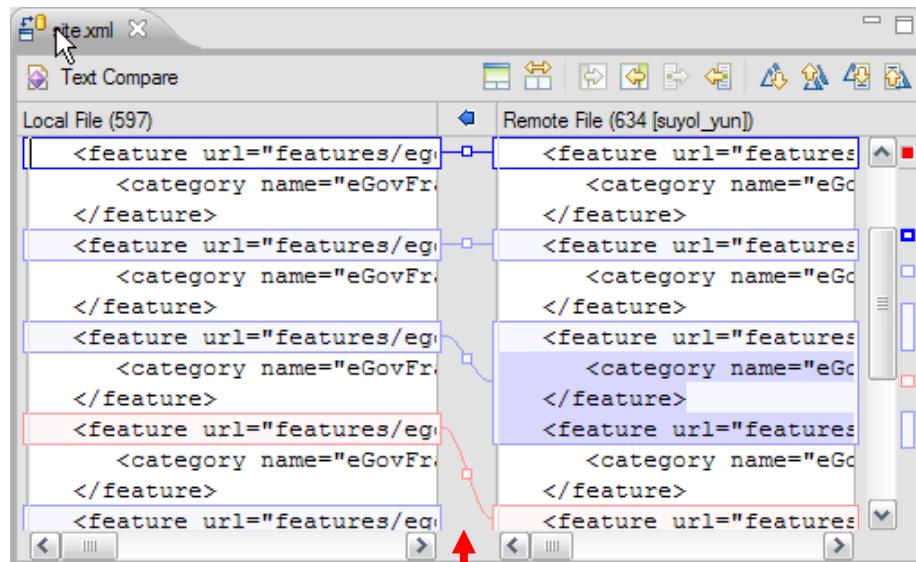
충돌 발생상황 예:
로컬 리비전 = 597 < 저장소 리비전 = 634



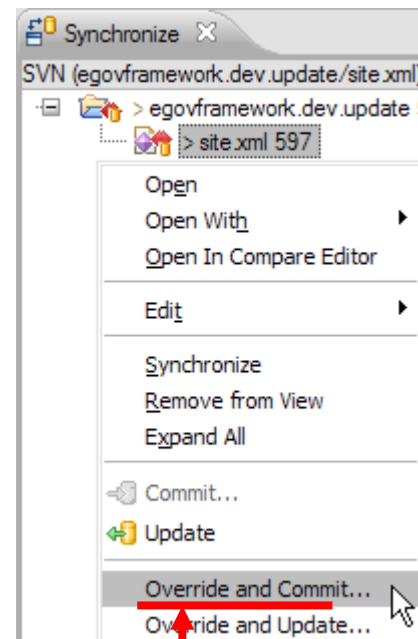
❑ 충돌 해결 방안

- postpone: 즉시 반영없이, 소스 코드를 계속 수정. 수정이 완료된 시점에 다시 시도 (충돌 해결 안함)
- diff: 충돌된 상황을 비교. 로컬과 저장소의 소스의 차이를 비교
- edit: 로컬의 파일을 다시 열어 수정
- mine-full: 저장소의 리비전 내용을 무시하고, 로컬의 내용으로 커밋.
- theirs-full: 로컬의 수정 내역을 무시하고, 저장소의 리비전으로 대체하여 업데이트

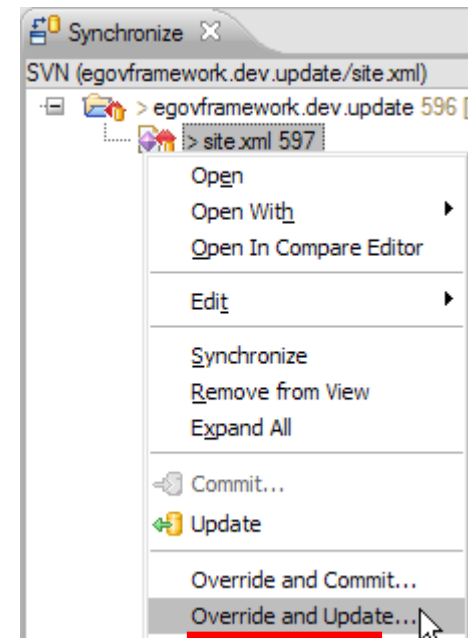
❑ 충돌 해결 방안 예



diff/edit:
로컬과 저장소의 소스를
모두 열어 비교/편집



mine-full:
충돌이 발생한 파일에서 오른쪽 클릭 >
Override and Commit... 선택



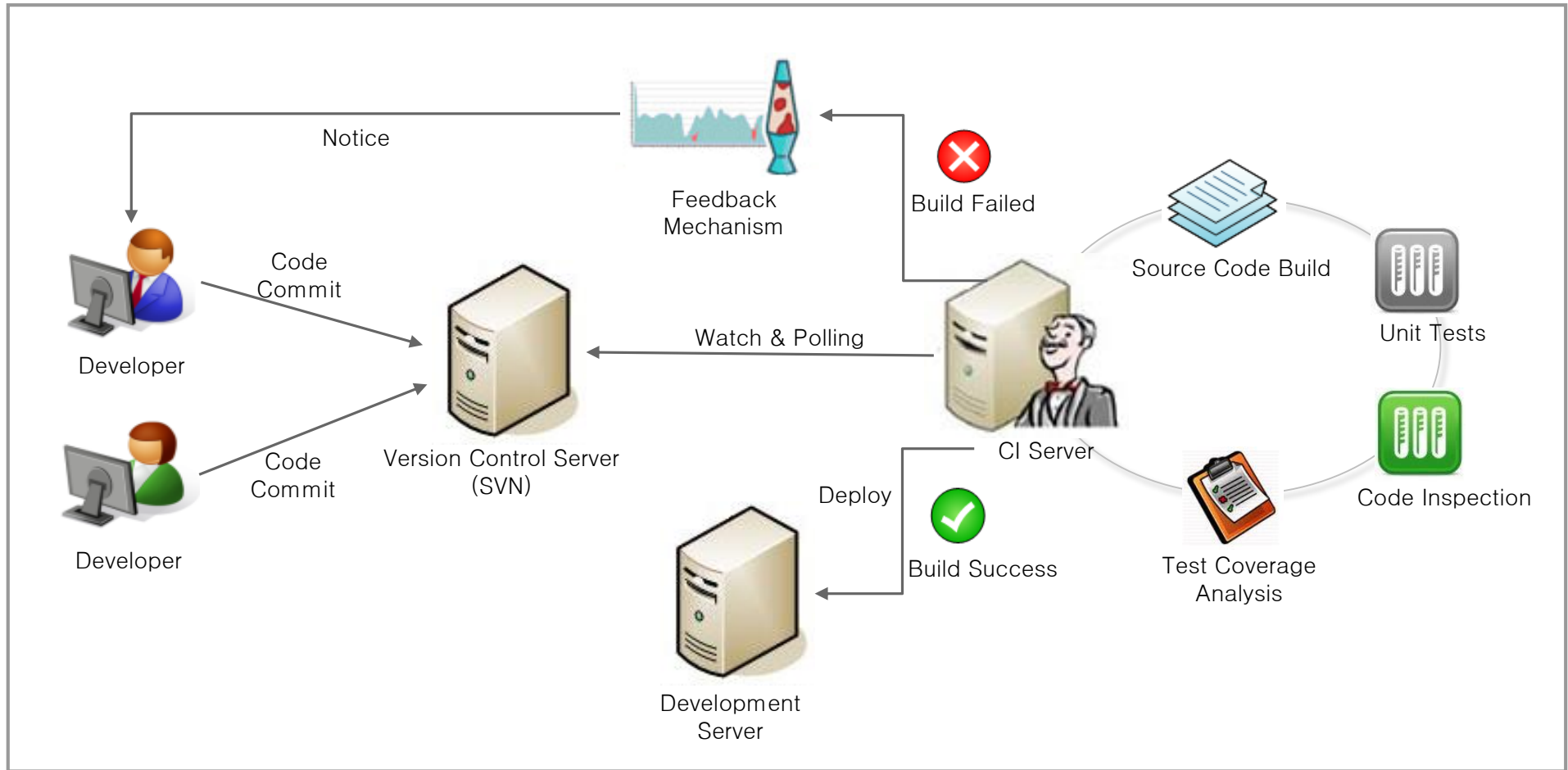
theirs-full:
충돌이 발생한 파일에서 오른쪽 클릭 >
Override and Update... 선택

3. CI 서버



1. CI 정의
2. CI 효과와 특징
3. Hudson 개요
4. Hudson 특징 및 주요기능
5. Hudson 대시보드
6. Hudson 프로젝트 메뉴
7. Hudson - Junit 테스트 연동
8. Hudson - 코드 인스펙션 도구 연동

여러 명으로 구성된 팀이 작업한 것을 자주-각 팀원이 하루에 한 번 이상, 매일 여러 번-통합하는 것을 가리키는 소프트웨어 개발 **Practice**로, 개발자, 버전 관리 저장소, 통합 빌드 머신 등으로 구성 됨



□ 효과

- 흔히 발생하는 일반적인 위험을 줄여준다.
- 에러를 초기에 발견할 수 있다. 자주 통합할수록 에러가 발생하는 범위가 좁아 에러를 잡는 것이 수월해진다.
- 잘 동작하고 에러가 거의 없는 S/W를 유지할 수 있다.
- 언제 어느 때라도 배포할 수 있는 S/W를 생성해낸다.
- 반복적인 수작업을 줄여준다.
- 프로젝트 가시성을 좋게 해준다.
- 개발 팀이 S/W 제품에 대해 보다 큰 자신감을 갖게 해준다.

□ 특징

- 소스 코드 일관성 유지 : 소스 관리 시스템이 필요하다.
- 자동 빌드 : 빌드는 CI 툴에 의해 자동으로 이루어져야 한다.
- 자동 테스트 : 소스 검증과 함께 기능/비기능 테스트를 매번 검증함으로써 코드의 품질을 높일 수 있다.
- 일일 체크아웃과 빌드 : 빌드 주기가 길어질수록 개발자 소스의 오류 발생 확률이 높아지고 그 만큼 시간을 낭비하게 된다.

Hudson은 오픈 소스 CI 서버로 소스 빌드 스크립트를 사용한 자동 빌드 및 빌드 결과를 개발자에게 피드백 하는 메커니즘을 제공함

Hudson

Hudson > DEV

새 작업

Hudson 관리

개발자

빌드 기록

Edit View

Delete View

Project Relationship

Check File Fingerprint

빌드 대기 목록

빌드 대기 항목 없습니다.

빌드 실행 상태

No.	상태
1	대기 중

All

DEV

RTE

+

S	W	작업 ↓	최근 성공	최근 실패	최근 소요 시간
		egovframework-dev-dep-14	2 days 18 hr (#47)	19 days (#30)	5 min 45 sec
		egovframework-dev-isu-14	2 days 18 hr (#43)	1 mo 9 days (#14)	4 min 31 sec
		egovframework-dev-tst	2 days 15 hr (#31)	—	1 min 25 sec
		egovframework-dev-tst-ant	2 days 15 hr (#30)	—	46 sec
		egovtest-maven-plugin	2 days 15 hr (#32)	—	1 min 4 sec

아이콘: S M L

Legend

for all

for failures

for just latest builds

자율 재실행 켜기

소개 내용 입력

초기화면: 프로젝트 리스트와 빌드 상태 및 빌드 성공 여부를 보여줌

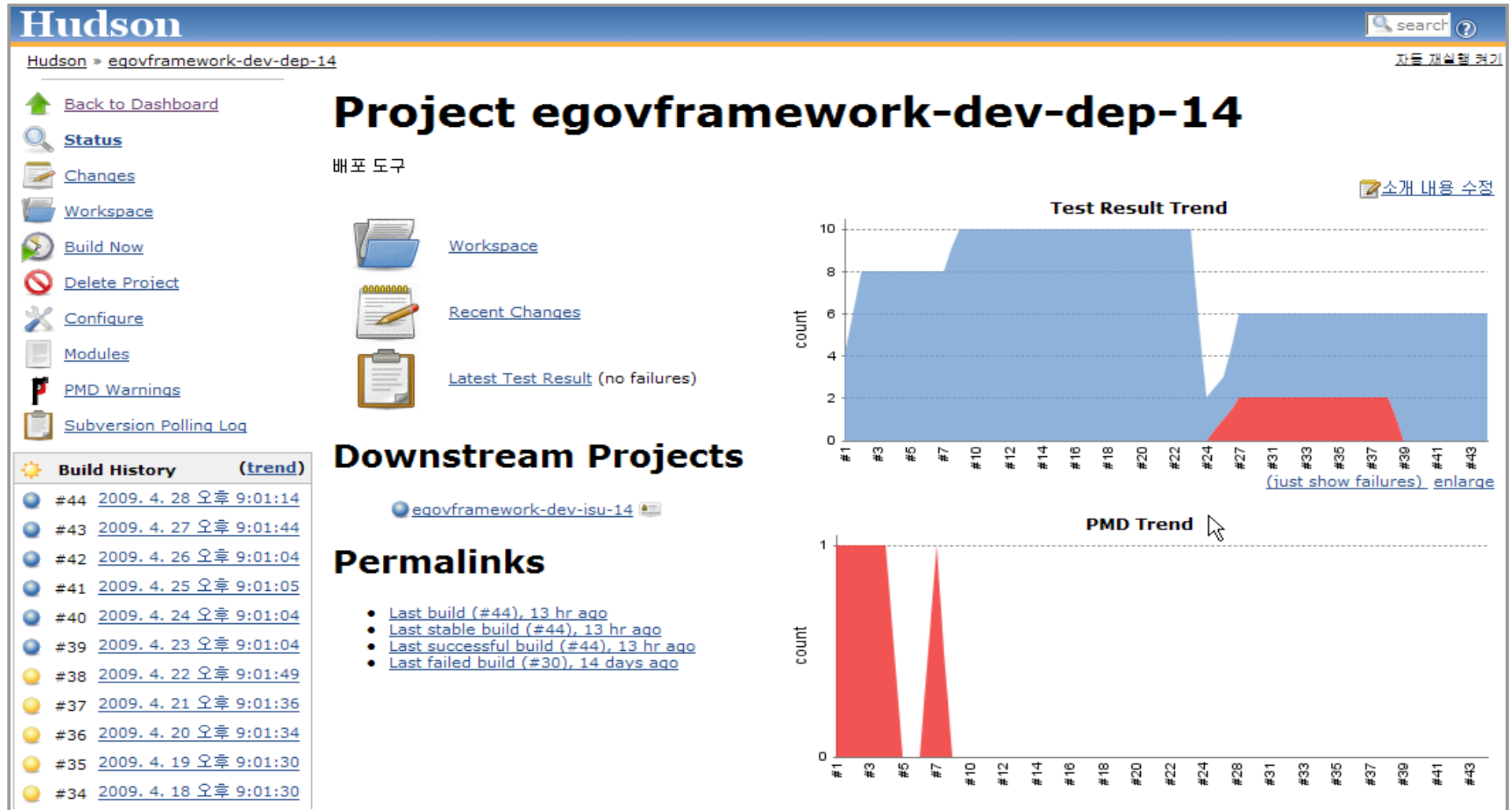
□ Hudson 특징

- 쉬운 Installation : hudson.war형태로 servlet container에 배포
- 웹 기반 UI를 통한 쉬운 설정 및 즉각적인 피드백
- 플러그인 기반의 확장성 제공
- 깔끔하고 잘 정의된 문서 및 API

□ Hudson 주요 기능

- 소프트웨어 자동 빌드
 - 빌드 주기에 따른 일일 빌드 또는 주간 빌드 기능 제공
- 지속적이고 자동화된 빌드 검증
 - SCM 폴링(polling) 기능을 통한 최신 코드 기반의 빌드 수행
- 지속적이고 자동화된 빌드 테스트
 - 테스트 스위트 실행을 통한 코드 품질 검증
- 빌드 후속 절차 자동화
 - 컴파일 된 코드의 패키징 및 테스트 리포팅 기능 제공

Hudson은 빌드 결과 및 테스트 결과 등을 보여 주는 대시보드 기능을 제공함



Hudson은 등록된 프로젝트의 소스 코드 변경 내용 추적을 위한 메뉴와 수동 빌드 메뉴 그리고 프로젝트 빌드 상태를 보여주는 메뉴 등을 제공함

Hudson » DEV » egovframework-dev-dep-14

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Workspace](#)
[Build Now](#)
[Delete Project](#)
[Configure](#)
[Modules](#)
[PMD Warnings](#)
[Subversion Polling Log](#)

Project egovfram

배포 도구

[Workspace](#)
[Recent Changes](#)
[Latest Test Result](#) (no failures)

Downstream Projects

[egovframework-dev-isu-14](#)

Permalinks

- [Last build \(#47\), 2 days 17 hr ago](#)
- [Last stable build \(#47\), 2 days 17 hr ago](#)
- [Last successful build \(#47\), 2 days 17 hr ago](#)
- [Last failed build \(#30\), 19 days ago](#)

Build History (trend)

#47	2009. 5. 1 오후 9:01:51
#46	2009. 4. 30 오후 9:01:30
#45	2009. 4. 29 오후 9:01:30
#44	2009. 4. 28 오후 9:01:14
#43	2009. 4. 27 오후 9:01:44
#42	2009. 4. 26 오후 9:01:04
#41	2009. 4. 25 오후 9:01:05

기본 메뉴

- **Changes**
: 빌드 버전별로 SCM에서 지난 버전에 비해서 변경된 내용에 대해 변경자와 커밋시에 개발자가 추가한 comment를 확인할 수 있음
- **Workspace**
: 프로젝트의 빌드 디렉터리로 브라우저를 통해 빌드에 사용된 파일 등을 확인할 수 있음
- **Build Now**
: 스케줄에 상관없이 지금 강제적으로 빌드
- **Build History**
: 빌드 수행 시기와 빌드 상태 및 성공 여부를 보여줌

Hudson은 JUnit 테스트 리포트를 출력해 주는 기능을 제공하며 테스트의 **Progress** 누적 그래프를 프로젝트 초기 화면에 출력함

JUnit 테스트 리포트

Test Result : egovframework.dev.dep

0 failures (±0)

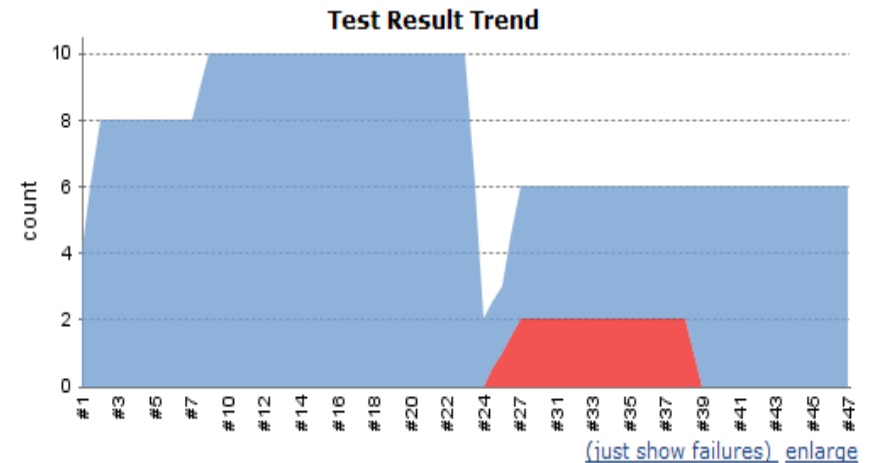
6 tests (±0)

All Tests

Class	Duration	Fail	(diff)	Total	(diff)
DeployModuleServiceTest	0.12 sec	0		2	
PropertyServiceTest	32 ms	0		2	
ScheduleServiceTest	77 ms	0		2	

프로젝트의 단위 테스트 결과 및 테스트 소요 시간 등에 관한 리포트를 보여줌

테스트 Progress 누적 그래프



프로젝트 빌드에 따른 JUnit 테스트 누적 그래프로 빌드 번호와 테스트 method 개수로 빌드 결과를 보여줌