

Continuous Integration for Staged Build (for Java)

Productivity Innovation Lab,
NHN Corp
2011.12

1. Course Introduction

1.1 Objective

- ❖ Understand NHN Quality Criteria
- ❖ Understand CI / Staged Build
- ❖ Learn how to set up Hudson(Jenkins) and mandatory plugins

1.2 WHO AM I

❖ JunHo Yoon

❖ Work Experience

- 2008~ Productivity Innovation Lab, NHN
- 2004~2008 SW Laboratory, Samsung Electronics

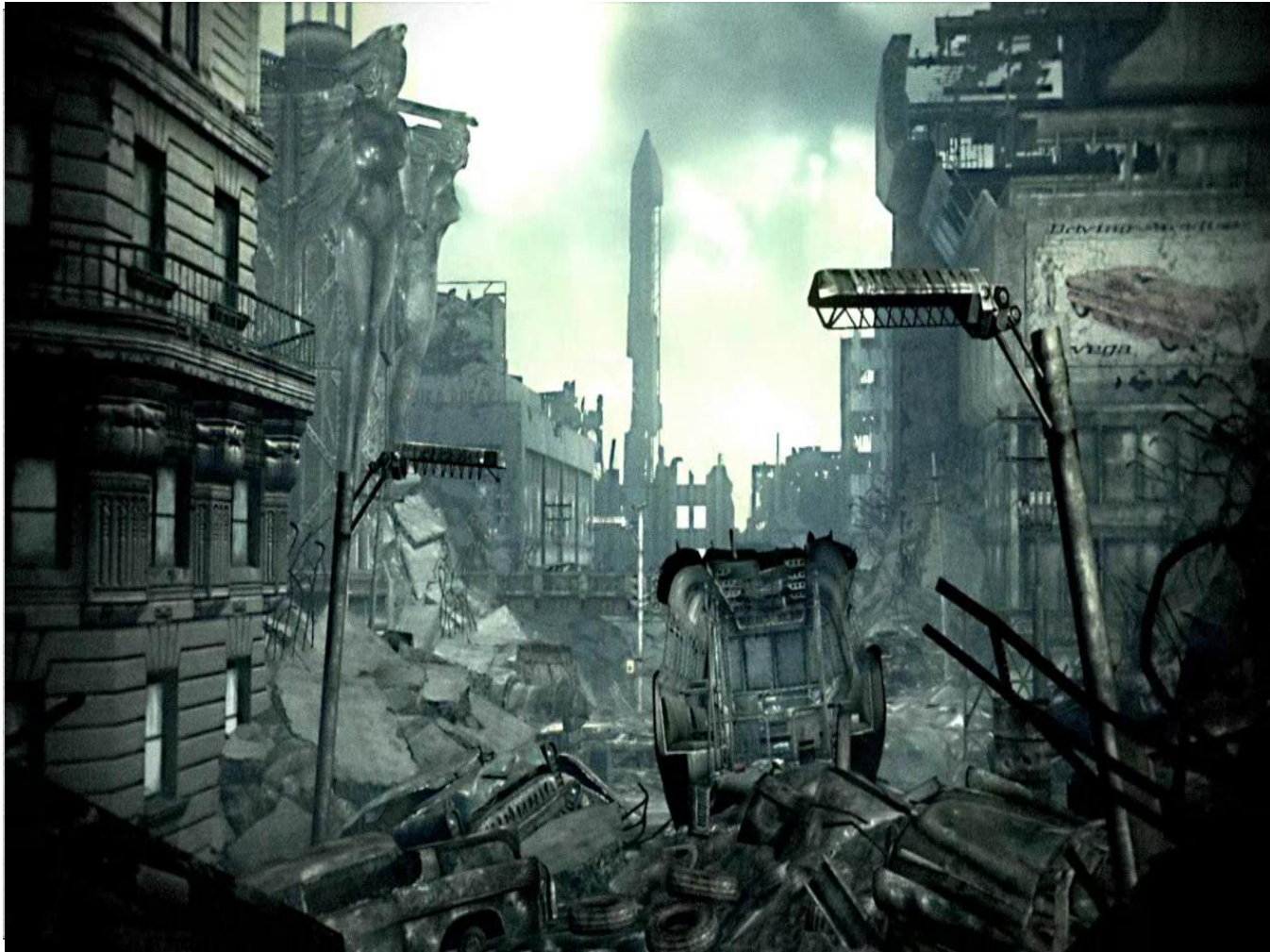
❖ Areas of expertise

- Software Engineering in SW Development Wide
- Enterprise Web Application Development
- Open Source Project

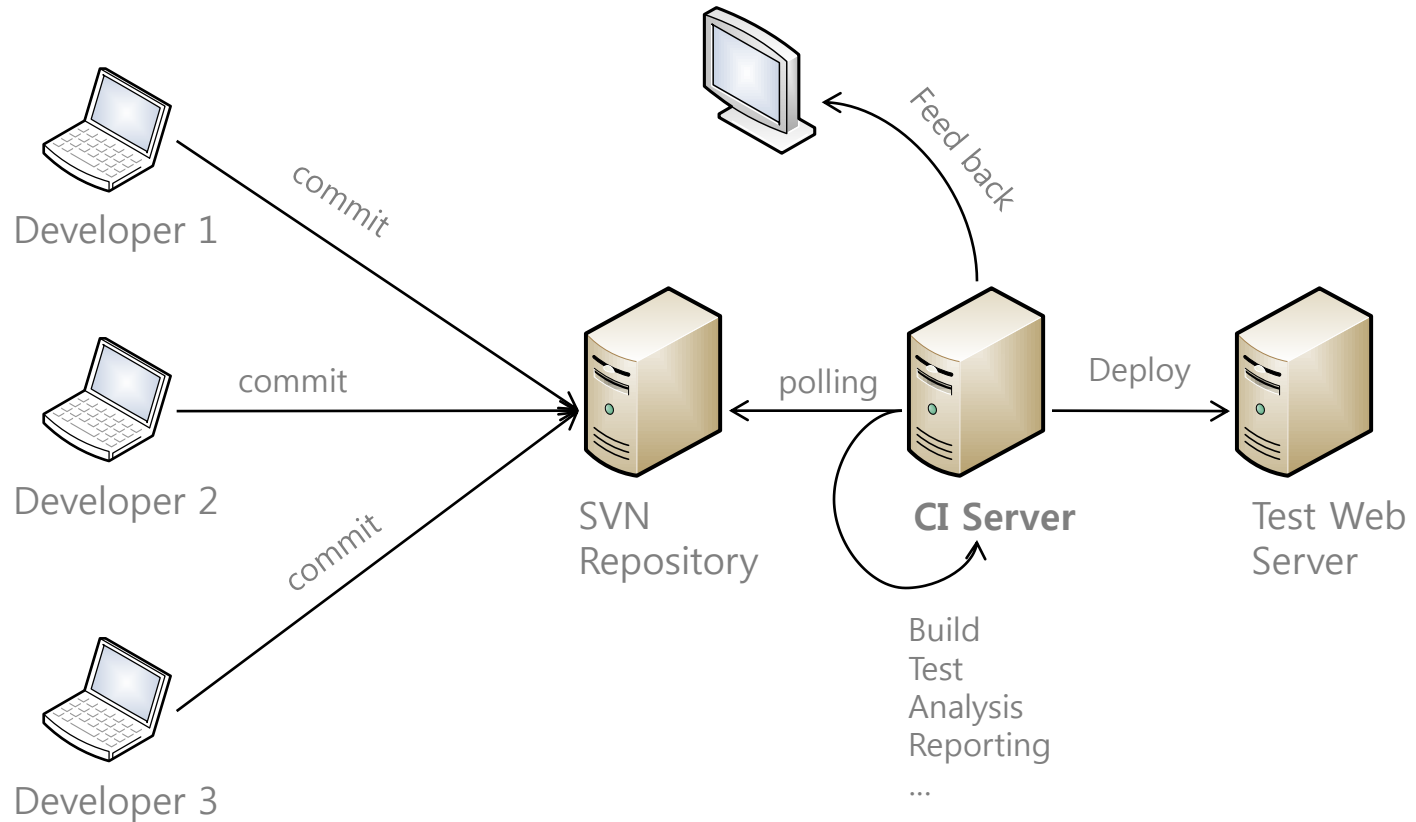
2. Staged Build



2.1 Broken Window Theory

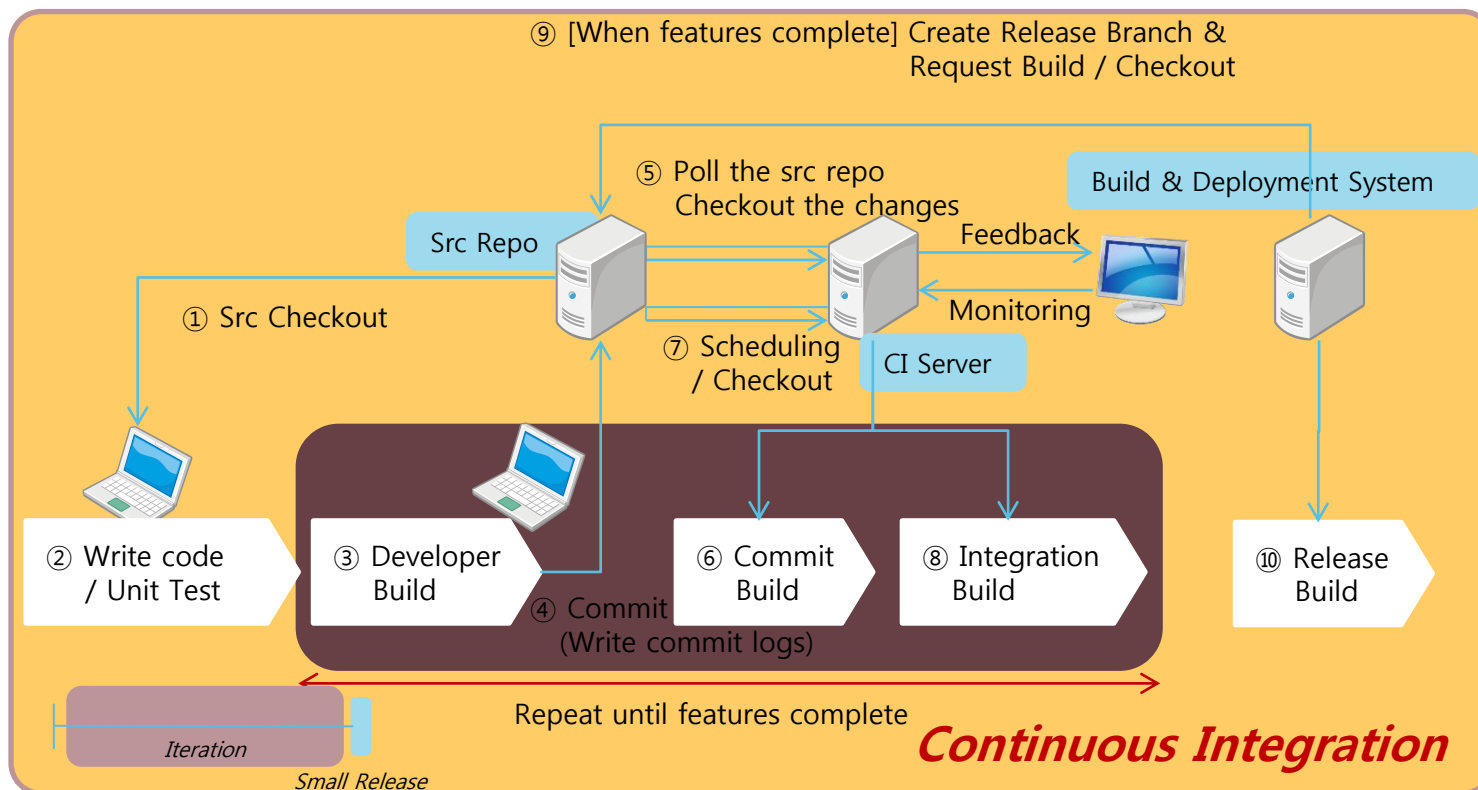


2.2 Continuous Integration(CI)



2.3 Staged Build

Build Integration procedure which is repeatedly performed to release working and tested SW at the end of each iteration until the development finishes
→ Advanced CI Usage



2.3 Staged Build

Commit Build

Builds which is performed in the separated build server whenever **code repository changes occur** to detect changes collisions from several developers.

- When**
- Schedule : Whenever developer commit changes info source repo.
 - Be able to change pooling timing(10min?)
 - Duration : within 10 min for build and test.

- What**
- Detect collision b/w changes from several developers.
 - Execute all unit tests developed by several developers
 - Tests which is independent from DB, Platform, Network,
 - DAO CRUD tests (if necessary)
 - Perform code reviews on major changes

- Post-Condition**
- Build success
 - All passed unit test



2.3 Staged Build

Integration Build

Builds which is performed in the separated build server periodically with more advanced integration test and code analysis

When	<ul style="list-style-type: none">• Schedule : Daily(Nightly). Change the interval if necessary.• Duration : with 2 hours
What	<ul style="list-style-type: none">• Configure build env and compile• Run unit and integration tests with real server like test environments.<ul style="list-style-type: none">- Tests dependent on middleware.- DAO CRUD tests / Automated UI Test- long time taking regression test• Run code analysis<ul style="list-style-type: none">- Coding Convention, Code Coverage, Static Analysis, Duplicate Analysis, Cyclomatic Complexity• (When finishing dev scope) Perform dev team own sanity test
Post-Condition	<ul style="list-style-type: none">• Build success• Satisfy Quality Practice Criteria<ul style="list-style-type: none">- Coding standard conformance rate, code coverage, static analysis defect rate• (When finishing dev scope) More than 90% pass rate of sanity test



2.3 Staged Build

Why separate commit build and integration build

Maximize CI Server operation efficiency

- ✓ Commit build is performed to detect collisions of committed code from several developers. Therefore fast feedback is necessary.
 - Minimum build activity
 - Needs to be real-time.

- ✓ Integration build is performed with full analysis and test to detect more defect behind. It takes time.
 - Maximize build activity
 - Not needs to be real-time.

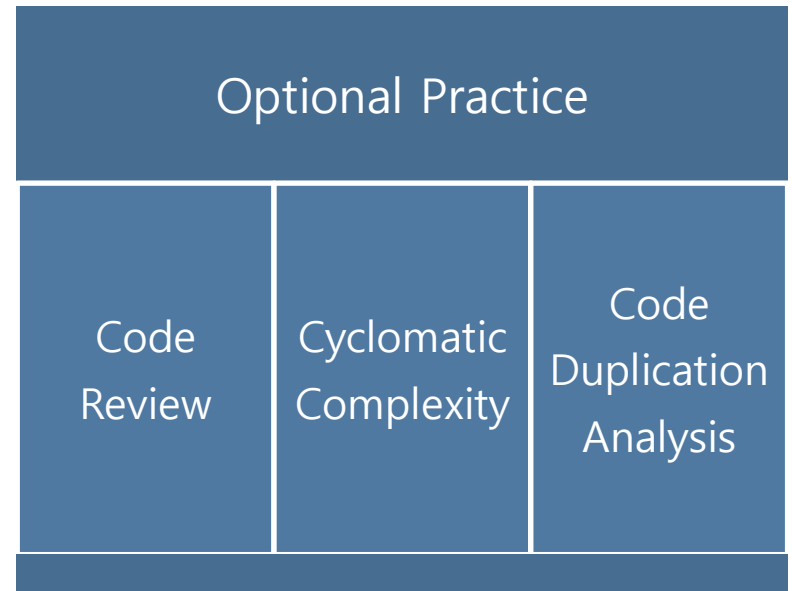
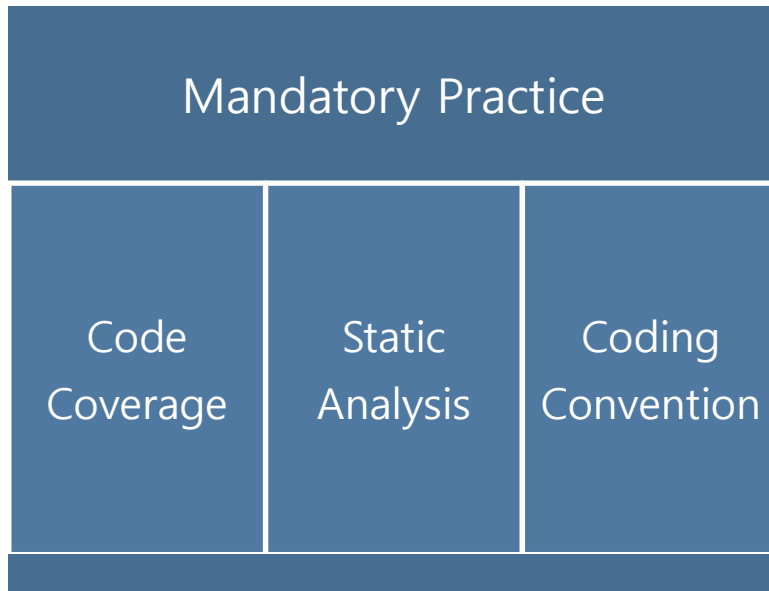
- By separating them,
Minimize CI server load and Give fast feedback to developers.

2.4 Quality Practices

Mandatory Practices and Optional Practices

❖ Quality Practices

- The minimum quality activities performed by NHN developers to guarantee the defect less SW
- Enforced from 2009



2.4 Quality Practices

Mandatory Practice - Code Coverage

❖ Should perform developer's automated tests and check the **code coverage** to improve code quality

- Test should be measurable and repeatable.
- Test coverages from not only Unit Test but also Integration test can be accumulated.

Metrics	Measures	Red	Yellow	Green	Gold
Statement Code Coverage(%) $A/B*100$	A. Tested Statements B. Statements	$COV < 30\%$	$30\% \leq COV < 50\%$	$50\% \leq COV < 70\%$	$70\% \leq COV$
Branch Code Coverage(%) $A/B*100$	A. Tested Branches B. Branches	$COV < 20\%$	$20\% \leq COV < 40\%$	$40\% \leq COV < 60\%$	$60\% \leq COV$

2.4 Quality Practices

Mandatory Practice - Static Analysis

❖ Should **remove Static analysis defects** found by Static

- Run static analysis tools and review the defect found.
- Run static analysis tools again when finishing development and Find out the status of defects remained.

Metrics1	Measures	Red	Yellow	Green	Gold
Static Analysis Defect Density (Count/KLOC) $A/B*100$	A. Weighted count of remained static analysis defect B. Total LOC	$6 \leq \text{Density}$	$4 \leq \text{Density} < 6$	$2 \leq \text{Density} < 4$	$\text{Density} < 2$

2.4 Quality Practices

Mandatory Practice - Coding Convention

❖ Should **confirm NHN Coding Standard** to keep the same style in newly created and modified code.

- Using the Custom NHN checkstyle extensions(Java)
- Using the N'SIQ CppStyle (C/C++)

Metrics	Measures	Red	Yellow	Green	Gold
Coding Standard Conformance Rate (CSCR / %) $A/B*100$	A. Count of files in which no violation found B. Count of total checked files	$CSCR < 30\%$	$30\% \leq CSCR < 70\%$	$70\% \leq CSCR < 90\%$	$90\% \leq CSCR$

2.4 Quality Practices

Optional Practice - Code Review

❖ Review the newly created and modified code

- How : Choose one of offline or online review
- What : Define criteria which code will be reviewed
- Review code as much as possible. However not mandatory.

Metrics	Measures	Red	Yellow	Green	Gold
Code Review Rate (%) $A/B*100$	A. Total LOC of files reviewed B. Total LOC of modified or created files	$CR < 30\%$	$30\% \leq CR < 60\%$	$60\% \leq CR < 80\%$	$80\% \leq CR$

2.4 Quality Practices

Optional Practice - Cyclomatic Complexity

❖ Identify the complexity code and **Reduce complexity**

- Find out complex methods which need to be refactored using tools
- Check the test coverage of complex methods
- Reduce complex methods. However not mandatory.

Metrics	Measures	Red	Yellow	Green	Gold
CC \geq 30 rate (%) = A/B	A. The count of CC \geq 30 method B. The count of total methods	Not measured or Not meet org goal	Meet org goal	< 0.5%	= 0%

- Cyclomatic Complexity : Simply the count of if / while / for statements per method

2.4 Quality Practices

Optional Practice - Code Duplication Reduction

❖ Reduce and Refactor duplicated code.

- Identify and prioritize highly duplicated code and refactor them with generalization/reuse/abstractions.
- Each project can perform Code Duplication Reduction based on their own decision,
Code Duplication Reduction is not the mandatory subject to be collected.

❖ Recommendation

- Reduce High Prioritized Duplicated code
- Reduce Normal, Low Prioritized Duplicated code by each own decision

High	Normal	Low
$50 > \text{Duplicated Line}$	$25 < \text{Duplicated Line} \leq 50$	$\text{Duplicated Line} \leq 25$

2.4 Quality Practices

Overall Rating Criteria

❖ Code Quality(CQ)?

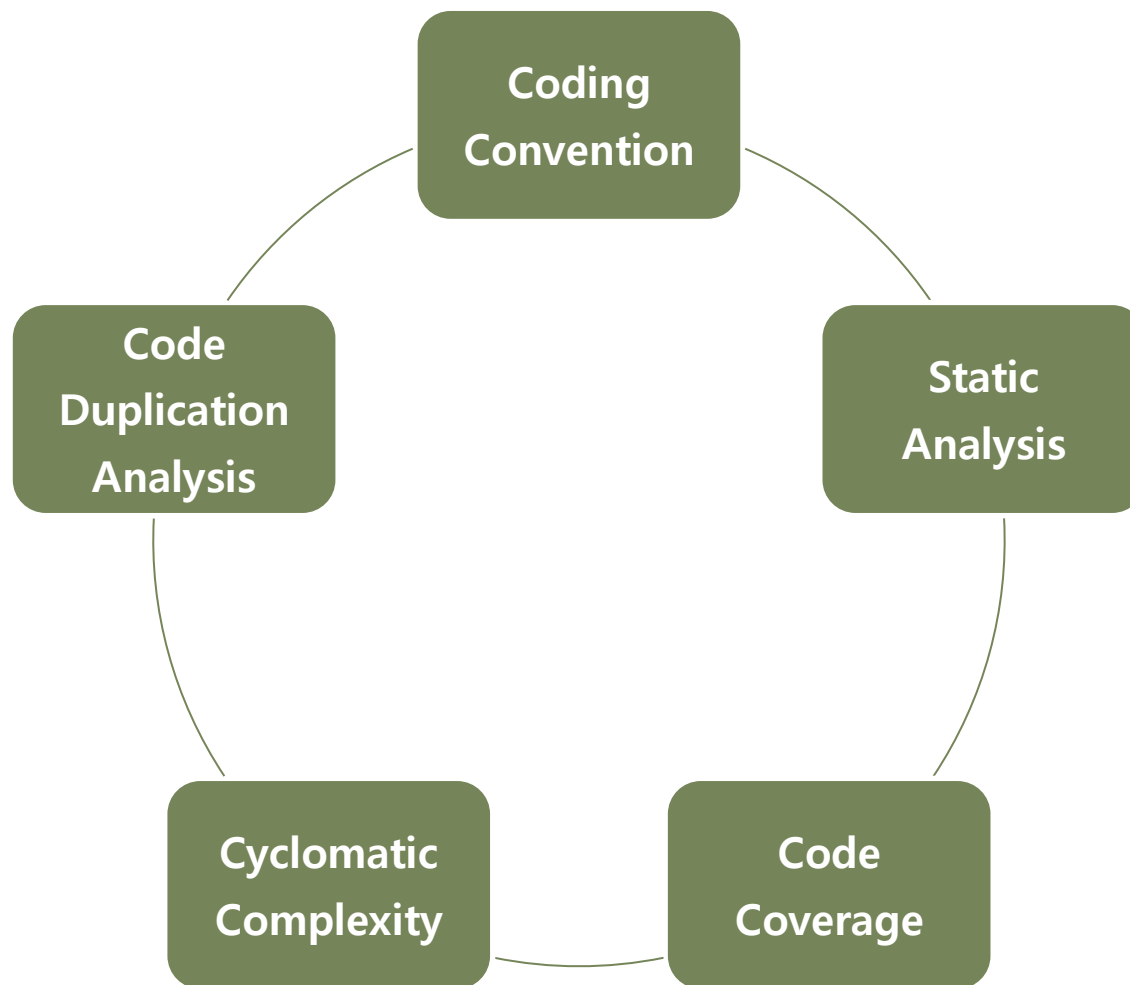
- Indicator to determine the result of Quality Practice performance efficiency
- Varies from development area.

❖ Code Quality Rating Criteria (e.g : Portal service)

Target Quality Metric	Weight	Red	Yellow	Green	Gold
Code Coverage	30				
Coding Standard Conformance Rate	20	CQ<20	20≤CQ<40	40≤CQ<50	50≤CQ≤70
Static Analysis Defect Density	20				

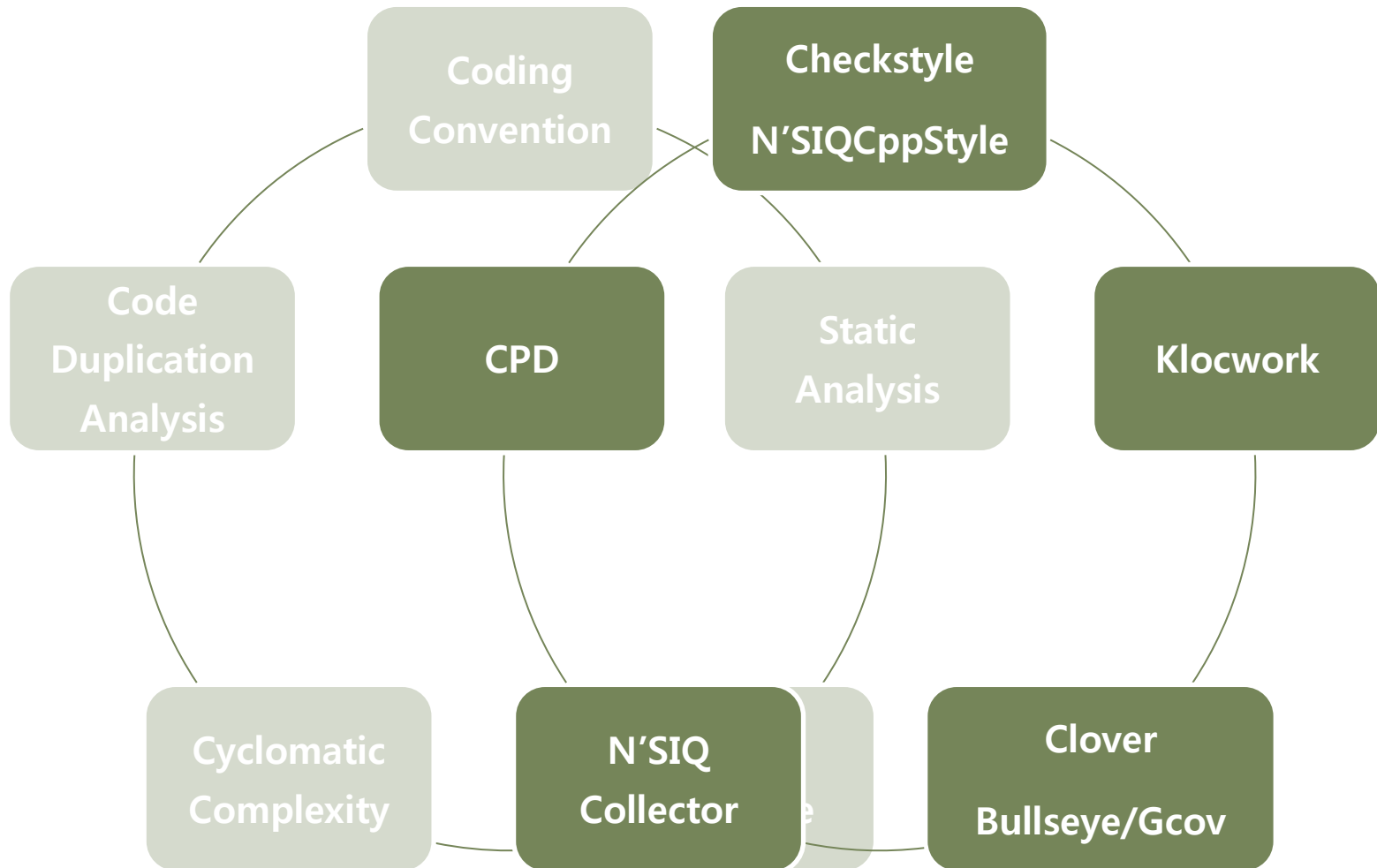
3. Hudson/Jenkins

3.1 Quality Practice on CI



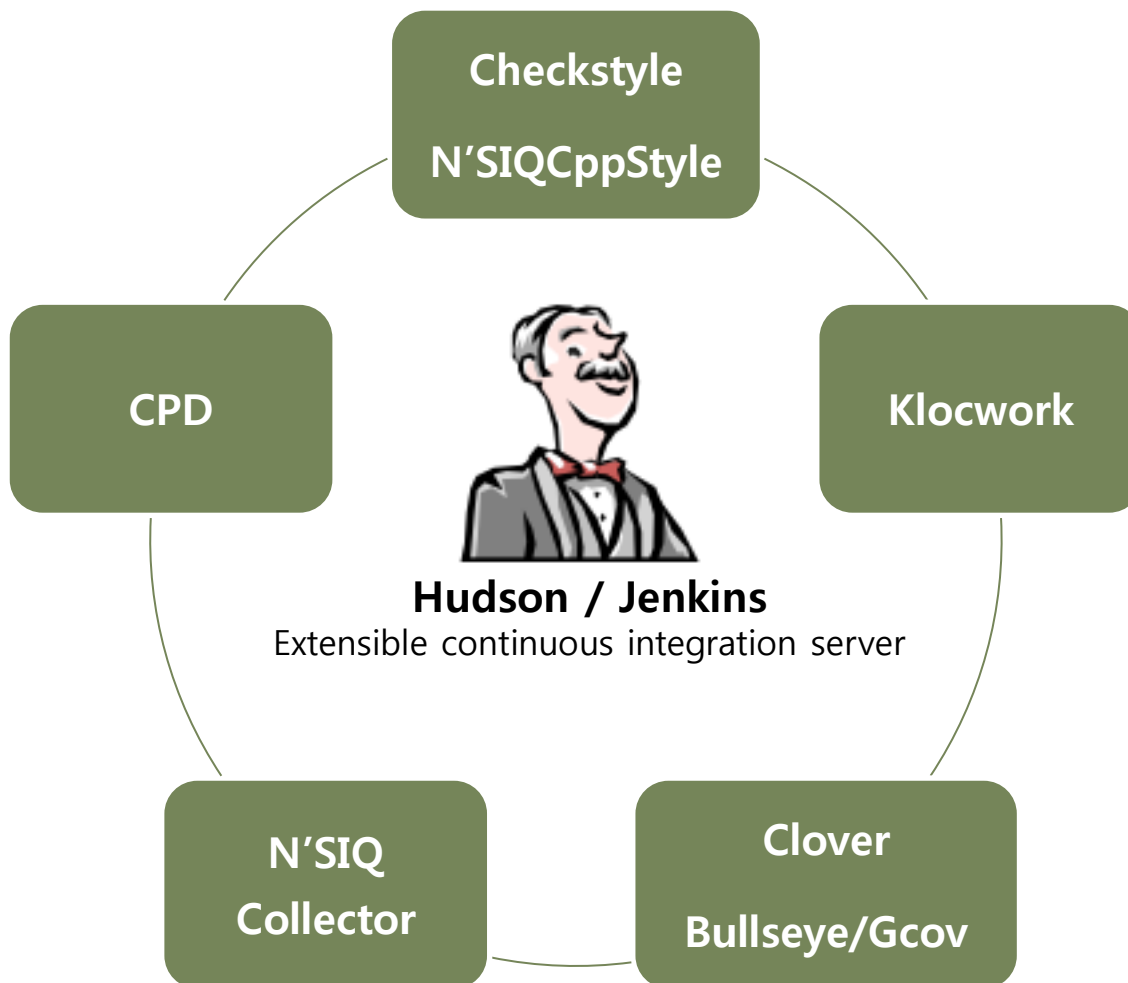
3.2 Quality Practice on CI

Tool support



3.2 Quality Practice on CI With Hudson

All tools are executed or collected by Jenkins



3.3 Hudson Introduction

About

- ❖ Jenkins monitors executions of repeated jobs, such as building a software project or jobs run by **cron**. Among those things, current Jenkins focuses on the following two jobs:
 - ❖ Current Jenkins focuses on the following two jobs:
 - Building/testing software projects continuously
 - Monitoring executions of externally-run jobs

Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#16)	9 months (#3)	39 seconds
bsh	6 months (#11)	10 months (#2)	59 seconds
dtd-parser	6 months (#8)	N/A	1 minute
fi	28 days (#586)	1 month (#567)	7 minutes
fi (weekly)	6 days (#53)	13 days (#52)	5 minutes
glassfish	4 hours (#104)	1 day (#88)	1 hour
hudson	4 minutes (#201)	N/A	1 minute



3.3 Hudson Introduction

Hudson vs Jenkins



Hudson search 로그인 | 가입 자동 재실행 키기

Hudson

개발자 빌드 기록

빌드 대기 목록
빌드 대기 항목이 없습니다.

빌드 실행 상태

#	상태
1	대기 중
2	대기 중

- NHN Test Automation Framework
- Home : <https://dev.naver.com/projects>

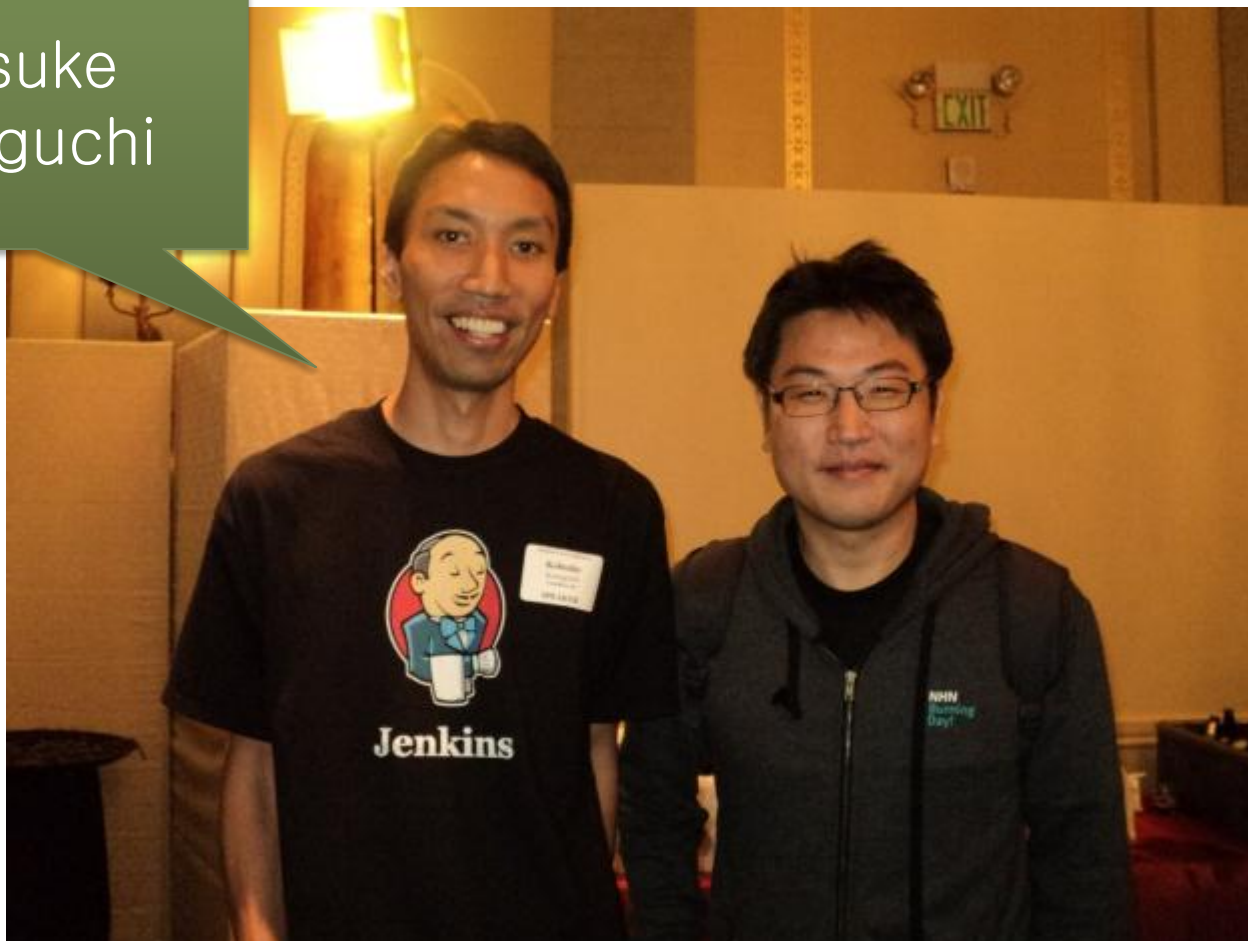
All	Core	Plugin	STAF Services	Tests	
S	W	작업 ↓	최근 성공	최근 실패	최근 소요 시간
		ntaf-acceptance-test	13 hr (#125)	—	34 sec
		ntaf-eclipse-plugin	11 hr (#121)	29 days (#81)	13 sec
		ntaf-hudson-plugin	12 hr (#78)	2 mo 19 days (#5)	30 sec
		ntaf-main-module	13 hr (#792)	2 days 4 hr (#783)	2 min 47 sec



3.3 Hudson Introduction

Jenkins Creator

Kyosuke
Kawaguchi



3.3 Hudson Introduction

Hudson History

❖ From Summer, 2004

• bu

• m

• Ru

The screenshot shows the Hudson web interface in a Mozilla Firefox browser window. The address bar shows the URL `http://kohsuke.sfbay/hudson/`. The interface features a green header with the Hudson logo and a navigation menu on the left. The main content area displays a table of build jobs.

Job	Last Success	Last Failure	Last Duration
java.net cvs keep-alive	25 minutes (#389)	N/A	1 seconds
java.net log processor (daily)	8 hours (#4)	N/A	1 minutes
java.net log processor (weekly)	4 days (#1)	N/A	4 minutes
java.net mail automation	2 hours (#20)	N/A	0 seconds
java.net version check	20 minutes (#319)	6 days (#160)	3 seconds
javadoc	10 hours (#19)	1 days (#18)	11 minutes
jax-fast	30 minutes (#354)	1 days (#335)	9 minutes
jax-fast-2.0	1 hours (#134)	36 minutes (#135)	8 minutes
jaxb-ri	43 minutes (#396)	1 hours (#393)	6 minutes
msv nightly	19 hours (#9)	N/A	15 minutes
stapler	7 days (#39)	9 days (#38)	17 seconds
txw	7 days (#9)	7 days (#6)	44 seconds
xson	10 days (#6)	N/A	1 minutes

At the bottom of the table, there are two buttons: `atom 0.3 for all` and `atom 0.3 for failures`. The status bar at the bottom left shows the text `完了`.

3.3 Hudson Introduction

Hudson History

❖ 2006

Build Queue

Job	Status
hudson	⊘
jaxb-ri	⊘

Build Executor Status

No.	Status
1	Idle
2	Idle
3	Building javanet-maven-repository-daemon #826 ⊘
4	Building jaxb-ri #3181 ⊘
5	Building glassfish #105 ⊘
6	Idle

Job List

Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#16)	9 months (#3)	39 seconds
bsh	6 months (#11)	10 months (#2)	59 seconds
dtd-parser	6 months (#8)	N/A	1 minute
fi	28 days (#586)	1 month (#567)	7 minutes
fi (weekly)	6 days (#53)	13 days (#52)	5 minutes
glassfish	4 hours (#104)	1 day (#88)	1 hour
hudson	4 minutes (#201)	N/A	1 minute
istack-commons	12 days (#19)	16 days (#5)	14 seconds
iapex	3 days (#55)	9 hours (#64)	1 minute
java-ws-xml community discussion updater	4 minutes (#16146)	10 hours (#16125)	1 minute
java.net acl processor	18 hours (#162)	N/A	0 seconds

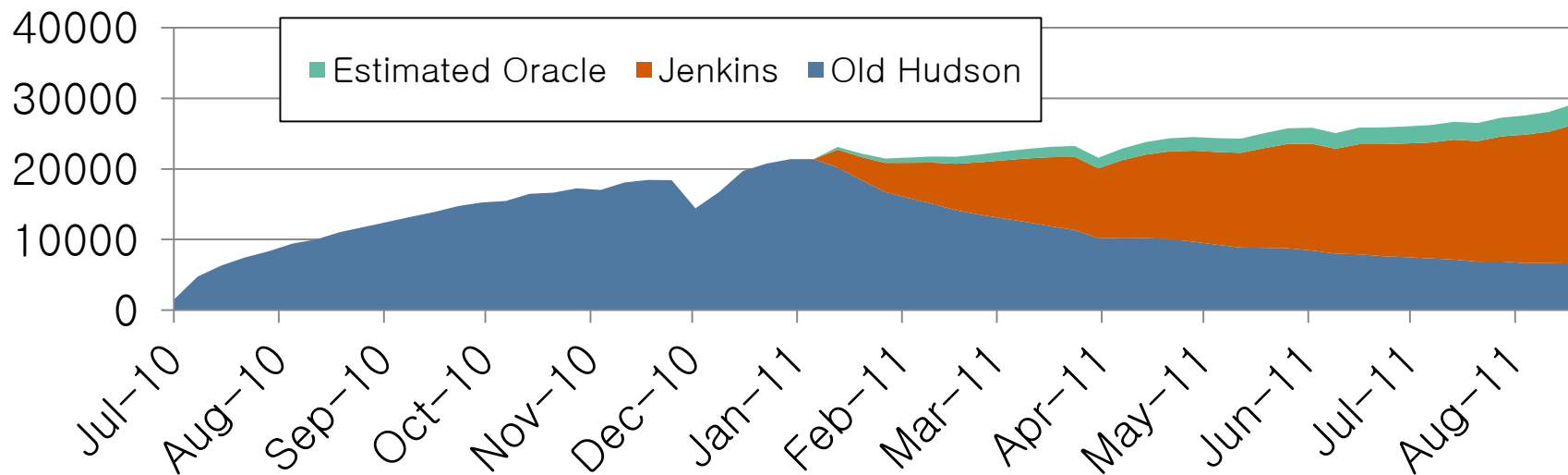


3.3 Hudson Introduction

Jenkins History

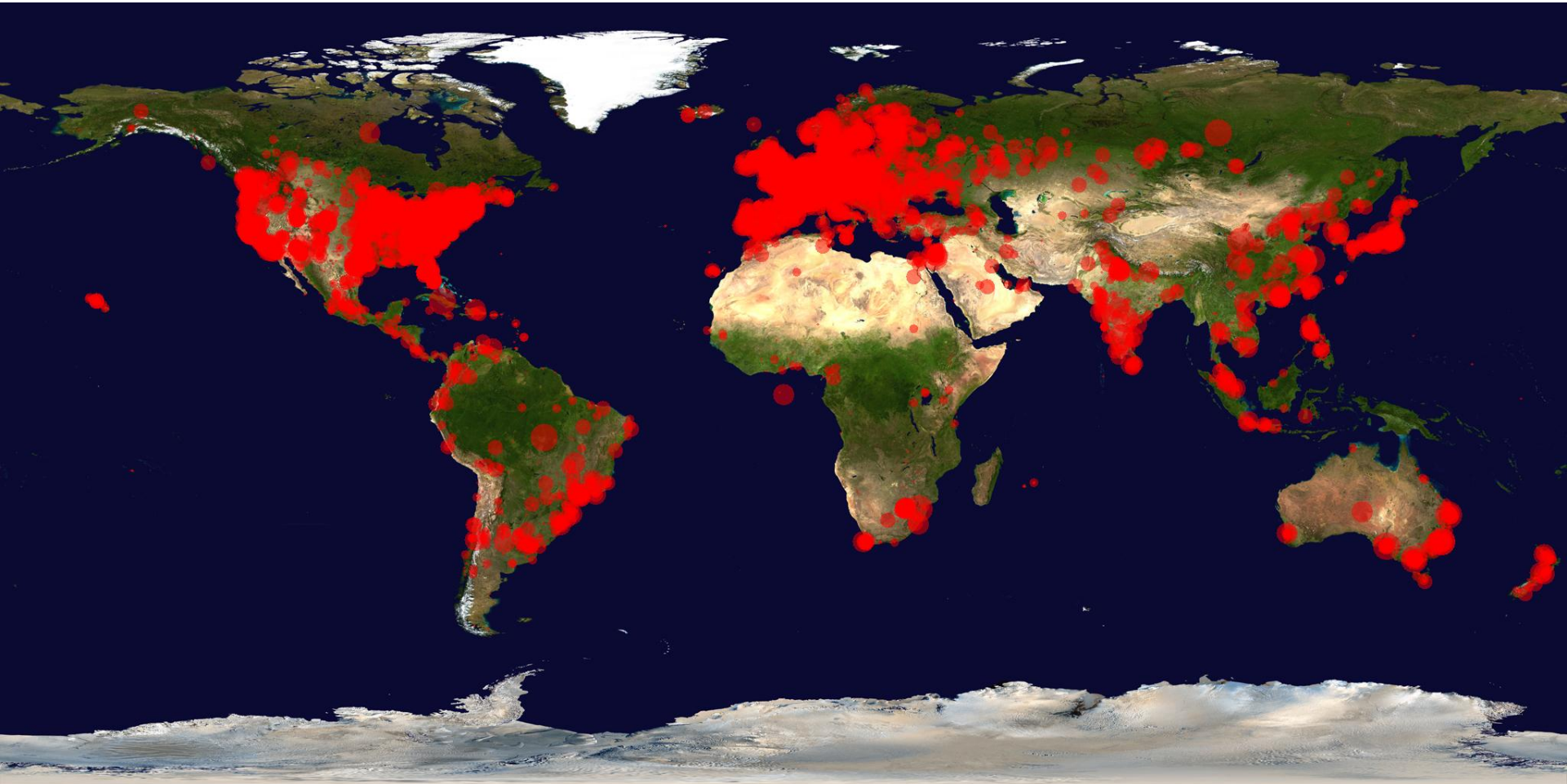
❖ Jan. 2011. Divorce from Oracle

- Oracle: "you do it our way or highway"
- Community chose highway: 214 to 14
 - ✓ That's when we became Jenkins



3.3 Hudson Introduction

Usage All Around the World



3.4 Hudson Features

- Easy installation
- Easy configuration
- Change set support
- Permanent links
- RSS/E-mail/IM Integration
- After-the-fact tagging
- JUnit/TestNG test reporting
- **Distributed builds**
- File fingerprinting
- **Plugin Support**
- **Easy plugin development environment support**

3.4 Hudson Features

Layout - Overall

The screenshot shows the Hudson web interface. At the top, there is a blue header with the 'Hudson' logo and a search bar. Below the header, the breadcrumb 'Hudson > All' is visible. On the left side, there is a sidebar with navigation links: 'New Job', 'Manage Hudson', 'People', 'Build History', and 'Delete View'. The main content area displays a table of jobs. The table has columns for 'S' (Success), 'W' (Warning), 'Job', 'Last Success', 'Last Failure', 'Last Duration', and 'NHN Project'. The first row shows a job named 'Sample1' with a last success of '1 mo 14 days (#32)' and a last failure of '1 mo 15 days (#25)'. Below the table, there are links for 'Icon: S M L' and a legend for RSS feeds: 'for all', 'for failures', and 'for just latest builds'. On the left side of the main content, there are two panels: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two executors in 'Idle' status).

S	W	Job ↓	Last Success	Last Failure	Last Duration	NHN Project
		Sample1	1 mo 14 days (#32)	1 mo 15 days (#25)	1 min 4 sec	S 1 샘플프로젝트

#	Status
1	Idle
2	Idle

3.4 Hudson Features

Layout – Project

Hudson

Hudson » [ntaf-main-module](#) search ? 로그인 | 가입

자동 재실행 커기

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Coverage Report](#)

[N'SIQ Collector](#)

[Subversion Polling Log](#)

Project ntaf-main-module

- [NTAF Main Module](#)
- FitNesse version : 20090828

[Coverage Report](#)

[N'SIQ Collector](#)

[Last Successful Artifacts](#)

- [ntaf-core-20100731-SNAPSHOT.jar](#)
- [ntaf-fitlibrary-20100731-SNAPSHOT.jar](#)
- [ntaf-fitness-20100731-SNAPSHOT.jar](#)

Build History (trend)

#	Time	Size
#792	2010. 8. 25 오전 2:01:32	8MB
#791	2010. 8. 24 오전 2:01:32	8MB
#790	2010. 8. 23 오후 5:33:38	8MB
#789	2010. 8. 23 오후 5:29:38	8MB
#788	2010. 8. 23 오후 5:07:25	8MB

Disk Usage: Workspace 157MB, Builds 3GB

Disk Usage Trend

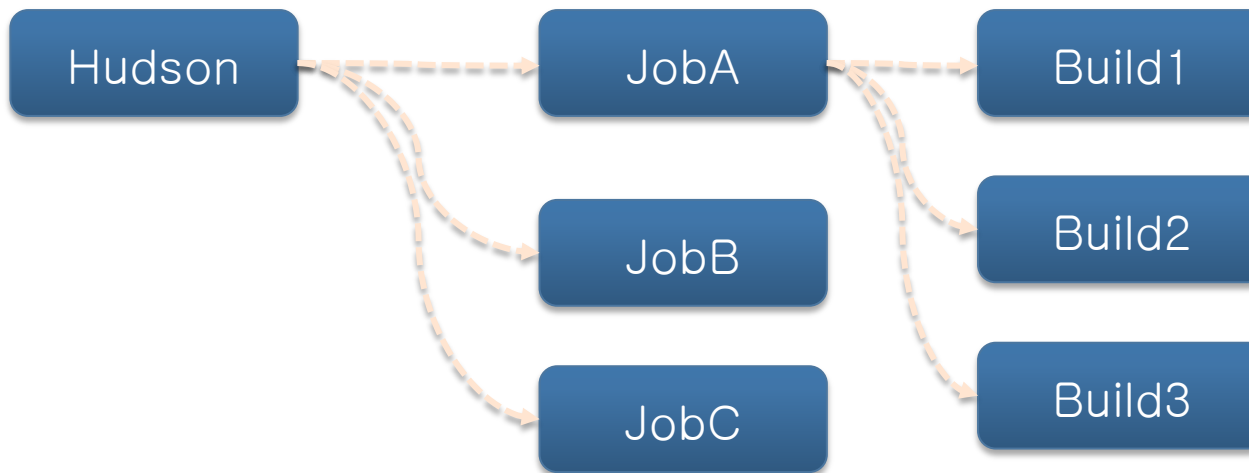
Test Result Trend

33 / Staged Build for Java

3.4 Hudson Features

Structural Architecture

❖ **Conceptual == Physical layer**



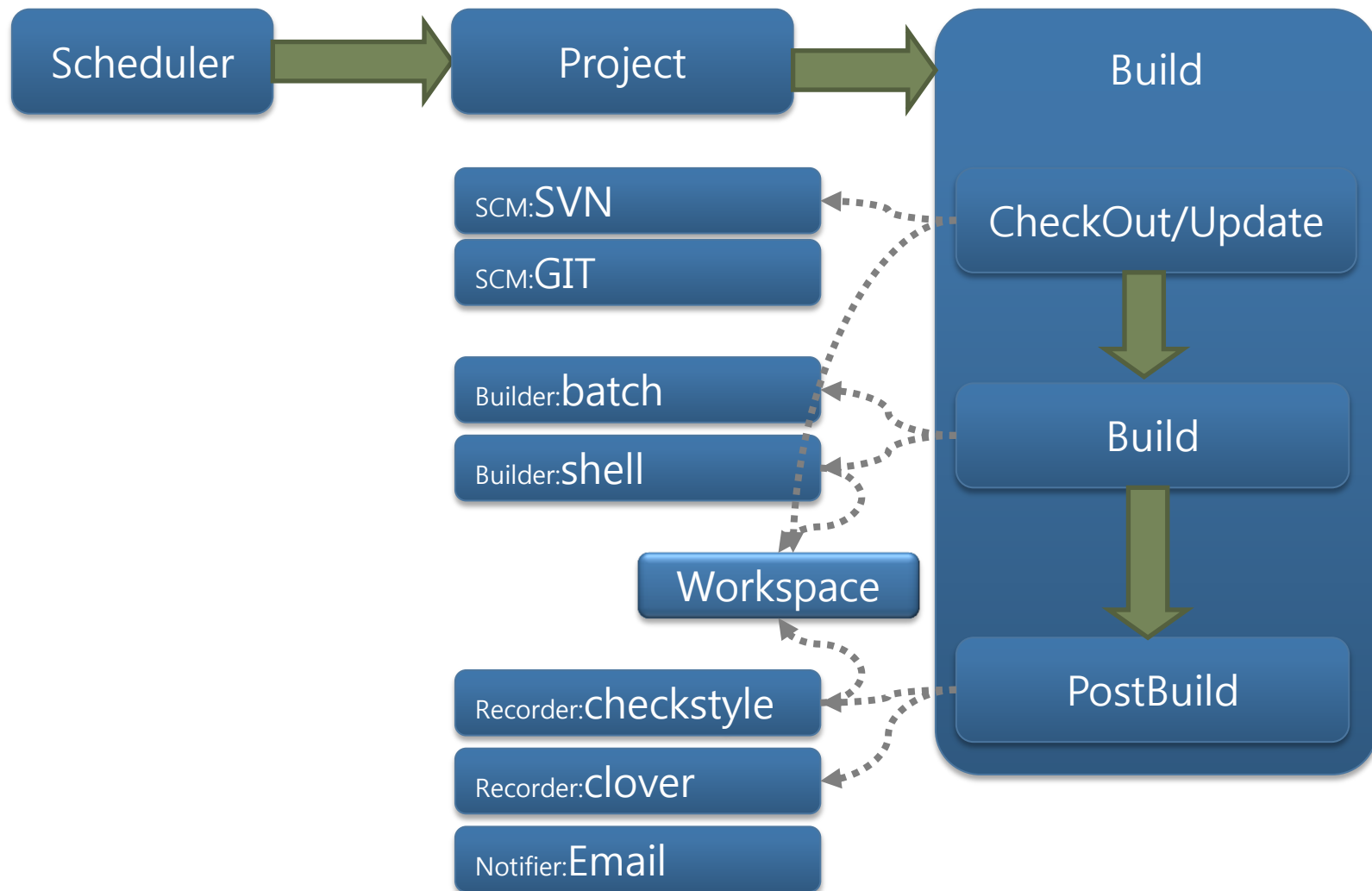
❖ **Each concept is mapped to Object.**

❖ **Each object keeps its persistency with XML**


```
Hudson hudson = Hudson.getInstance();  
((AbstractProject)hudson.getJob("jobA")).getBuildByNumber(3);
```

3.4 Hudson Features

Runtime Architecture



3.5 Hudson plugins

- Source code management
 - Build triggers
 - Build tools
 - Build wrappers
 - Build notifiers
 - Slave launchers and controllers
 - Build reports
 - Artifact uploaders
 - Other post-build actions
 - External site/tool integrations...
- 
- ClearCase Plugin
 - File System SCM Plugin
 - Mercurial Plugin
 - Perforce Plugin
 - Harvest Plugin
 - Team Foundation Server Plugin
 - Template Project Plugin
 - Accurev Plugin
 - CVS Plugin
 - ...

More than 400 plugins



* For more details refer

<http://wiki.hudson-ci.org/display/HUDSON/Plugins> Build for Java

3.5 Hudson plugin

Hudson plugin developed By NHN

- Hudson QD Plugin : Sync collected metric with Quality Dashboard
- Hudson N'SIQ Collector Plugin : Show LOC, Complexity collected by N'SIQCollector
- Hudson Klocwork Plugin : Show static analysis result analyzed by Klocwork
- Hudson NHN Auth Plugin : Make hudson authenticated using MyNEXT ID
- Hudson CovComplPlot Plugin : Show Coverage/Complexity relation graph
- Hudson SimpleUpdateSite Plugin : NHN Custom Hudson Plugin updatesite

3.6 Hudson and QualityDashboard

Manager Needs

I'm Steve Jobs.
I'd like to see all apple
projects status in a big
picture.

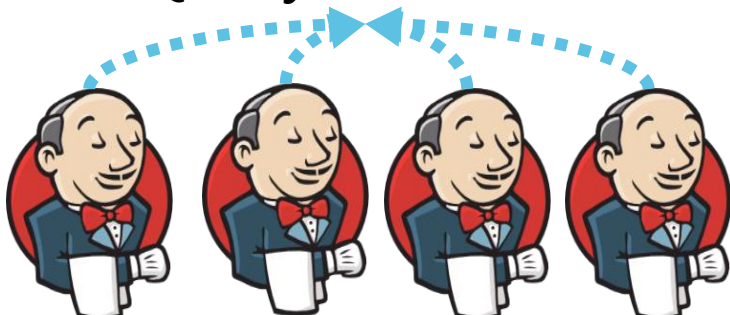


3.6 Hudson and QualityDashboard

Quality Governance

		● 45	● 51	● 18	● 10	프로젝트
프로젝트명	조회기준일	Code Quality	코딩 표준 준수 수율 (%)	Code Coverage (Branch) (%)	Code Coverage (Statement) (%)	
		?	?	?	?	
팀 [동영상서비스] 뮤직	2011.08.30	진행률 0,00 ●			측정안함	
팀 통합DB관리시스템	2011.08.22	진행률 52,63 ●	팀 100 ●	측정안함	42.1 ●	
팀 Coverage4Batis	2011.08.22	진행률 55,32 ●	팀 100 ●	팀 85 ●	측정안함	
팀 Markupstyle	2011.08.22	진행률 66,32 ●	팀 100 ●	팀 82.1 ●	측정안함	
팀 통합DB관리시스템	2011.08.22	52.63 ●	팀 100 ●	측정안함	42.1 ●	
팀 [Design Studio] 디자인	2011.08.22	진행률 36,41 ●	팀 91.47 ●	팀 0.33 ●	측정안함	

Quality Dashboard



3.6 Hudson and QualityDashboard

Quality Governance

- ❖ Quality Dashboard (<http://nsiq.nhncorp.com/>)
- ❖ Each Hudson Project report its metrics to QD by QD plugin

The screenshot displays the NSIQ Quality Dashboard interface. At the top, there are navigation links for 'HOME', '프로젝트관리', '프로젝트분석', 'N'SIQ 품질지표', and '라이브러리관리'. The date '2010-08-25 (수)' is shown in the top right corner. The main content area is titled 'Quality Dashboard' and contains a table with the following data:

단계	N'SIQ 품질지표 전체	측정대상		품질기준 만족		품질지표 측정 실행	
		포함 (=A+B)	미포함	만족 (A)	불만족 (B)	실행 (A)	미실행 (B)
분석	1	0	1	0	0	0	0
설계	1	0	1	0	0	0	0
구현	17	6	11	2	4	5	1
QA테스트	17	10	7	0	10	0	10
운영	1	1	0	0	1	0	1
Total	37	17	20	2	15	5	12

Below this table, there is a section for '품질지표' (Quality Indicators) with a sub-table:

품질지표명	품질기준	품질지표	측정일자	측정항목	개
구현 단계					
C-FCR	기능 구현율	100 (%)		(A) 구현된 기능 명세 수 (B) 구현하기로 합의된 기능 명세 수	17개 19개
C-CSCR	코딩 표준 준수율	100 (%)	2010.08.24	(A) 표준 준수한 파일(코딩 컨벤션 항목) 수 (B) 적용 대상 파일(코딩 컨벤션 항목) 수	17개 19개
C-COV	Code Coverage (Branch)	52.3 (%)	2010.08.24	(A) 테스트 수행된 Branch 수	250개 478개

4. CI Lab

1-1-1

4.1 Install Hudson

Download

❖ All Instructions are available in

- <http://dev.naver.com/projects/hudsonedu/wiki/Java실습스크립트>

❖ Hudson Download

- <http://hudson-ci.org/downloads/war/>
- Download 1.395.1 version

❖ Tomcat Download

- <http://tomcat.apache.org/> → Tomcat 6.0 → Binary Distributions → Core zip link
- Download and unzip

4.1 Install Hudson

Startup Hudson

- Put the downloaded hudson.war into {TOMCAT_HOME}/webapps
- Run \${TOMCAT_HOME}/bin/startup.bat(win) or startup.sh (linux)
- Open <http://localhost:8080/hudson> and see the following page



The screenshot shows the Hudson web interface. At the top, there is a blue header with the word "Hudson" and a search bar. Below the header, there are several navigation links: "새 작업" (New Job), "Hudson 관리" (Manage Hudson), "개발자" (Developers), and "빌드 기록" (Build History). A central message reads: "Hudson에 오신 것을 환영합니다. 시작하려면 새 작업을 만들어 주시기 바랍니다." (Welcome to Hudson. To get started, please create a new job.) To the right, there are links for "자동 재실행 커기" (Automatic Retry) and "소개 내용 입력" (Enter Introduction). Below the navigation links, there are two sections: "빌드 대기 목록" (Build Queue) and "빌드 실행 상태" (Build Execution Status). The "빌드 대기 목록" section shows "빌드 대기 항목이 없습니다." (No build items in queue). The "빌드 실행 상태" section contains a table with two columns: "#", "상태", and "이름".

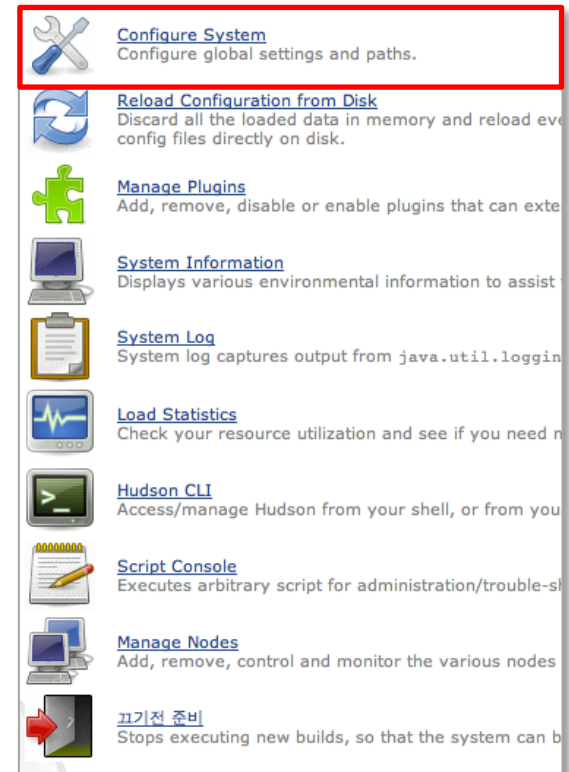
#	상태	이름
1	대기 중	
2	대기 중	

4.1 Install Hudson

Configure System menu

❖ “Manage Hudson” → “Configure System” : Hudson Global Configuration

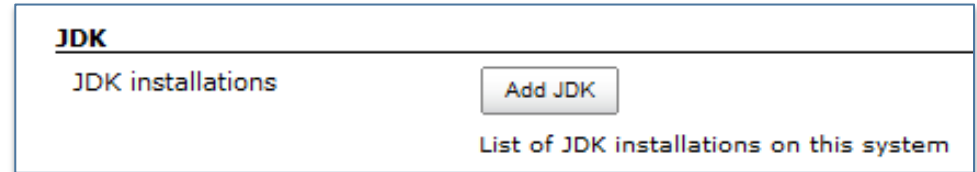
- Setup general Hudson env configuration
- Setup Build Tool option (Maven, JDK, Ant, Shell) and Configure the plugins' global behavior
 - ✓ Automatically installable (Ant, Maven, JDK only)



4.1 Install Hudson

Setup JDK

- Click Add JDK



❖ If JDK is already installed

1. Uncheck Install automatically
2. Put your own JDK label(e.g : jdk 1.6.0_25) in the "name" field
3. Put JDK path in JAVA_HOME(e.g : C:\Program Files\Java\jdk1.6.0_25)

❖ If JDK is not installed

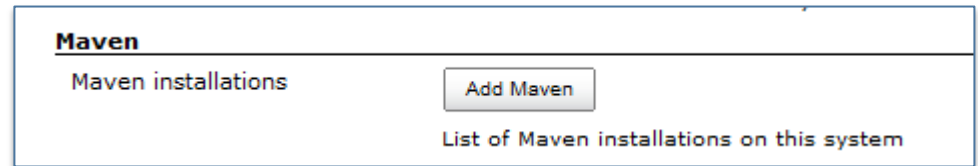
1. Put your own JDK label(e.g : jdk 1.6.0_25) in the "name" field
2. Choose JDK version being installed

❖ Click "Save" button in the bottom

4.1 Install Hudson

Setup Maven

- Click Add Maven



❖ If Maven is already installed

1. Uncheck Install automatically
2. Put your own Maven label (e.g. Maven 2.2.1) in the "name" field
3. Put Maven path in MAVEN_HOME(e.g. C:\dev\apache-maven-2.2.1)

❖ If Maven is not installed

1. Put your own Maven label(e.g. Maven 2.2.1) in the "name" field
2. Put Maven path in MAVEN_HOME(e.g. C:\dev\apache-maven-2.2.1)

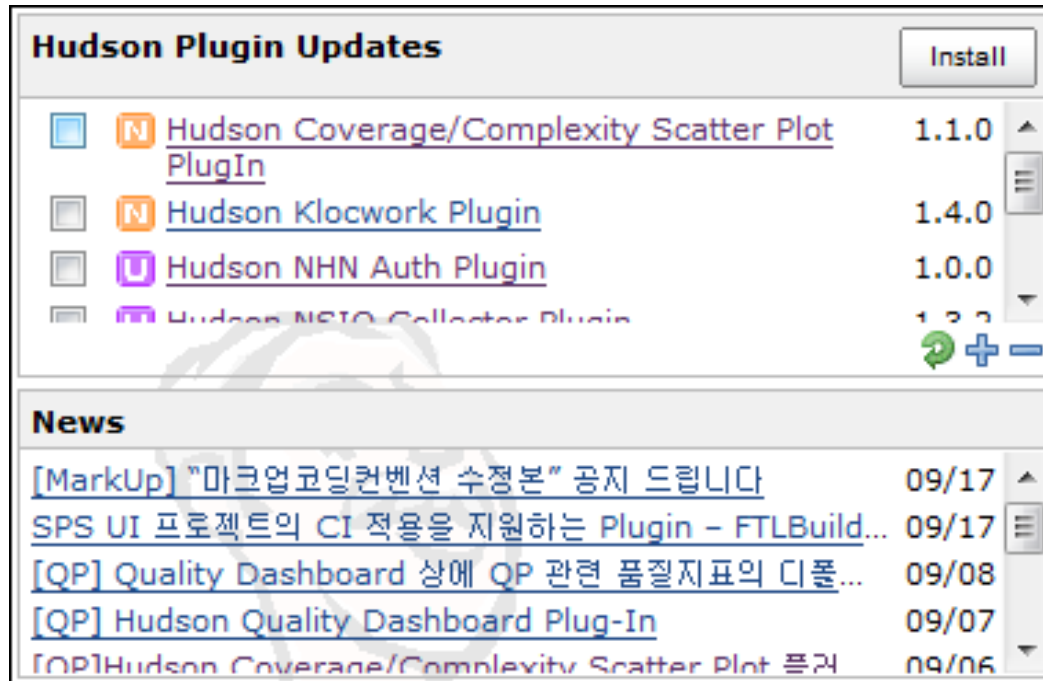
❖ Click "Save" button in the bottom

❖ You can setup Ant same way as Maven





4.1 Install Hudson

SimpleUpdateSite Plugin

- Enable each Hudson to connect Custom Plugin UpdateSite



The screenshot shows the 'Hudson Plugin Updates' interface. At the top right is an 'Install' button. Below it is a table of available plugins:

Plugin Name	Version
<input type="checkbox"/>  Hudson Coverage/Complexity Scatter Plot PlugIn	1.1.0
<input type="checkbox"/>  Hudson Klocwork Plugin	1.4.0
<input type="checkbox"/>  Hudson NHN Auth Plugin	1.0.0
<input type="checkbox"/>  Hudson NSIO Collector Plugin	1.2.2

Below the plugin list is a 'News' section with several entries:

News Title	Date
[MarkUp] "마크업코딩컨벤션 수정본" 공지 드립니다	09/17
SPS UI 프로젝트의 CI 적용을 지원하는 Plugin - FTLBuild...	09/17
[QP] Quality Dashboard 상에 QP 관련 품질지표의 디폴...	09/08
[QP] Hudson Quality Dashboard Plug-In	09/07
[QP]Hudson Coverage/Complexity Scatter Plot 플러그...	09/06

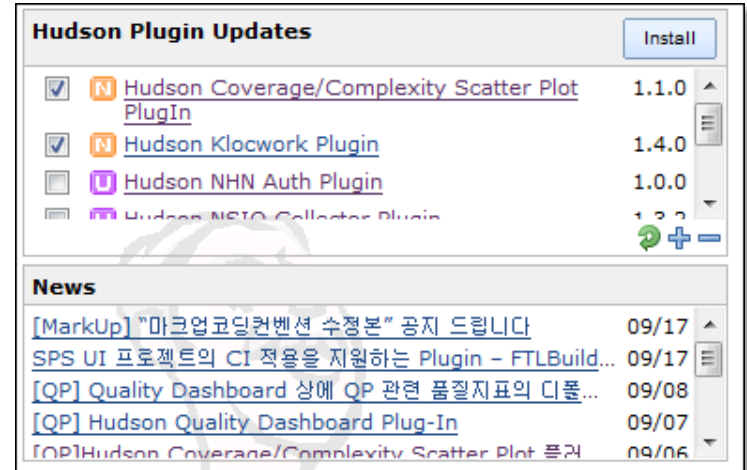
- If you're using Jenkins, you can download it from Jenking update site.

4.1 Install Hudson

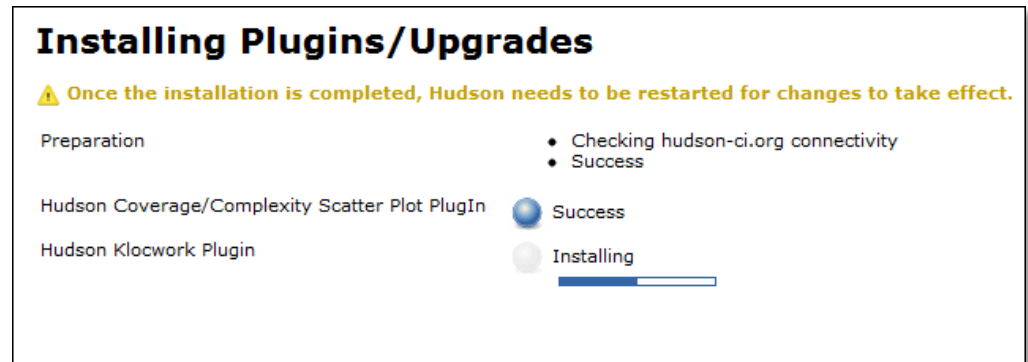
SimpleUpdateSite Plugin

- If you have Admin permission, show following.

1. Select plugins to be installed
 - ✓ N : New plugin
 - ✓ U : Updatable plugin
 - ✓ I : Already installed plugin



2. Click Install button



4.2 Lab Introduction

Step

- Create Hudson Project
- Setup Sample Project Checkout
- Setup Unit Test
- Setup QP Tools

4.2 Lab Introduction

Sample Project

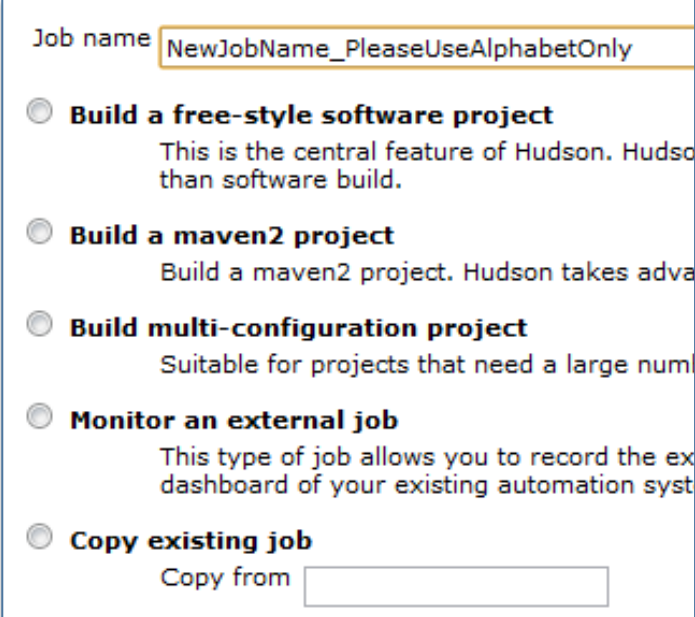
❖ **Sample project in dev.naver.com**

- <https://dev.naver.com/svn/hudsonedu/trunk>

4.3 Configure Hudson Project

Register new Hudson Project (Job)

- ❖ Click “New Job” to register new Hudson Project
- ❖ Job Name
 - Should be [a-zA-Z][a-zA-Z0-9_]*
 - Why??
 - ✓ It will be used as the folder name in which code are checked out and built.
- ❖ Select “Build a free-style software project”



Job name

Build a free-style software project
This is the central feature of Hudson. Hudson takes advantage of software build.

Build a maven2 project
Build a maven2 project. Hudson takes advantage of Maven.

Build multi-configuration project
Suitable for projects that need a large number of configurations.

Monitor an external job
This type of job allows you to record the execution of an external automation system on your dashboard.

Copy existing job
Copy from

Create Hudson Project with “edu_XX”

4.3 Configure Hudson Project

NHNProject Plugin

❖ NHNProject Plugin :

- Provide and display properties which represent project characteristics

The screenshot displays the Hudson web interface for a project named 'TEST22'. The interface is organized into several sections:

- Navigation Links:** A vertical list of links on the left side, each with an icon: 'Back to Dashboard' (green arrow), 'Status' (magnifying glass), 'Changes' (notepad), 'Workspace' (folder), 'Build Now' (play button), 'Delete Project' (red prohibition sign), 'Configure' (wrench), and 'HudsonPlugin TEST22' (magnifying glass).
- Project Information:** The title 'Project TEST22' is prominently displayed. Below it, there are icons for 'HudsonPlugin TEST22' (magnifying glass), 'Workspace' (folder), and 'Recent Changes' (notepad).
- Build History:** A section titled 'Build History' with a '(trend)' link. It includes two RSS feed icons: 'for all' and 'for failures'.
- Permalinks:** A section titled 'Permalinks' is located at the bottom right of the main content area.

4.3 Configure Hudson Project

NHNProject Plugin

- Installable from SimpleUpdateSite

❖ How to configure NHNProject plugin

<input checked="" type="checkbox"/>	Assign NHN Project
NHN Project Name	<input type="text" value="DevCenter"/>
	Please refer here for the NHN proj
Project Type	<input type="text" value="Sustaining"/>
Build Type	<input type="text" value="Integration Build"/>

Assign your project name / project type / build type

4.3 Configure Hudson Project

Src Repo

❖ Specify source code repository

Source Code Management

None
 Subversion

Modules

Repository URL ?

Local module directory (optional) ?

Use update
If checked, Hudson will use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

Revert
If checked, Hudson will do 'svn revert' before doing 'svn update'. This slows it down, but will prevent files being modified from build to build.

Repository browser ?

Specify your SVN repo

4.3 Configure Hudson Project

Build Trigger

❖ Build Triggers : Set up the Hudson build start event

- Build after other projects are build
- Build periodically
- Poll SCM : Execute build when detecting source code changes(commit)
- Cron expression

Min Hour Date Month Week

E.g)

Every minute → * * * * *

Every 5 min → */5 * * * *

4.3 Configure Hudson Project

Add Builders

- Build : Execute build command for various executor
 - ✓ Execute shell(Linux)
 - ✓ Invoke top-level Maven targets
 - Put maven goals necessary to build maven project

▪ E.g)

Invoke top-level Maven targets	
Maven Version	apache-maven-2.0.9
Goals	clean compile





- ✓ Execute Windows batch command (Windows)
- ✓ Invoke Ant : Execute ant target on ant build script
- Post-build Actions : Define tasks after build
 - ✓ Mostly import generated doc(e.g: xml) to Hudson for display

Specify maven goals (clean compile)

4.3 Configure Hudson Project

Build Now

❖ Not only scheduled build But also Direct build

- Click "Build Now" on left panel
- Show up new build with sequence in build history
- Can check build status in Build History
 - ✓ Success Build 
 - ✓ Unstable Build 
 - ✓ Failed Build 
 - ✓ Canceled Build 

Click Build Now

4.3 Configure Hudson Project

Build Log

❖ Hudson shows build log almost realtime.

- Click Build in Build panel and Click Console Output

Console Output

Skipping 4,043 KB.. [Full Log](#)

```
1-12-14 16:12:46 [DEBUG](MagicMatcher.java:556) testShort()  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:582) testShort(): testing '424c' against 'ffffffffffffffa'  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:79) getMatch()  
2011-12-14 16:12:46 [DEBUG](Magic.java:340) getMagicMatch(File): trying to match: GIF image data  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:161) test(File)  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:168) test(File): testing 'test.bmp' for 'GIF image data'  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:170) test(File):  
=== BEGIN MATCH INFO ==  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:171)  
mime type: image/gif  
description: GIF image data  
extension:  
offset: 0  
length: 0  
test: GIF8  
type: string  
comparator: =  
bitmask: 4294967295  
2011-12-14 16:12:46 [DEBUG](MagicMatcher.java:172) test(File):  
=== END MATCH INFO ===
```

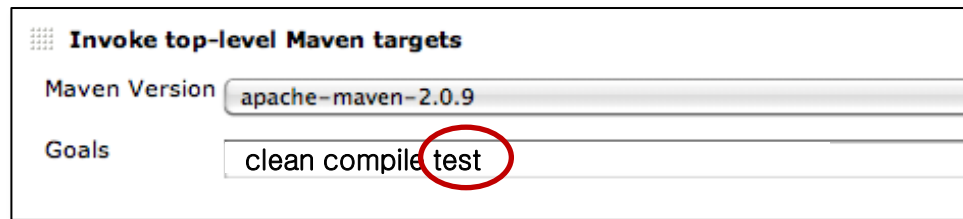
See console output

4.4 Enable Unit Test

Setup Unit Test Execution

❖ supports JUnit in nature for test result display

- Click configure button in left panel
- Add test goal in "Invoke top-level Maven targets"

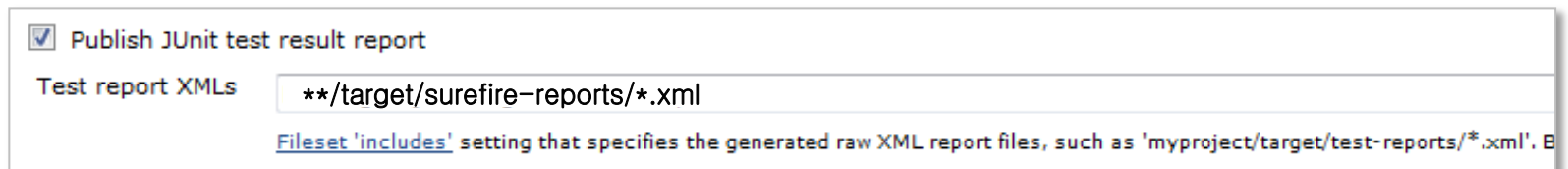


Invoke top-level Maven targets

Maven Version

Goals

- Click "Publish JUnit test result report" in Post-build Actions
- Input result result xml file location)



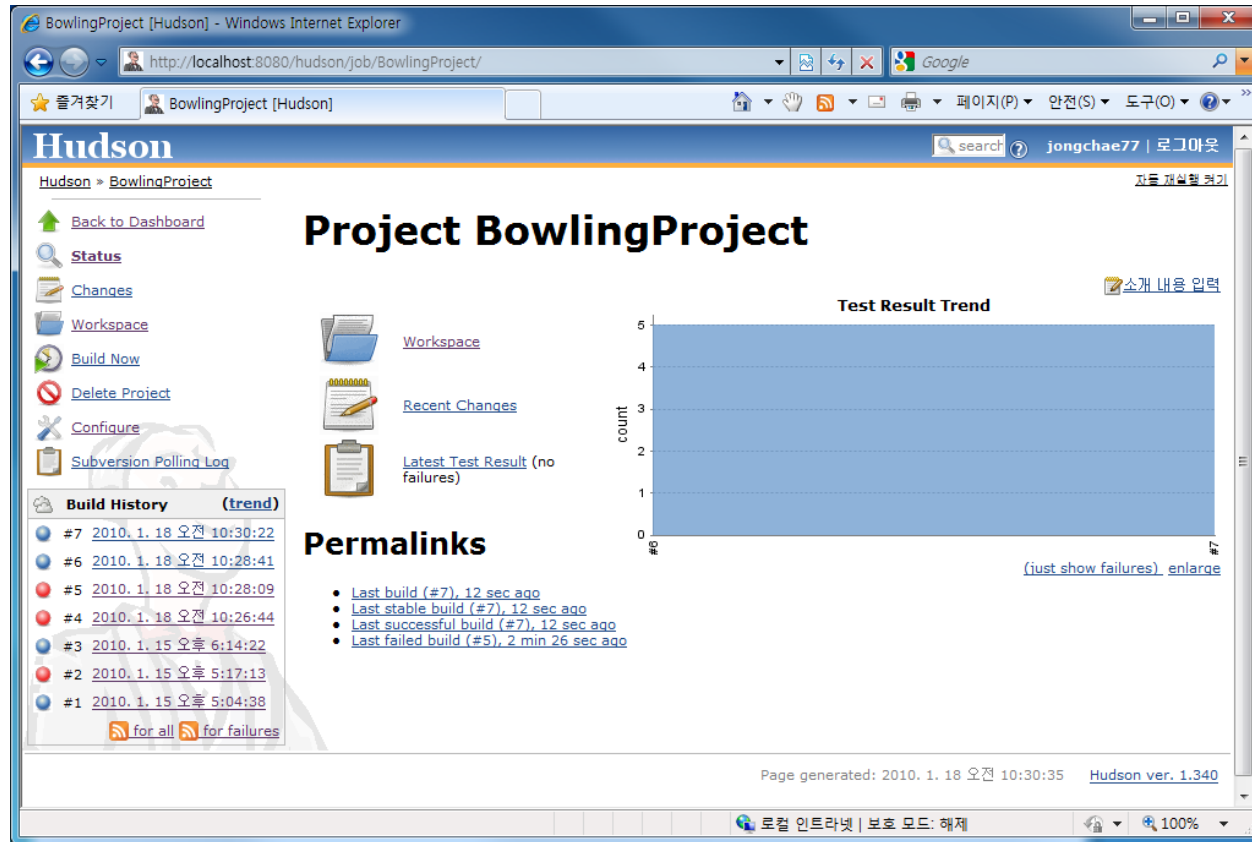
Publish JUnit test result report

Test report XMLs

[Fileset 'includes'](#) setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. B

4.4 Enable Unit Test

Unit Test Result

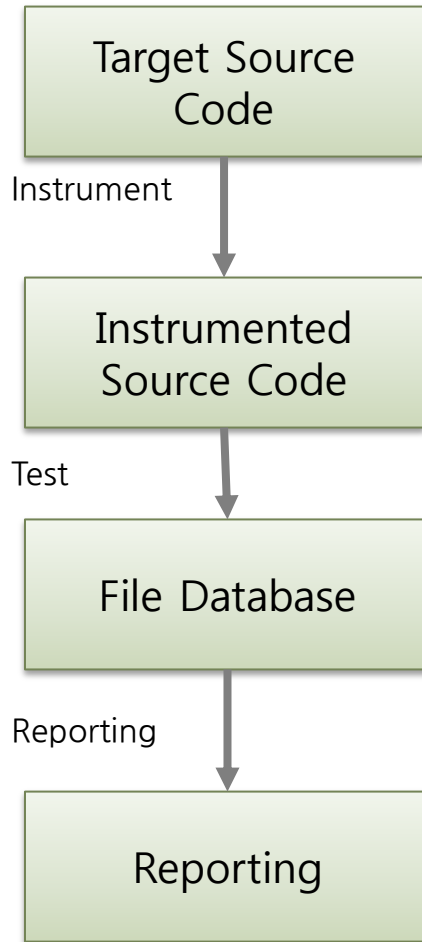


Specify maven goals and enable “publish JUnit test result”



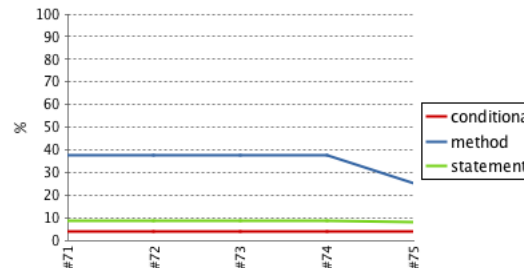
4.5 Code Coverage with Clover

How Code Coverage works



```
public class HelloWorld {  
    public static String nullPossibleString = null;  
  
    public static void main(final String[] args) {  
        System.out.println(helloWorld());  
  
        if (nullPossibleString.equals(null)) {  
            System.out.println("nullPossibleString is null.");  
        }  
    }  
  
    public static String helloWorld() {  
        return hello() + " " + world();  
    }  
}
```

```
public class HelloWorld {public static class __CLR3_0_24n4ngbimreni{publi  
    public static String nullPossibleString = null;  
  
    public static void main(final String[] args) {try{__CLR3_0_24n4ngbimr  
        __CLR3_0_24n4ngbimreni.R.inc(168);System.out.println(helloWorld())  
  
        __CLR3_0_24n4ngbimreni.R.inc(169);if (((nullPossibleString.equal  
        __CLR3_0_24n4ngbimreni.R.inc(172);System.out.println("nullPos  
    }  
    }finally{__CLR3_0_24n4ngbimreni.R.flushNeeded();}}  
  
    public static String helloWorld() {try{__CLR3_0_24n4ngbimreni.R.inc(1  
        __CLR3_0_24n4ngbimreni.R.inc(174);return hello() + " " + world();  
    }finally{__CLR3_0_24n4ngbimreni.R.flushNeeded();}}  
}
```



4.5 Code Coverage with Clover

Various Code Coverage

- Function(method) coverage
 - Statement(line) coverage
 - Decision(branch) coverage
 - Condition coverage – coverage for boolean sub expression
 - Condition / decision coverage – Decision + Condition Coverage
-
- ❖ **Mandatory to measure Statement or Branch(Decision) coverage in NHN**
 - ❖ **Method > Branch > Statement >= Condition(?) > Condition / decision**

http://en.wikipedia.org/wiki/Code_coverage



4.5 Code Coverage with Clover

Question

```
public int foo(int x, int y) {  
    int z = y;  
  
    if ((x > 5) && (y > 0)) {  
        z = x;  
    }  
  
    return x * z;  
}
```

```
assertEquals(49, foo.foo(7, 1));
```

- Branch coverage?
- Conditional coverage?
- Statement coverage?

4.5 Code Coverage with Clover

Modify maven build script

❖ Add clover plugin maven repo in pom.xml

```
<pluginRepository>
  <id>atlassian-m2-repository</id>
  <name>Atlassian Maven 2.x Repository</name>
  <url>http://repository.atlassian.com/maven2</url>
</pluginRepository>
```

❖ Caution!!

- Clover 3.0.2 is not compatible with Maven 3.X
- Clover 3.0.4 has lots of bugs
- Do not install project instrumented by clover into .m2 folder

4.5 Code Coverage with Clover

Modify maven build script

❖ Add Clover build plugin in pom.xml

- Put following lines into pom.xml

```
<build>
  <plugins>
    <plugin>
      <groupId>com.atlassian.maven.plugins</groupId>
      <artifactId>maven-clover2-plugin</artifactId>
      <version>3.0.2</version>
      <configuration>
        <licenseLocation>../../clover.license</licenseLocation>
        <generateHtml>true</generateHtml>
        <generateXml>true</generateXml>
        <generatePdf>false</generatePdf>
      </configuration>
    </plugin>
  </plugins>
</build>
```

4.5 Code Coverage with Clover

Modify maven build script

❖ Add Clover reporting plugin in pom.xml

```
<reporting>
  <plugins>
    <plugin>
      <groupId>com.atlassian.maven.plugins</groupId>
      <artifactId>maven-clover2-plugin</artifactId>
      <version>3.0.2</version>
      <configuration>
        <includesTestSourceRoots>true</includesTestSourceRoots>
        <licenseLocation>../../clover.license</licenseLocation>
        <encoding>UTF-8</encoding>
        <jdk>1.5</jdk>
      </configuration>
    </plugin>
  </plugins>
</reporting>
```

Modify your pom.xml

4.5 Code Coverage with Clover

Maven goals

❖ Clover2-Maven-Plugin Goal

Goal	Description
clover2:clean	Initialize Clover Database
clover2:setup	Initialize clover instrumentation feature
test	Run JUnit Test
clover2:clover	Make coverage report under ./target/site/clover folder

4.5 Code Coverage with Clover

Hudson Clover Plugin

❖ Install Hudson Clover Plugin

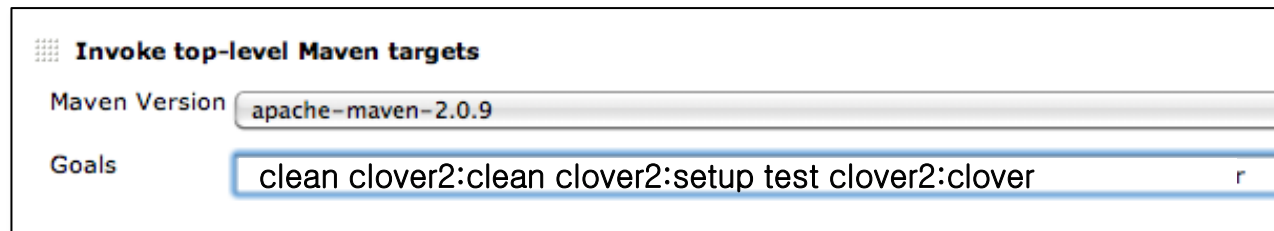
1. Manage Hudson → Manage Plugins → Available Tab
2. Select Clover Plugin and Click install button
3. Restart Tomcat

4.5 Code Coverage with Clover

Hudson Clover Plugin

❖ Setup Hudson Clover Plugin per Project

- Add Clover Goal into maven project

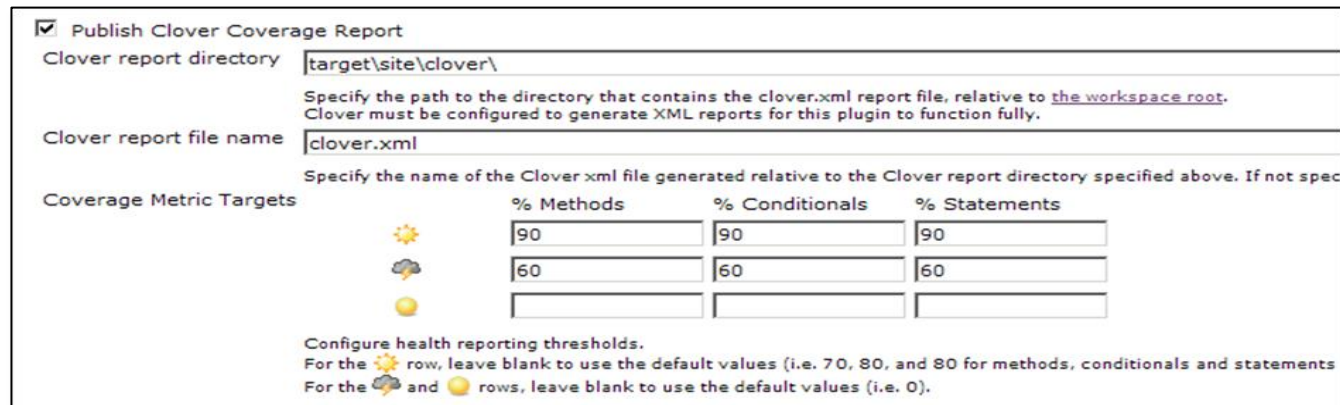


Invoke top-level Maven targets

Maven Version

Goals

- Enable "Publish Clover Coverage Report" and setup like following









Publish Clover Coverage Report

Clover report directory
Specify the path to the directory that contains the clover.xml report file, relative to [the workspace root](#). Clover must be configured to generate XML reports for this plugin to function fully.

Clover report file name
Specify the name of the Clover xml file generated relative to the Clover report directory specified above. If not specified, the default is clover.xml

Coverage Metric Targets

	% Methods	% Conditionals	% Statements
	<input type="text" value="90"/>	<input type="text" value="90"/>	<input type="text" value="90"/>
	<input type="text" value="60"/>	<input type="text" value="60"/>	<input type="text" value="60"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>

Configure health reporting thresholds.
For the  row, leave blank to use the default values (i.e. 70, 80, and 80 for methods, conditionals and statements)
For the  and  rows, leave blank to use the default values (i.e. 0).

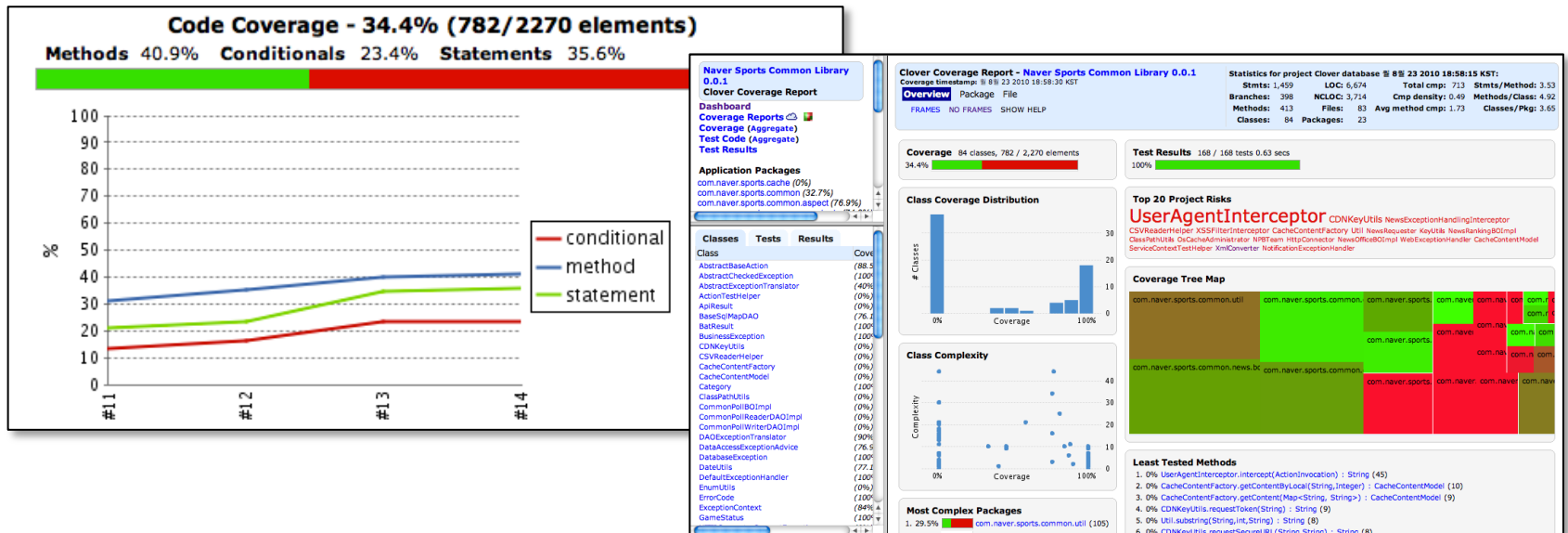
Add clover goal and enable clover reports

4.5 Code Coverage with Clover

Hudson Clover Plugin

Configure	Description
Clover report directory	Specify Clover Report(xml) location /target/site/clover in default
Clover report file name	Specify XML report file name clover.xml in default

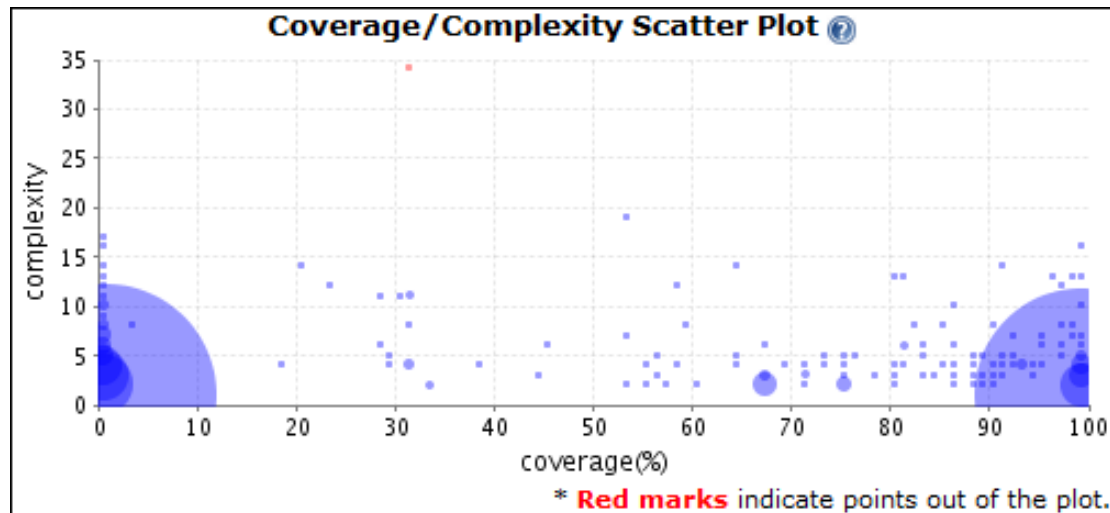
- After configuration, run "Build Now"
 - More than 2 builds with Clover, you'll see following



4.6 Coverage / Complexity Graph with CovComplPlot

Overview

- ❖ Hudson plugin which shows Coverage / Complexity relation graph which help the developer to choose the test necessary code.
- ❖ more complexity, more test!!



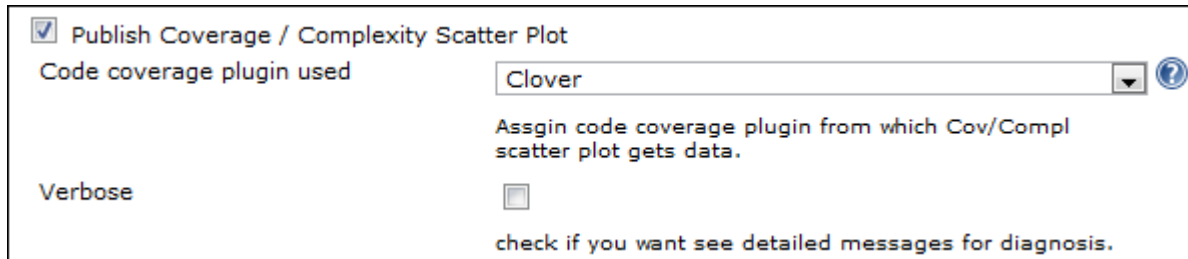
4.6 Coverage / Complexity Graph with CovComplPlot

Install and setup Hudson CovComplPlot Plugin


- Installable from Official Jenkins Update Site

❖ How to setup CovComplPlot Plugin

- Enable “Publish Coverage / complexity Scatter Plot” and choose coverage report type



Publish Coverage / Complexity Scatter Plot

Code coverage plugin used 

Assign code coverage plugin from which Cov/Compl scatter plot gets data.

Verbose

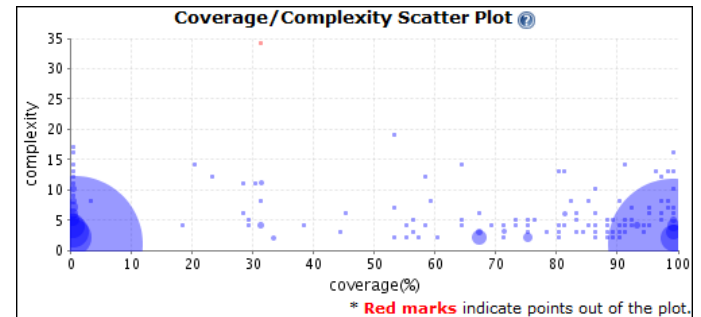
check if you want see detailed messages for diagnosis.

- Click “Build Now”

4.6 Coverage / Complexity Graph with CovCompPlot

Result

❖ Graph



❖ Click each grid to see what methods are located in the each grid.

Coverage / Complexity Scatter Plot

Method list(2055) in the range of coverage (90%~100%) and complexity (0~4)

Method	Complexity	Coverage(%)	Size	Covered
getPercentileValue(List<Double>,Double) : Double	4	94.74	19	18
getStatusCount(List<JiraIssue>,boolean) : List<JiraStatistic>	4	93.75	16	15
getStrategyResultCount(List<JiraIssue>,boolean) : List<JiraStatistic>	4	100	16	16
setAllBtsComponent(List<BtsComponent>) : void	4	93.33	15	14
setAllBtsIssueType(List<BtsIssueType>) : void	4	93.33	15	14
setAllBtsStatus(List<BtsStatus>) : void	4	93.33	15	14
setAllBtsVersion(List<BtsVersion>) : void	4	93.33	15	14
setAllProjectMember(List<ProjectMember>) : void	4	93.33	15	14
setAllCodeQualityGrade(List<CodeQualityGrade>) : void	4	93.33	15	14
getMemberInfo() : Map<String, String>	4	92.31	13	12
updateProjectQualityMeasure() : String	4	100	12	12
createSysLog(Class<?>,String,Object) : SysLog	4	90.91	11	10
deleteProjectMember() : String	4	100	11	11
updateProjectMilestone() : String	4	100	11	11
deleteProjectQualityMeasure() : String	4	90	10	9
insertProjectMember() : String	4	100	9	9
deleteProjectMilestone() : String	4	100	9	9
insertProjectMemberMulti(List<ProjectMember>) : int	4	100	9	9
updateProjectComment() : String	4	100	9	9
deleteProjectComment() : String	4	100	9	9

```

71 5
72 5
73 27965
74 23375
75 23375
76 5
77 23375
78 23375
79 5
80 4590
81 4590
82 4590
83 5
84 5
85 0
86 0
87 5
88 5
89 5
90 5
91 5
92 5
93 5
94 0
95 0
96 0
97 0
98 0
99 0
100 0
101 0
102 0
103 0
104 0
105 0
106 0
107 0
108 0
109 0
110 0
111 0

```

```

private synchronized void getAllBtsStatus(List<BtsStatus> btsStatusList) {
    try {
        Line 68, Col 3: statement executed 5 times. StatusManager.class:
        for (BtsStatus btsStatus : btsStatusList) {
            if (lookupBtsStatusItems.containsKey(btsStatus.getBtsProjectID())) {
                List<BtsStatus> obj = lookupBtsStatusItems.get(btsStatus.getBtsProjectID());
                obj.add(btsStatus);
            } else {
                List<BtsStatus> obj = new ArrayList<BtsStatus>();
                obj.add(btsStatus);
                lookupBtsStatusItems.put(btsStatus.getBtsProjectID(), obj);
            }
        } catch (Exception e) {
            e.getMessage();
        }
        if (obj != null && !obj.isEmpty()) {
            logger.info("BtsStatusManager All BtsStatus loaded..... " + lookupBtsStatusItems.size());
        }
    }
}

public synchronized Map<String, List<BtsStatus>> getAllBtsStatus() {
    return lookupBtsStatusItems;
}

private void setBtsStatus(String btsProjectID, List<BtsStatus> btsStatusList) {
    if (logger.isDebugEnabled()) {
        logger.debug("BtsProjectID: " + btsProjectID);
        logger.debug("BtsStatusList.size(): " + btsStatusList.size());
    }
    // 해당 BtsStatusItems에 해당 BtsProjectID가 있는지 확인하여 있는 경우 해당 정보를 삭제한다.
    if (lookupBtsStatusItems.containsKey(btsProjectID)) {

```

Enable CovCompPlot plugin



4.7 LOC & Complexity with N'SIQ Collector

N'SIQ Collector

- ❖ **General tool to measure code size and complexity**

- ❖ **What should Measure?**
 - LOC : Code without comments and blanks
 - Complexity : Depends on each team decision

- ❖ **Measures excludes followings**
 - Patch or Code developed by other teams, open source, outsourcing...
 - Auto generated code from Lex or Yacc.
 - Windows message loop which have higher complexity in nature

4.7 LOC & Complexity with N'SIQ Collector

Install and setup N'SIQ Collector

❖ Install N'SIQ Collector

- Download N'SIQCollector binary
(<http://dev.naver.com/projects/nsiqcollector>)
- Unzip the downloaded binary

❖ Install Hudson N'SIQ Collector Plugin

- Installable from Official Jenkins Update Site
- Manage Hudson → Configure System → Configure N'SIQ Collector
 - ✓ Input N'SIQ Collector executable location

Configure N'SIQ Collector

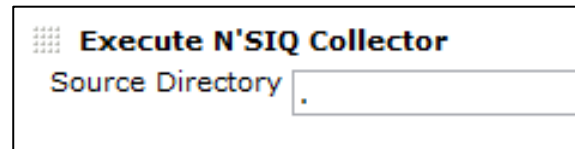
N'SIQ executable

4.7 LOC & Complexity with N'SIQ Collector

Install and setup N'SIQ Collector

❖ Setup N'SIQ Collector per Hudson plugin

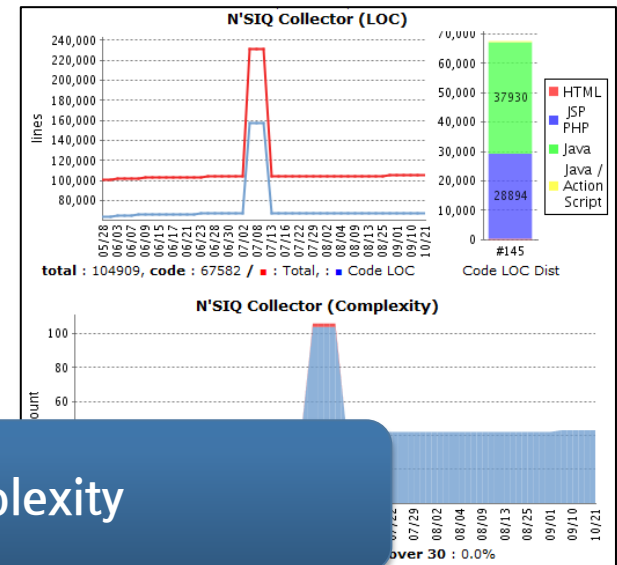
- Configure → Add build step → Execute N'SIQ Collector
- Put relative path to be analyzed in Source Directory field



- Click "Publish N'SIQ Collector" in Post-build Actions
 - ✓ Enable all checkbox in sub menu
- Save

❖ Result

- Click "Build Now"



Add N'SIQ Collector LOC and Complexity



4.8 Static Analysis with Klocwork

Static Analysis

Only Demo

❖ Test vs Static Analysis

- Test = Test Case Execution Time + Defect Cause Analysis Time
- Static Analysis = Only Analysis Time

❖ Static analysis detects possible defects in build time like following

❖ Static analysis reports the step how the defect is reproduced.

```
HttpClient client = getHttpClient();
PostMethod method = getPostMethod(rd, queryParameter, authorization);
client.executeMethod(method);
if ((method.getStatusCode() / 100) != 2) {
    throw new GatewayException("000");
}

OAuthToUserParser parser = new OAuthToUserParser();
OAuthResult result = parser.parse(method.getResponseBodyAsStream());

if (!"00".equals(result.getCode())) {
    throw new GatewayException("541");
}

resultBind(rd, method, result);
return
}
private void
if ("cc
Hea
if
}
}
} else
if
}
} rd
}
}
private Pos
PostMet
method.
method.
method.
if (Sta
rec
}
method.
```

Loc. Code line: 87, column: 9
NPE_CONST
Message Null pointer dereference of 'parser.parse(...).naverUser' where null comes from constant
Level Critical(1)
State Existing
Status Analyze
Comment
change

Error Trace

- ID 0 : **isValid()** - src/main/java/com/naver/api/security/OAuthToUserManager.java
1. line 61 : [source] 'parser.parse(...).naverUser'
- REF ID : 1
2. line 87 : [sink] parameter 'result.naverUser' of call to resultBind(...)
- REF ID : 4
- ID 1 : **parse()** - src/main/java/com/naver/api/security/parser/OAuthToUserParser.java
1. line 38 : [source] 'new OAuthResult(...).naverUser'
- REF ID : 2
2. line 57 : returned 'result'
- ID 2 : **OAuthResult()** - src/main/java/com/naver/api/security/parser/OAuthResult.java
1. line 23 : [source] 'this.naverUser'
- REF ID : 3
- ID 3 : **OAuthResult()** - src/main/java/com/naver/api/security/parser/OAuthResult.java
1. line 18 : [source] null assigned to 'naverUser'
- ID 4 : **resultBind()** - src/main/java/com/naver/api/security/OAuthToUserManager.java
1. line 99 : tracking 'result.getNaverUser()'
2. line 99 : [sink] direct dereference



4.8 Static Analysis with Klocwork

Klocwork Rules

Only Demo

L1~L4 : 101

01(Critical)	19	03(Error)	29
Cross-site Scripting (XSS)	2	Android Issues	4
Data Injection	2	Data Injection	2
Denial of Service	1	Denial of Service	1
Information Leaks	1	Ignored Return Values	3
Possible Runtime Failures	6	Information Leaks	1
Process and Path Injection	4	Possible Runtime Failures	1
Suspicious Code Practices	1	Resource Leaks	15
Unvalidated User Input	2	Unsafe Code Practies	1
		Unvalidated User Input	1
02(Severe)	40	04(Unexpected)	13
Android Issues	4	Data Injection	1
Denial of Service	3	Poor Error Handling	1
Process and Path Injection	2	Possible Runtime Failures	1
Redundant Code	5	Redundant Code	2
Suspicious Code Practices	9	Suspicious Code Practices	3
Threads and Synchronization Issues	2	Threads and Synchronization Issues	4
Unvalidated User Input	1	Weak Encapsulation	1
Use After Free	11		
Weak Encryption	3		

4.8 Static Analysis with Klocwork

How static analysis works

Only Demo

❖ NPE.COND Defect

- Guess if static analysis assumes that all method parameter can be given as null value
 - Excessive false alarm
- What if there is a condition in which check the parameter is null or not...
 - Is it safe to say null value can be given as parameter?

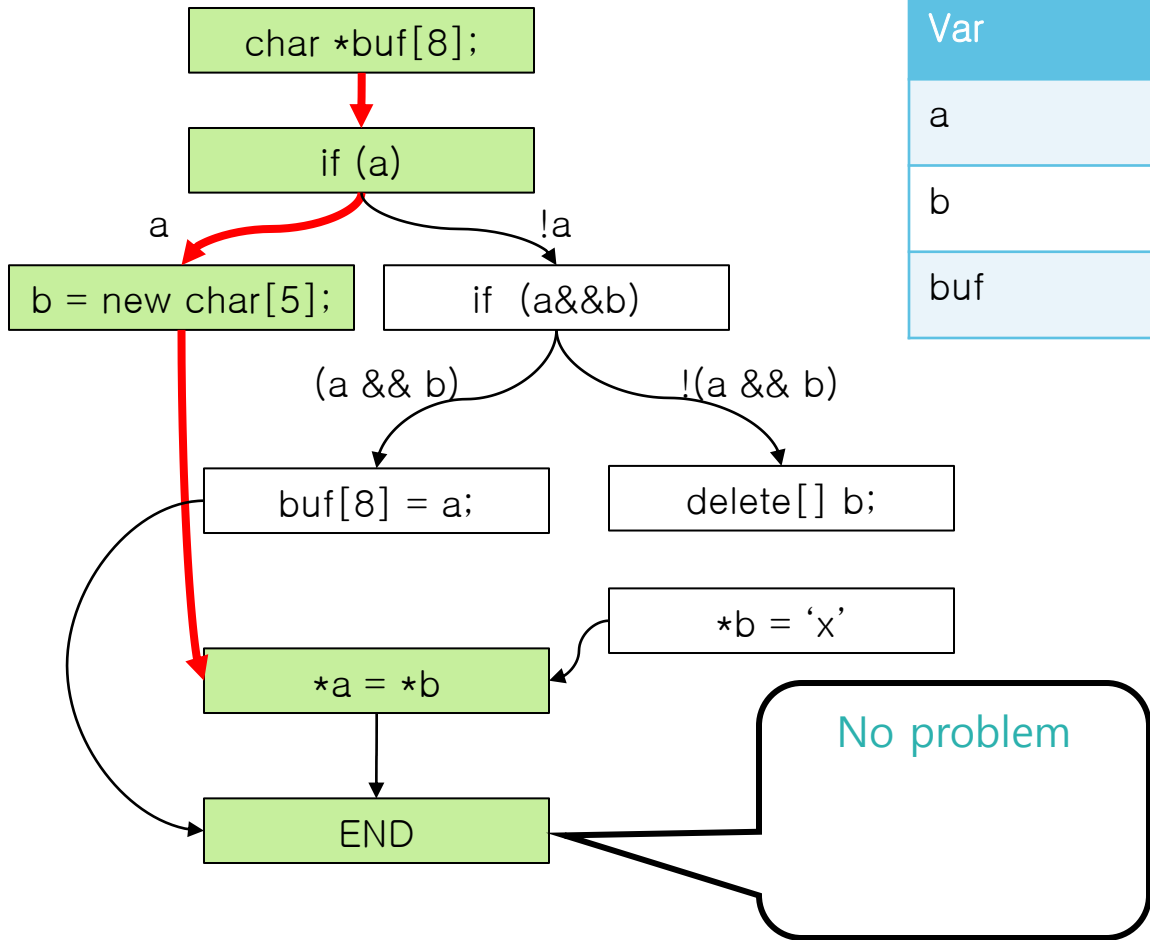
```
#define SIZE 8
void set(char *a, char *b) {
    char *buf[SIZE];
    if (a) {
        b = new char[5];
    } else {
        if (a && b) {
            buf[size] = a;
            return;
        } else {
            delete[] b;
        }
        *b = 'x';
    }
    *a = *b;
}
```

4.8 Static Analysis with Klocwork

How static analysis works

Only Demo

❖ Value Tracing



Var	State
a	Not NULL
b	Not NULL
buf	Not NULL

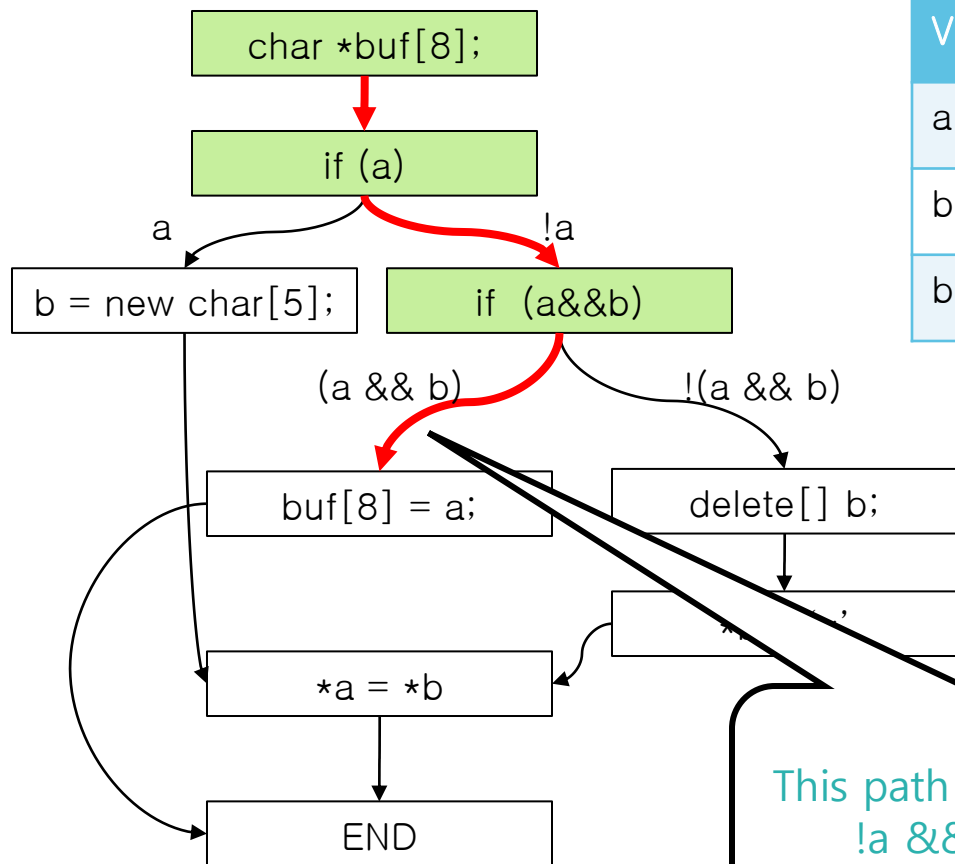


4.8 Static Analysis with Klocwork

How static analysis works

Only Demo

❖ Impossible path



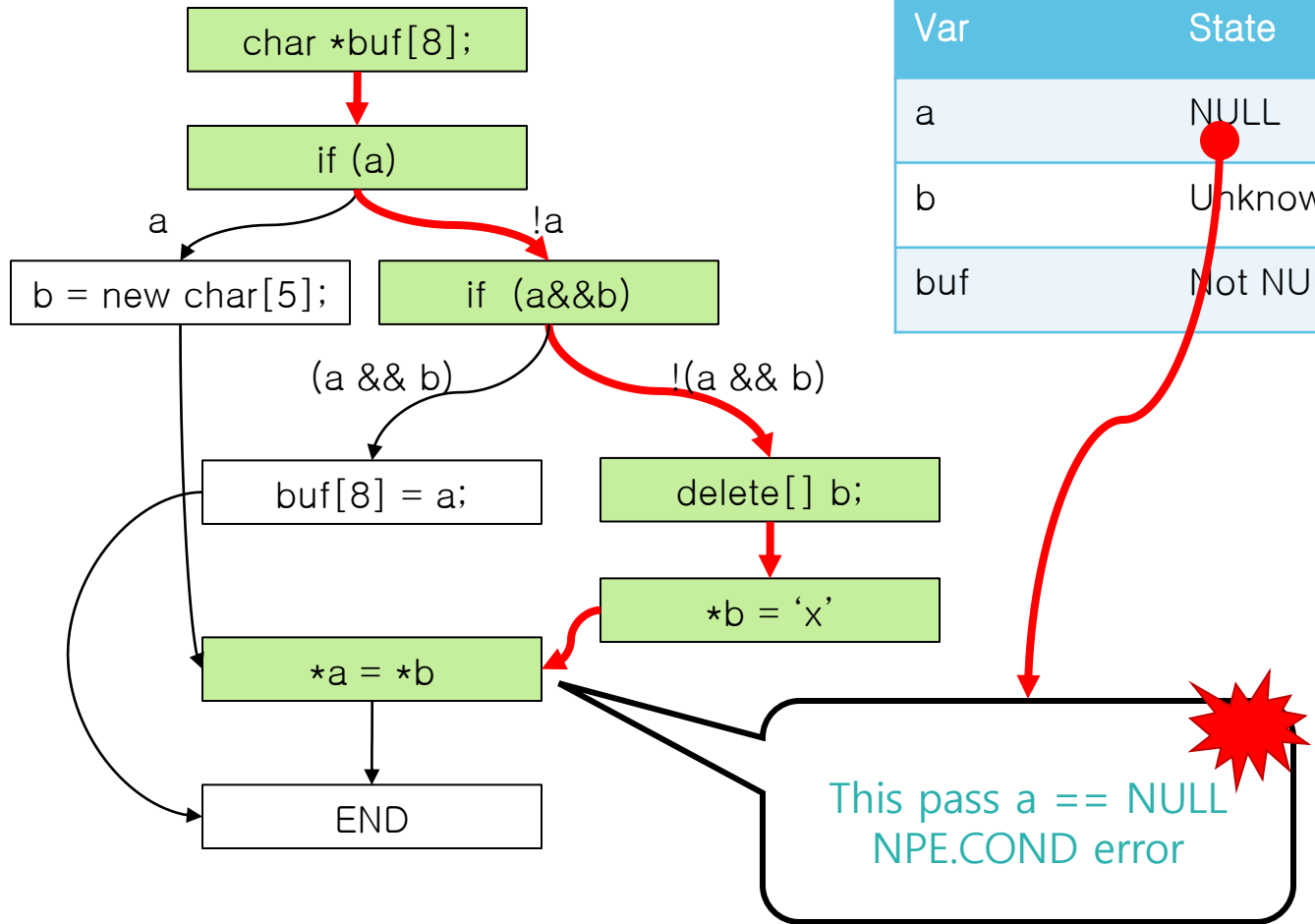
Var	State
a	NULL
b	Unknown
buf	Not NULL

4.8 Static Analysis with Klocwork

How static analysis works

Only Demo

❖ Error Case



❖ Limitation on Static Analysis

- If there is no Source Code
 - ✓ Lib / Dll / Jar
- If the value is given from external env
 - ✓ socket / scanf
- If no one know which class will be wired in the other class in compile time
 - ✓ E.g) Spring

❖ How to overcome.

- Remove unnecessary interfaces.
- Provide Knowledge base
(a user defined info about the methods with no source code)

4.8 Static Analysis with Klocwork

Configure Klocwork Plugin per project

Only Demo

❖ Add “Klocwork Builder”

- Configure → Add build step → Click “Execute Klocwork”
- Input “Execute Klocwork” configuration.

- ✓ Build Spec
 - “maven”
- ✓ Config Filename
 - “pom.xml”
- ✓ Build Parameter
 - “kw:run -P klocwork”
- ✓ Knowledge Base
 - Additional Info for library
 - “Java_general”

The screenshot shows the 'Build' configuration dialog for the 'Execute Klocwork' step. The configuration is as follows:

Field	Value
Klocwork Version	9.X
Build Spec	maven (java)
Source Encoding	UTF-8
Config Filename	pom.xml
Build Parameter	clean compile -P klocwork
Knowledge Base	java_general
Compatibility mode	<input checked="" type="checkbox"/>

Buttons: Advanced..., Delete

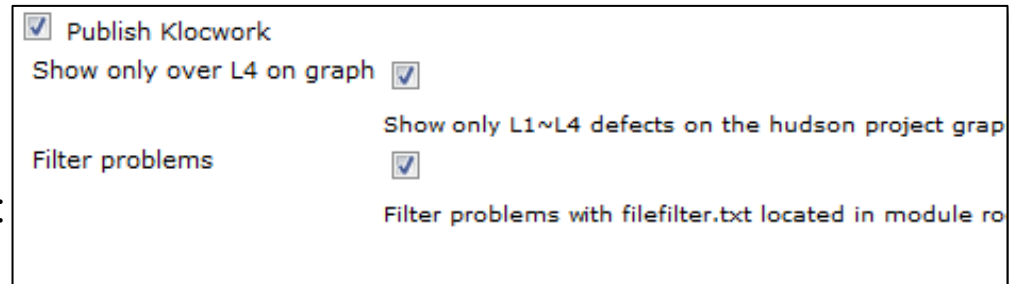
4.8 Static Analysis with Klocwork

Configure Klocwork Plugin per project

Only Demo

❖ Enable Klocwork Publisher

- Click "Publish Klocwork" in Post-build Action
- Show only over L4 on graph : Click If you want to see only L1~L4 errors



Publish Klocwork

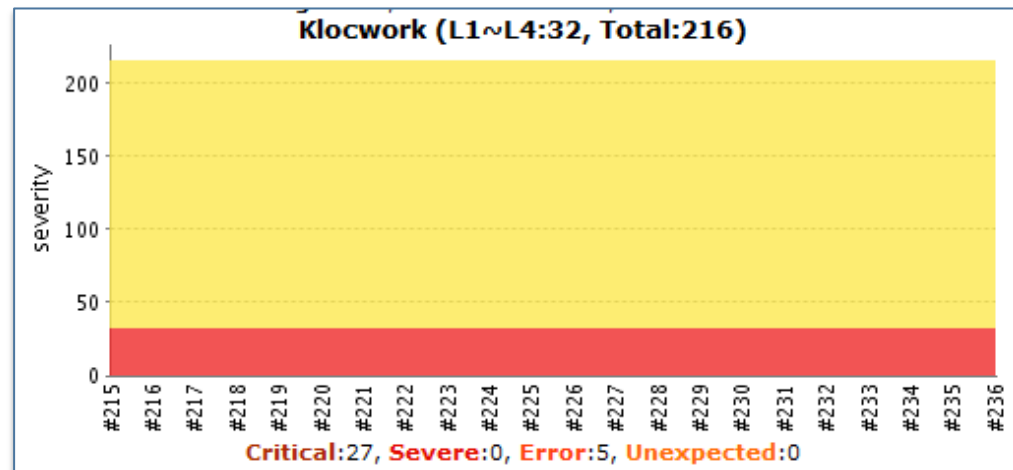
Show only over L4 on graph

Filter problems

Show only L1~L4 defects on the hudson project graph

Filter problems with filefilter.txt located in module root

❖ Click Build Now



4.8 Static Analysis with Klocwork

Configure Klocwork Plugin per project

Only Demo

- ❖ Click the shown graph.
 - Dig into the defect details

Klocwork Result				
Summary				
item	crit.	severe	error	unexp.
cs_open_bloc/src/main/java/com/nhncorp/cs/bo	4	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/common	0	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/dao	0	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/dto	0	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/support/ds	1	0	4	0
cs_open_bloc/src/main/java/com/nhncorp/cs/support/encryption	5	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/support/external	12	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/support/login	1	0	0	0
cs_open_bloc/src/main/java/com/nhncorp/cs/util	4	0	1	0

```
368 job.start();
369
370 StepResult stepResult = null;
371 if (pctx != null) {
372     stepResult = clonePctxFile(pctx, sdocBase, tdocBase, spubPath, tpubPath);
373     job.setStepResult(stepResult);
374 }
375
376 job.end();
377 return job;
378 }
379
380 /**
381  * @param PlanC
382  * @param pctx
383  * @param uidL
384  * @param uidO
385  * @param tdoc
386  * @param tpub
387  * @return
388  */
389 private JobResult
390 JobResult :
391 job.start()
392
393 StepResult
394 if (pctx !=
395 stepRes
396 job.set
397 }
398
399 job.end()
400 return job;
401 }
402
403 /**
```

Loc. line: 373, column: 13
Code NPE_CONST
Message Null pointer dereference of 'stepResult.exception' where null comes from constant
Level Critical(1)
State Existing
Status Analyze
Comment
Error Trace

- ID 0 : **StepResult()** - cs_open_bloc/src/main/java/com/nhncorp/cs/support/external/StepResult.java
1. line 17 : [source] null assigned to 'exception'
- ID 1 : **StepResult()** - cs_open_bloc/src/main/java/com/nhncorp/cs/support/external/StepResult.java
1. line 22 : [source] 'this.exception'
- REF ID : 0
- ID 2 : **clonePctxFile()** - cs_open_bloc/src/main/java/com/nhncorp/cs/support/external/CloneArtifactsRunnable.java
1. line 462 : [source] 'new StepResult().exception'
- REF ID : 1
2. line 530 : returned 'result'
- ID 3 : **toString()** - cs_open_bloc/src/main/java/com/nhncorp/cs/support/external/StepResult.java
1. line 77 : [sink] direct dereference
- ID 4 : **setStepResult()** - cs_open_bloc/src/main/java/com/nhncorp/cs/support/external/JobResult.java
1. line 99 : [sink] 'stepResult.exception'
- REF ID : 3
- ID 5 : **cloneArtifactFiles()** - cs_open_bloc/src/main/java/com/nhncorp/cs/support/external/CloneArtifactsRunnable.java
1. line 372 : [source] 'stepResult.exception'
- REF ID : 2
2. line 373 : [sink] parameter 'stepResult.exception' of call to setStepResult(...)
- REF ID : 4

Add “Execute Klocwork” build step
Enable “Publish Klocowork”
Click “Build Now” and see defects found



4.9 Coding Standard Conformance with Checkstyle

Checkstyle

- ❖ **Most famous Java coding style checker**

- NHN defined our own coding style rules by customizing checkstyle rules.

4.9 Coding Standard Conformance with Checkstyle

Install Hudson Checkstyle Plugin

- Configure Hudson → Manage plugins → Available Tab → Click Checkstyle Plugin → Install → restart

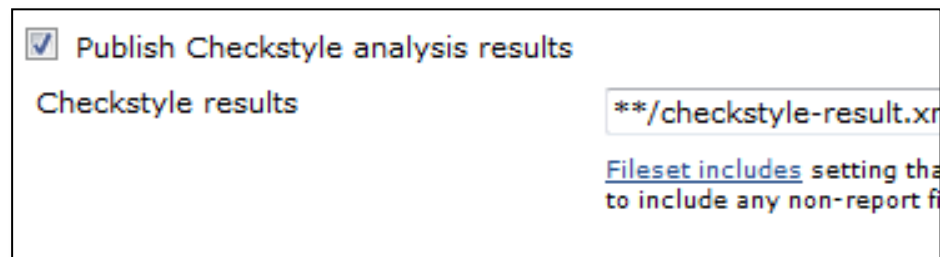
<input type="checkbox"/>	This plugin posts build results to Twitter.
Build Reports	
<input type="checkbox"/>	CCCC Plugin This plugin generates the trend report for CCCC (C and C++ Code Counter).
<input checked="" type="checkbox"/>	Checkstyle Plugin This plugin generates the trend report for Checkstyle , an open source static code a
<input type="checkbox"/>	Clover Plugin This plugin allows you to capture code coverage reports from Clover . Hudson will g
	CodeScanner Plugin

4.9 Coding Standard Conformance with Checkstyle

Configure Hudson Checkstyle Plugin per project

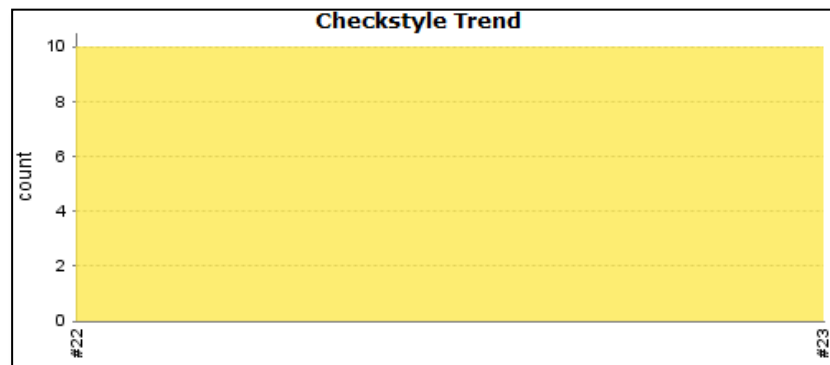
- ❖ **Add checkstyle goal in maven goal list**
 - Add checkstyle:checkstyle goal in front of other goals

- ❖ **Add checkstyle publisher**
 - Check "Publish Checkstyle analysis results" and specify the Checkstyle xml results path



Configuration interface for the Hudson Checkstyle plugin. It shows a checked checkbox for "Publish Checkstyle analysis results". Below it, the text "Checkstyle results" is followed by a text input field containing the fileset pattern `**/checkstyle-result.xml`. A note below the input field states: "Fileset includes setting that to include any non-report files".

- ❖ **Run "Build Now"**
 - Dig into graph



Configure Checkstyle in Hudson project
Click "Build Now" and see found violations

4.9 Calculate QP metrics with QD plugin

Quality Dashboard Plugin

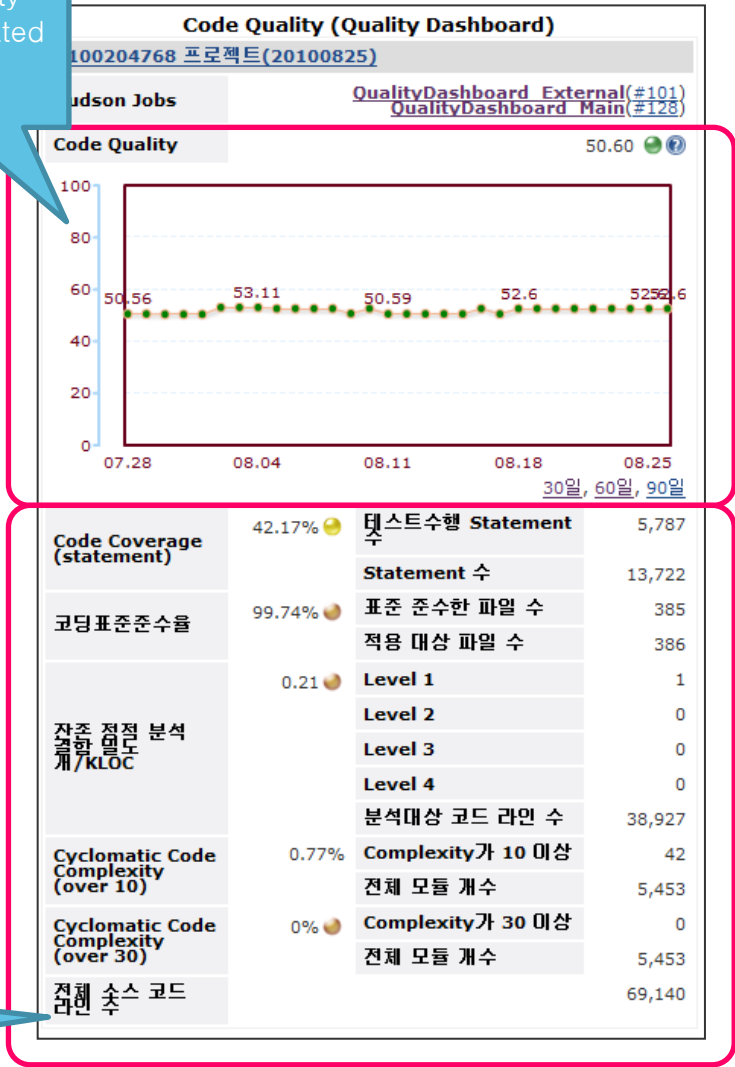
❖ Hudson QD PlugIn

- Send collected metric from Hudson to Quality Dashboard
- Show Code Quality Value calculated by Quality Dashboard
- Summarize multiple Hudson project metrics

❖ Measures

- Code Coverage (Statement / Branch)
- Coding Standard Conformance Rate
- Static Analysis Defect Density
- Complexity / LOC

Code Quality Value calculated by QD



Summarized Measures



4.9 Calculate QP metrics with QD plugin

Configure Quality Dashboard Plugin

❖ Configure QD Plugin in a project

- Configure Project → Enable “Publish to Quality Dashboard”
 - ✓ Select the Hudson Plugins used.

<input checked="" type="checkbox"/> Publish to Quality Dashboard	
프로젝트 타입	수집전용
API 사용 인증키	1219e275dcab36a43ffd5d7deb9e899c228c93bd57a40c80a46242ac7ff242e76
간접 정적 분석 결과 업로드 방법	Klocwork
Code Coverage(branch)	None
Code Coverage(statement)	Clover(Statement)
코딩 표준 준수율	CheckStyle
Cyclomatic Code Complexity	NSIQ Collector for complexity
소스 코드 라인수	NSIQ Collector for loc
Verbose	<input type="checkbox"/>

4.9 Calculate QP metrics with QD plugin

Project Type

❖ values

- Collect/Send : When you want to send the metrics if the metrics are collected.
- OnlyCollect : When you want to only collect metrics.
- OnlySend : When you want to send metrics collected by the other projects
- NoCollect/NoSend : When you disable this project

❖ How to use

- ✓ When you like to summarize A, B, C project and You want to send the collected metrics only when C is built.
 - A : OnlyCollect, B : OnlyCollect, C: Collect/Send
 - A, B, C's API Key should be same
- ✓ When you like to summarize A, B, C project whenever each project is built, However you want to send the metrics to QualityDashboard one a week.
 - A : OnlyCollect, B : OnlyCollect, C: OnlyCollect,
Create Separate Dummy D Project and set it OnlySend
 - A, B, C, D's API Key should be same

Enable the collection of Coverage / Coding Style / Cyclomatic Complexity / LOC / Static Analysis Defect Density

Thanks

