

# SW 테스트 자동화 기법

(주)시네틱스 대표: 한동준

handongjoon@gmail.com

dongjoon.han@synetics.kr

#PROFESSIONAL IT EDU-PLATFORM

code.presso()

## • 과정 구성

- 단위 테스트와 정적 분석 도구를 활용한 SW 검증 실습
- SW 검증 기술 및 SW 테스트 자동화 도구들을 활용하여, SW 개발 초기부터 단위 테스트와 정적 분석을 자동으로 검증하는 구성 방안 실습

#PROFESSIONAL IT EDU-PLATFORM

## • 과정 특징

- 단위 테스트 및 정적 분석 자동화 기법에 대한 이해
- 지속적 통합과 단위 테스트/정적 분석을 연동한 지속적 검증 방안에 대한 이해

CONTENTS / 002

# 테스트 자동화와 지속적 통합의 관계

# DevOps = Development + Operations

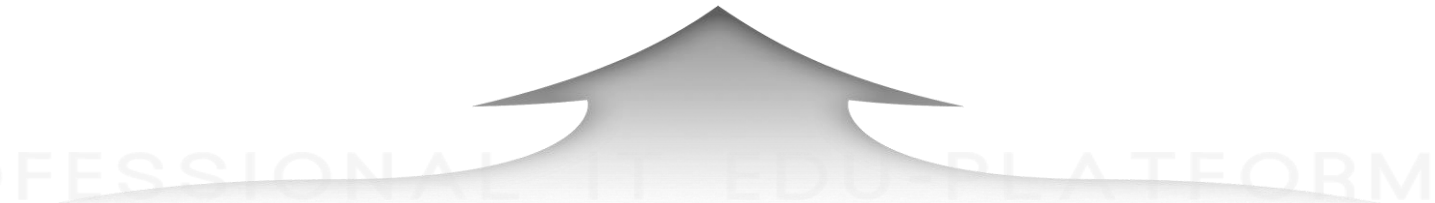
- 단순하게는 개발과 운영의 합성어
- 개발(기획, 설계, 구현, 테스트) 과 운영(배포, 인프라, 모니터링) 의 **밀접한 연계**를 의미
- 애플리케이션과 서비스를 더 빠르게 제공하기 위한 조직의 **문화, 방법, 툴체인**

- 소프트웨어는 비즈니스 지원이 아닌 비즈니스의 중심
- 고객과 사용자는 더 빠른 대응, 더 나은 품질과 안정성을 요구
- 기존의 역할과 책임, 개발 방식, 아키텍처, 인프라의 한계

## IT 기술 트렌드의 변화



## 품질과 안정성이 확보된 소프트웨어를 더 빠르게 제공

- 
- 소프트웨어 개발/운영 담당자의 원활한 의사 소통
  - 개발/운영 프로세스의 단순화 및 자동화 (휴먼 에러 최소화)
  - 지속적인 소프트웨어 품질 및 안정성 확보

## 자동화된 지속적 통합(CI) / 지속적 배포(CD) 환경 구축

• 통합 빌드, 정적 분석, 테스트 커버리지, 함수 복잡도

**SW 통합 오류를 개발 초기부터 예방하는 것**

코딩 시작한 날 • 문제 발생 시 당일 조치

**모든 SW 개발에 적용 가능한 정의**

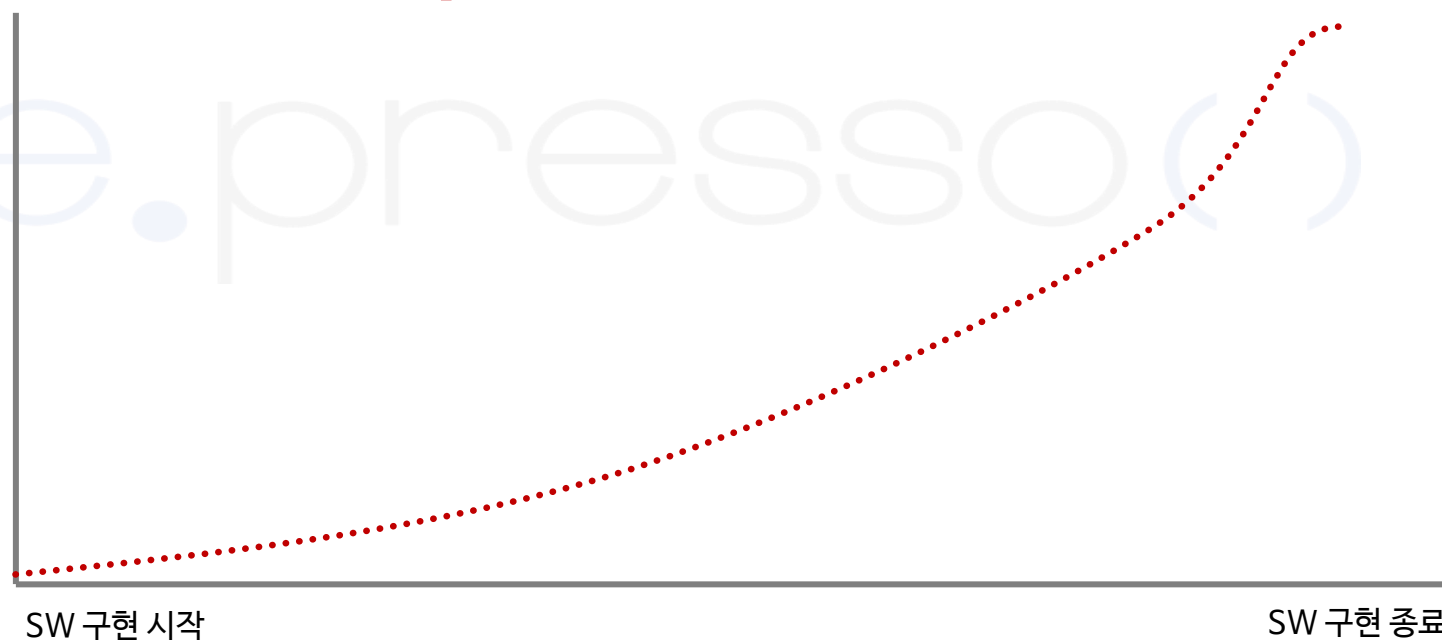


## “소프트웨어 구현 후반부에”

### 소스코드/소프트웨어 문제들

- 빌드 오류 수
- 정적분석 위반사항 수
- 보안 룰셋 위반사항 수
- 테스트 커버리지 불만족 비율
- 함수, 모듈 크기 위반율
- 복잡도 위반율

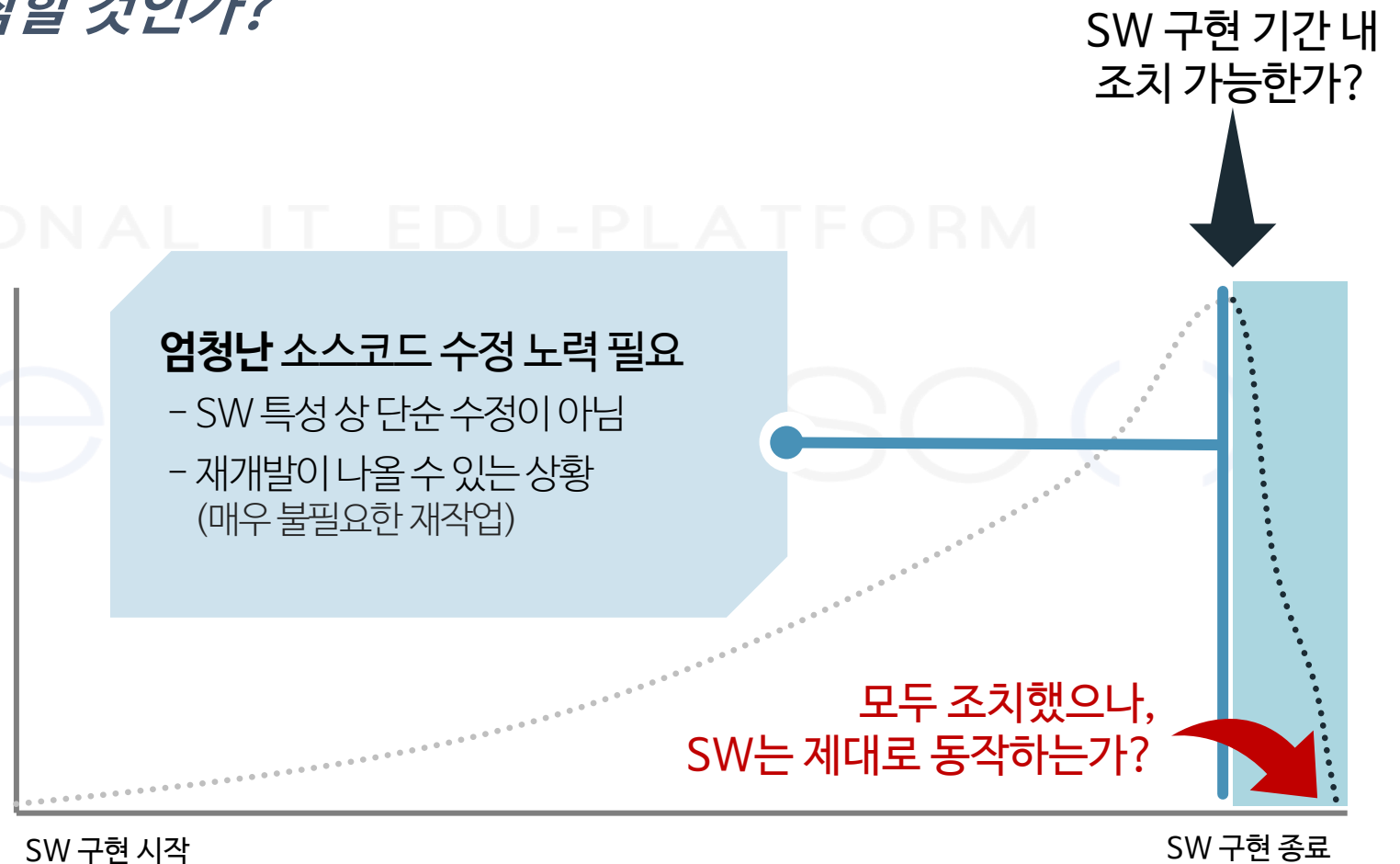
“소스코드가 ISO 26262 요건을  
**너무** 많이 위반했는데요...”



*SW 구현 종료까지 **조치** 가능한가?  
조치가 끝나면 SW가 잘 동작할 것인가?*

## 소스코드/소프트웨어 문제들

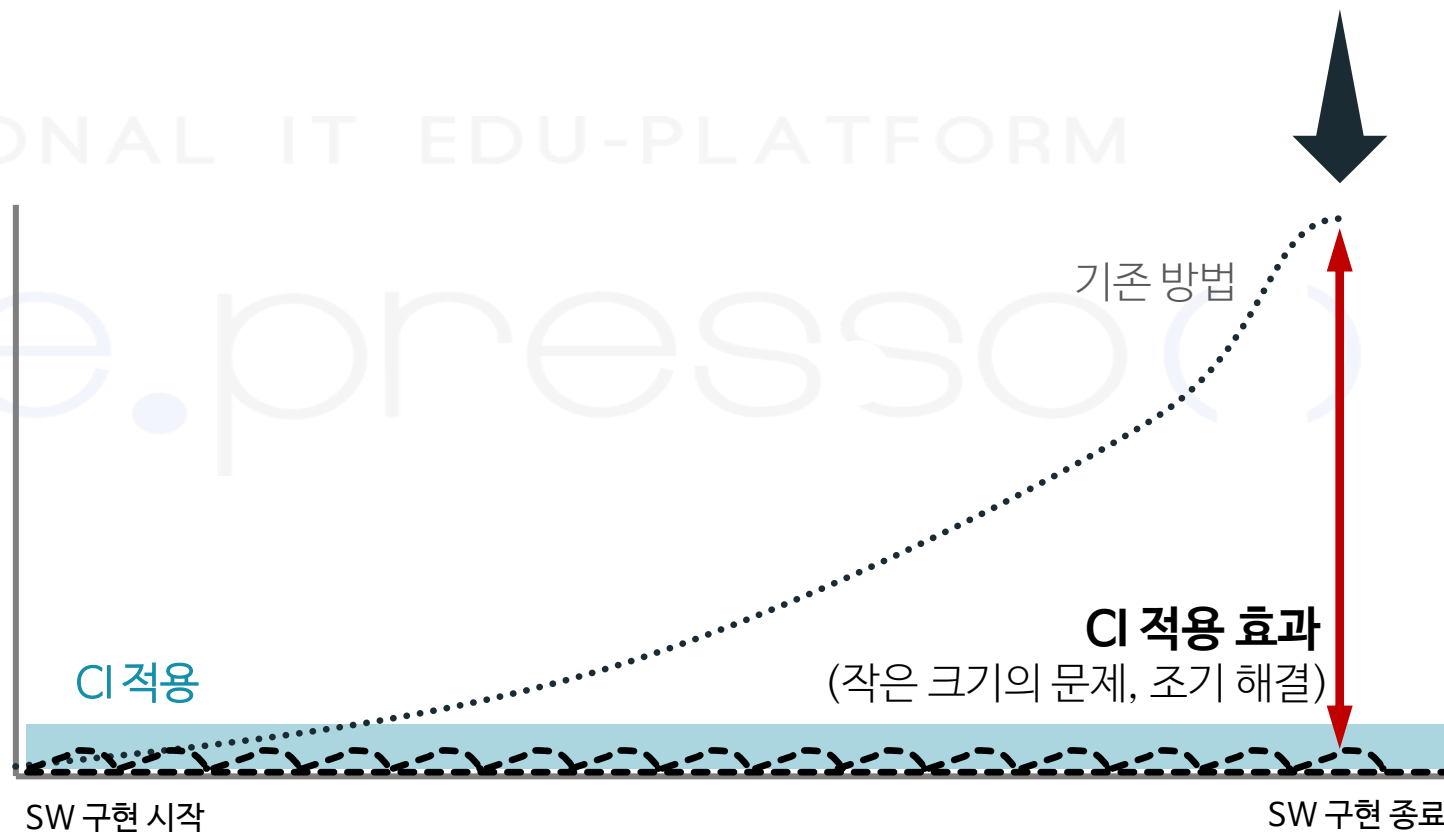
- 빌드 오류 수
- 정적분석 위반사항 수
- 보안 룰셋 위반사항 수
- 테스트 커버리지 불만족 비율
- 함수, 모듈 크기 위반율
- 복잡도 위반율



SW 구현 시작부터 지속적 통합(CI)을 적용,  
도구가 **알아서** 검사/보고하여 품질과 안정성 조기 확보

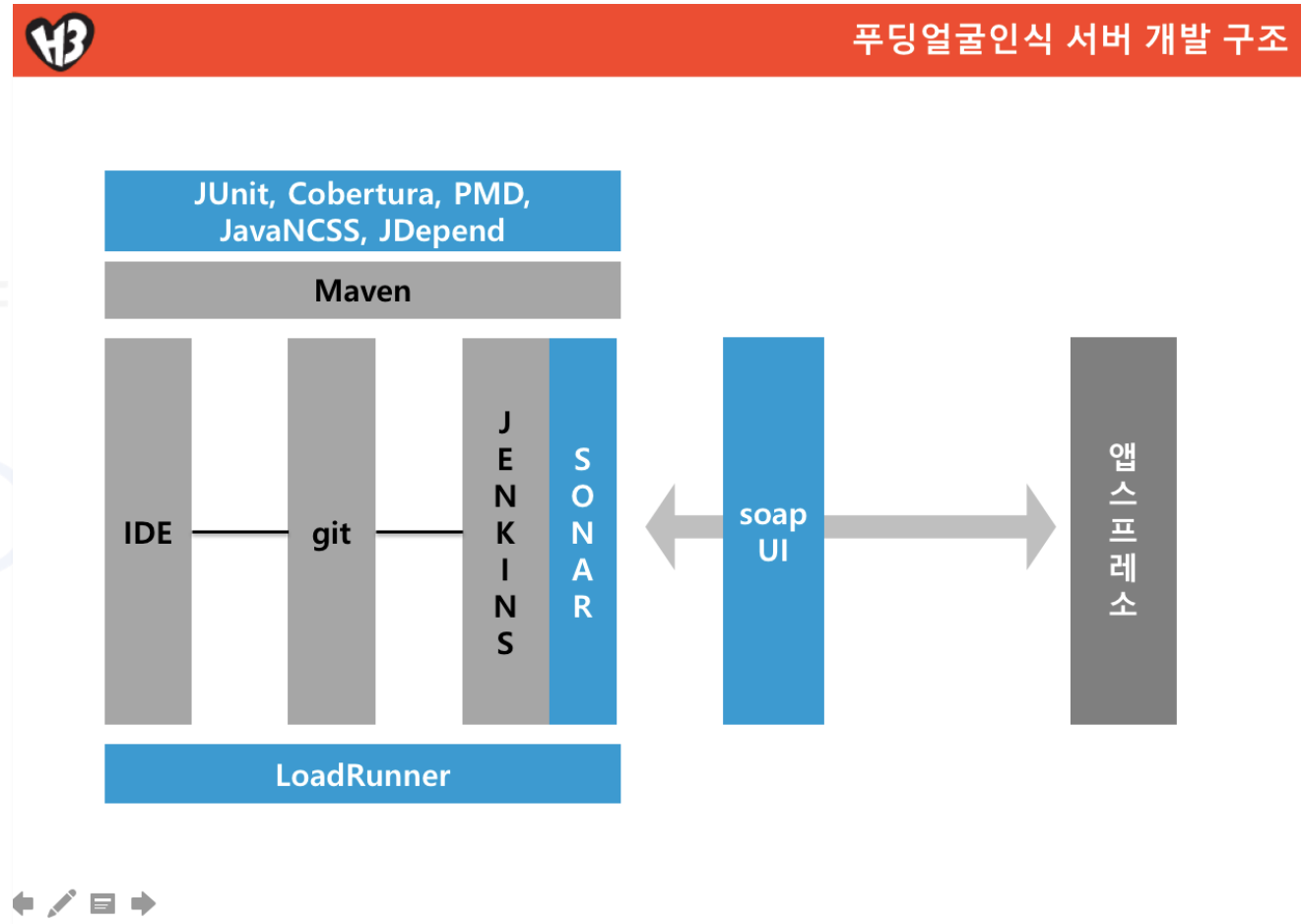
## 소스코드/소프트웨어 문제들

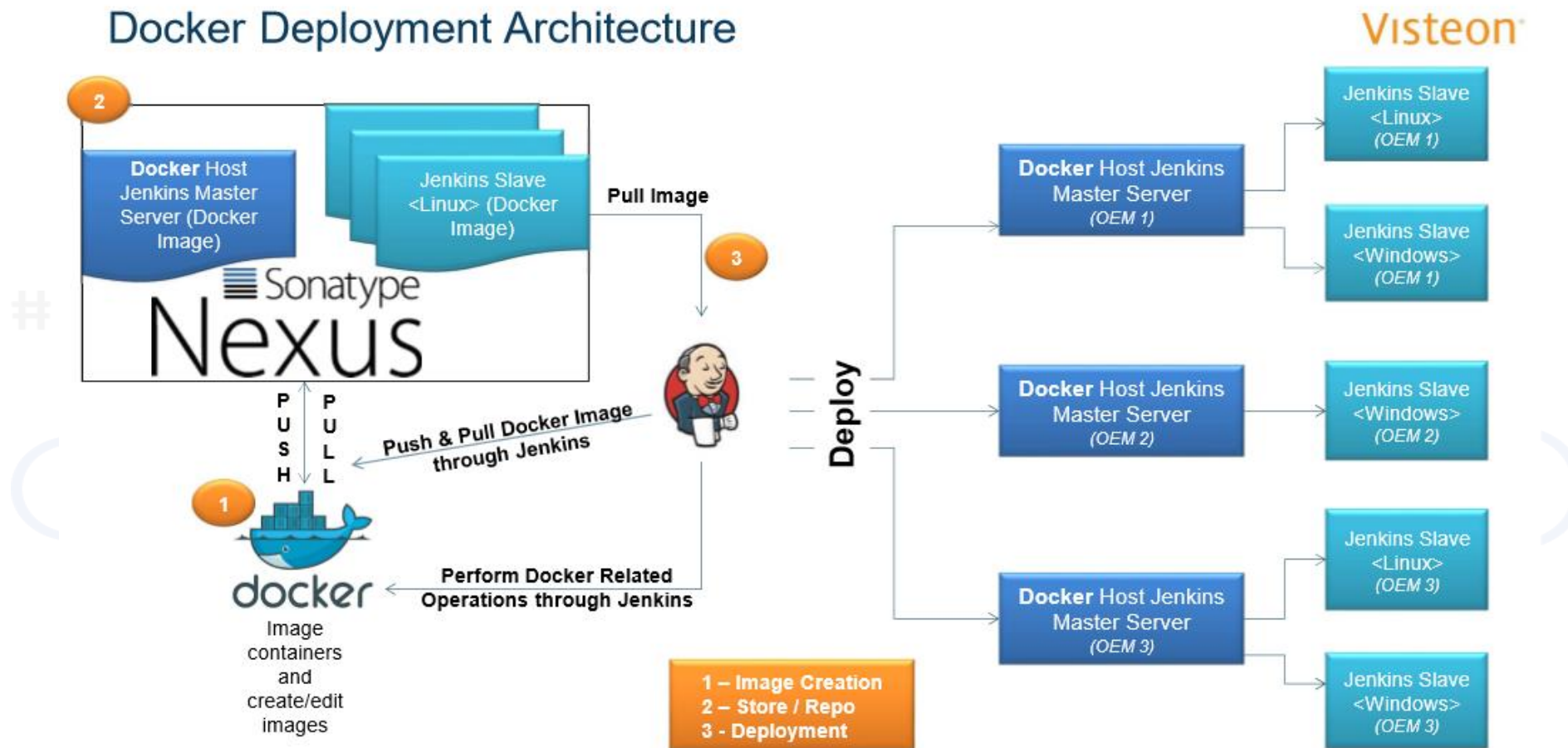
- 빌드 오류 수
- 정적분석 위반사항 수
- 보안 룰셋 위반사항 수
- 테스트 커버리지 불만족 비율
- 함수, 모듈 크기 위반율
- 복잡도 위반율



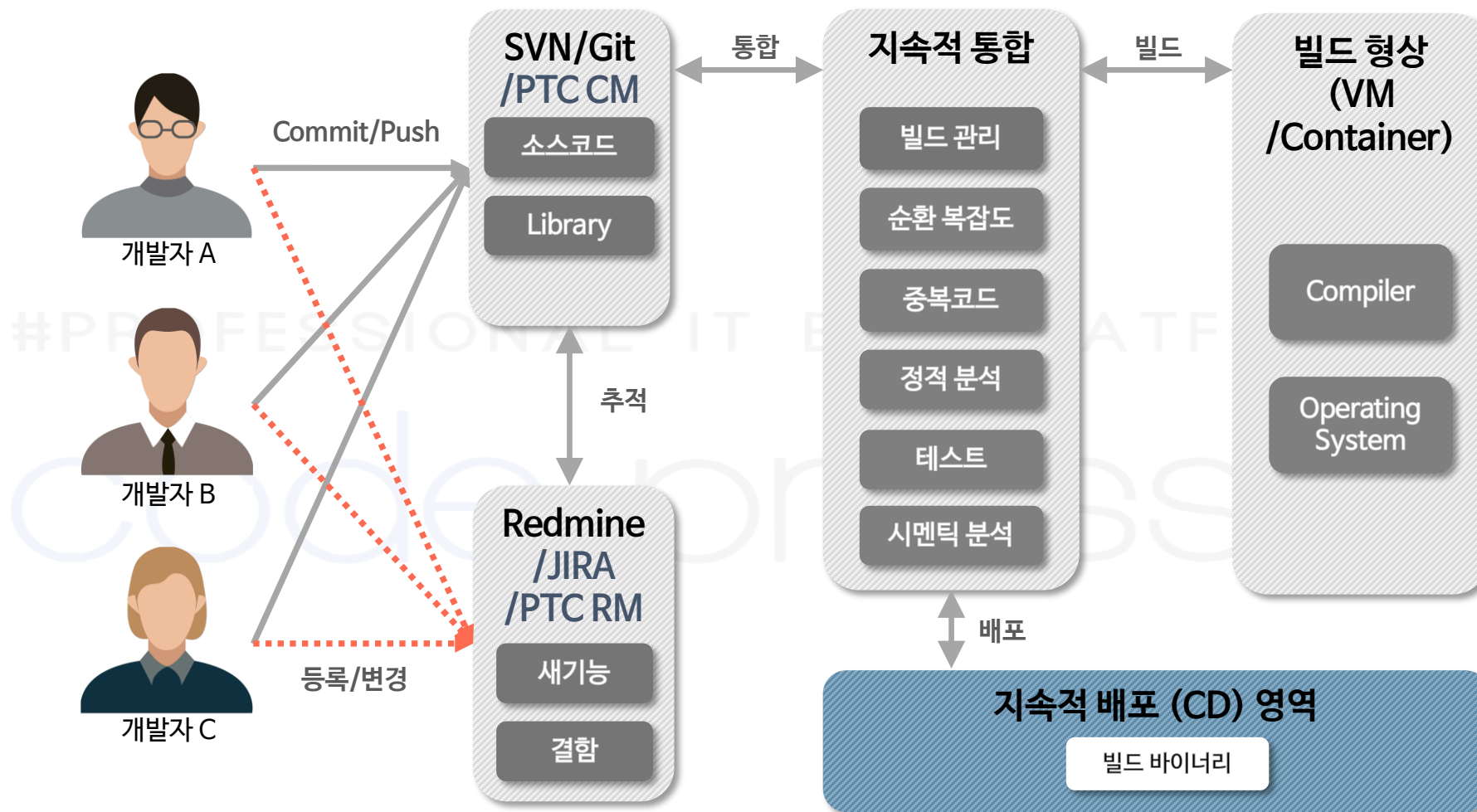
- 소스코드/소프트웨어 빌드 및 품질 문제점을 빠르게 식별하고 조치
- 빌드, 품질 검사, 문제점 보고, 배포 자동화로 휴먼 에러 및 투입 자원 감소
- 빌드 환경의 형상을 소프트웨어 폐기까지 유지
- 오픈소스를 기반으로 구축 가능 및 다양한 구축 사례 공유

[ 2012 H3 컨퍼런스 발표 ] - 10년전 한동준 대표의 지속적 통합 사용사례 발표





출처) 자동차 부품 회사 Visteon의 CI 환경 - 2018 Agile in Automotive



- 1 개발자가 소스코드를 Git 저장소에 Commit/Push
- 2 Jenkins가 정해진 시간에 Git에서 지정된 소스코드 가져오기
- 3 Jenkins가 정해진 스크립트에 따라 빌드/검증 수행
- 4 문제점 발생 E-Mail, Messenger, 이슈 관리 시스템 보고
- 5 Jenkins가 정해진 스크립트에 따라 배포

자동화



- SW 개발을 돕는 새로운 기술의 비교적 **신중한 접근**
- **웹/앱** 개발하는 그들만의 도구라는 **오해**
- 호환이 잘 안되는 **너무 다양한 컴파일러** 및 버전
- **민감한 빌드 환경**
- **상용 도구 위주**의 SW 품질 확보
- 이미 사용하고 있지만 외부에 **공유를 꺼려함**

## [ QAC/QAF ]

 **PRQA Plugin**

73 Added by [Mads Nielsen](#), last edited by [Oscar Forner Martinez](#) on Aug 26, 2016 ([view change](#))


**Jenkins**

- Home
- Mailing lists
- Source code
- Bugtracker
- Security
- Advisories
- Events
- Donation

**Plugin Information**

Plugin ID	prqa-plugin	Changes	In Latest Release Since Latest Release
<b>Latest Release</b>	2.1.0 ( <a href="#">archives</a> )	<b>Source Code</b>	<a href="#">GitHub</a>
<b>Latest Release Date</b>	Sep 26, 2016	<b>Issue Tracking</b>	<a href="#">Open Issues</a>
<b>Required Core</b>	1.480.3	<b>Pull Requests</b>	<a href="#">Pull Requests</a>
<b>Dependencies</b>		<b>Maintainer(s)</b>	<a href="#">Igor Kostenko</a> (id: isanych)


## [ VectorCAST ]

 **VECTOR software**

Home > Resources > Whitepapers

### Building a Flexible and Automated Testing Infrastructure Using VectorCAST/Manage and Jenkins

## [ CodeSonar ]

 **CodeSonar Plugin**

8 Added by [Mads Nielsen](#), last edited by [Andrius Ordoian](#) on Sep 12, 2016 ([view change](#))

**Jenkins**

- Home
- Mailing lists
- Source code
- Bugtracker
- Security
- Advisories
- Events
- Donation

**Plugin Information**

Plugin ID	codesonar	Changes	In Latest Release Since Latest Release
<b>Latest Release</b>	2.0.4 ( <a href="#">archives</a> )	<b>Source Code</b>	<a href="#">GitHub</a>
<b>Latest Release Date</b>	Sep 12, 2016	<b>Issue Tracking</b>	<a href="#">Open Issues</a>
<b>Required Core</b>	1.509.2	<b>Pull Requests</b>	<a href="#">Pull Requests</a>
<b>Dependencies</b>	<a href="#">credentials</a> (version: 1.24) <a href="#">job-dsl</a> (version: 1.37, optional)	<b>Maintainer(s)</b>	<a href="#">Pragma Josra</a> (id: pragma_josra)

## [ Polyspace ]

 **Polyspace\_Jenkins\_plugin.zip**

by [Ram](#)  
21 Jul 2014 (Updated 01 Sep 2016)

Plugin to integrate Polyspace verification into your Jenkins job

[Watch this File](#)

Be the first to rate this file!

3 Downloads (last 30 days)  
File Size: 9.82 KB  
File ID: #47148  
Version: 1.1.0.1

## [ LDRA ]

**Jenkins**

### LDRA interfaces with the Jenkins Continuous Integration Platform.

The LDRA tool suite offers support for Jenkins, a continuous integration platform that automates and manages the build process for large-scale, system-level software projects. Jenkins connects the build and version control systems with quality management and automated testing. By linking the LDRA tool suite to Jenkins, the platform can invoke the LDRA tool suite to perform analysis, track test and verification fulfilment, and assess overall improvements or regressions in code quality over time.

## [ Sparrow ]

**소프트웨어 개발 프로세스 통합**

- ✓ 기존의 개발환경을 그대로 활용하여 소스코드 분석 및 결과 확인
- ✓ GUI 분석 도구로 일괄 점검
- ✓ REST API / CLI 분석 도구로 배치 · 스케줄링 분석
- ✓ REDMINE, JIRA, Jenkins, Hudson, FRISM, AURORA, Harvest, eCAMS 등과 연동하여 보안 및 품질 기준에 부합하지 않는

CONTENTS / 014


# CPP 빌드를 위한 Item 생성 1

- 메인화면의 '새로운 Item' 클릭


- 최신 버전은 Maven 프로젝트가 Freestyle 프로젝트에 포함

Enter an item name


» This field cannot be empty, please enter a valid name




**Freestyle project**  
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.




**Maven project**  
Maven 프로젝트를 빌드합니다. Jenkins은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.



**Pipeline**  
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.




**External Job**  
이 유형의 작업은 원격 장비처럼 Jenkins 외부에서 동작하는 프로세스의 실행을 기록하는 것을 허용합니다. 그렇게 설계되어서, 기존의 자동 시스템의 대시보드로서 Jenkins을 사용할 수 있습니다.



**Multi-configuration project**  
다양한 환경에서의 테스트, 플래폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.

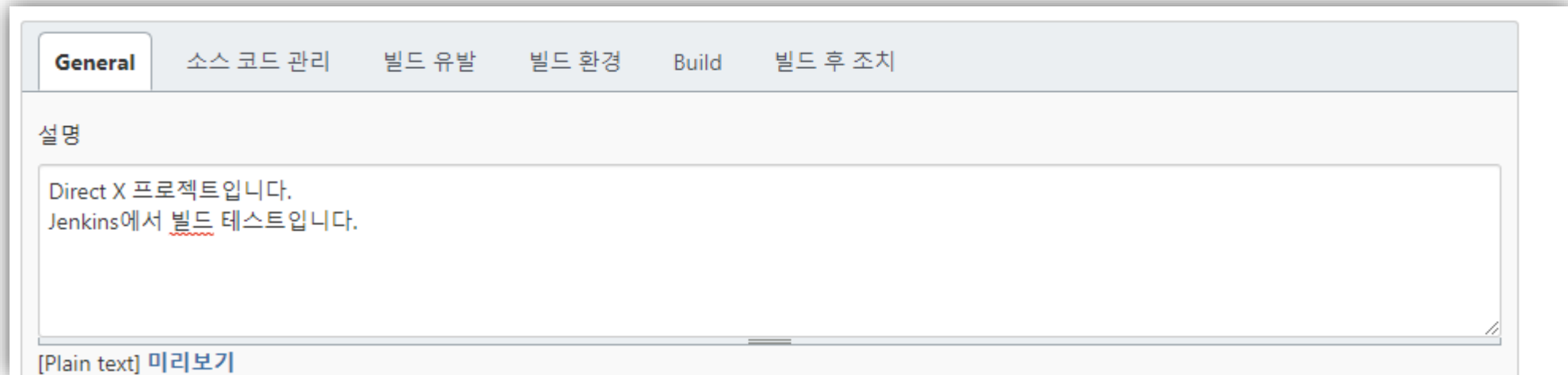
if you want to create a new item from other existing, you can use this option:







Copy from

OK

- Job에 대해 알리고 싶은 내용을 작성



The screenshot shows the 'General' tab of a Jenkins job configuration page. The 'Description' field is active, containing the text: 'Direct X 프로젝트입니다.' followed by 'Jenkins에서 빌드 테스트입니다.' where '빌드' is underlined in red. Below the text area, there is a label '[Plain text]' and a blue link '미리보기' (Preview).

- ☐ 오래된 빌드 삭제 
- ☐ 이 빌드는 매개변수가 있습니다 
- ☐ 빌드 안함 (프로젝트가 다시 빌드를 할 때까지 새로운 빌드가 실행되지 않습니다.) 
- ☐ 필요한 경우 concurrent 빌드 실행 

설정 항목	설명
오래된 빌드 삭제	이 항목을 설정하지 않으면 Jenkins는 빌드 결과를 계속 유지한다. 그러나 하드디스크 용량 등을 고려했을 때, 지난 빌드 결과는 삭제하는 것이 필요할 수도 있다. 이 항목을 선택하면 특정 기간이나 빌드 수만큼 빌드 결과를 보관할 수 있다.
이 빌드는 매개변수가 있습니다.	빌드 시 매개변수를 정의하고 사용할 수 있다. 매개변수는 단순히 Text도 가능하지만, Subversion의 tag 목록도 가능하다.
빌드 안함	이 항목을 선택하면 빌드가 수행되지 않는다. 모든 빌드 설정은 유지하면서, 잠시 빌드를 사용하지 않을 때 활용한다.
필요한 경우 concurrent 빌드 실행	Jenkins는 하나의 Job을 빌드 중 또 다시 해당 빌드를 시작하면 Queue에 쌓이게 된다. 그러나 Queue에는 1개까지만 유지되고, 나머지 빌드는 무시된다. 이 항목을 선택하면 빌드 머신이 가능한 경우 동시 빌드를 수행한다.

- 빌드를 위해 소스코드 저장소에서 소스코드를 가져오는 설정
- Git의 경우 Branch 지정이 중요
  - Master 또는 Main

소스 코드 관리

☐ None  
☒ Git

Repositories

Repository URL ?

Credentials ?

Branches to build ?

Branch Specifier (blank for 'any') X ?

- 빌드를 언제 수행할 것인지 지정

빌드 유발

☐ Build after other projects are built

☐ Build periodically

☐ Poll SCM

?

?

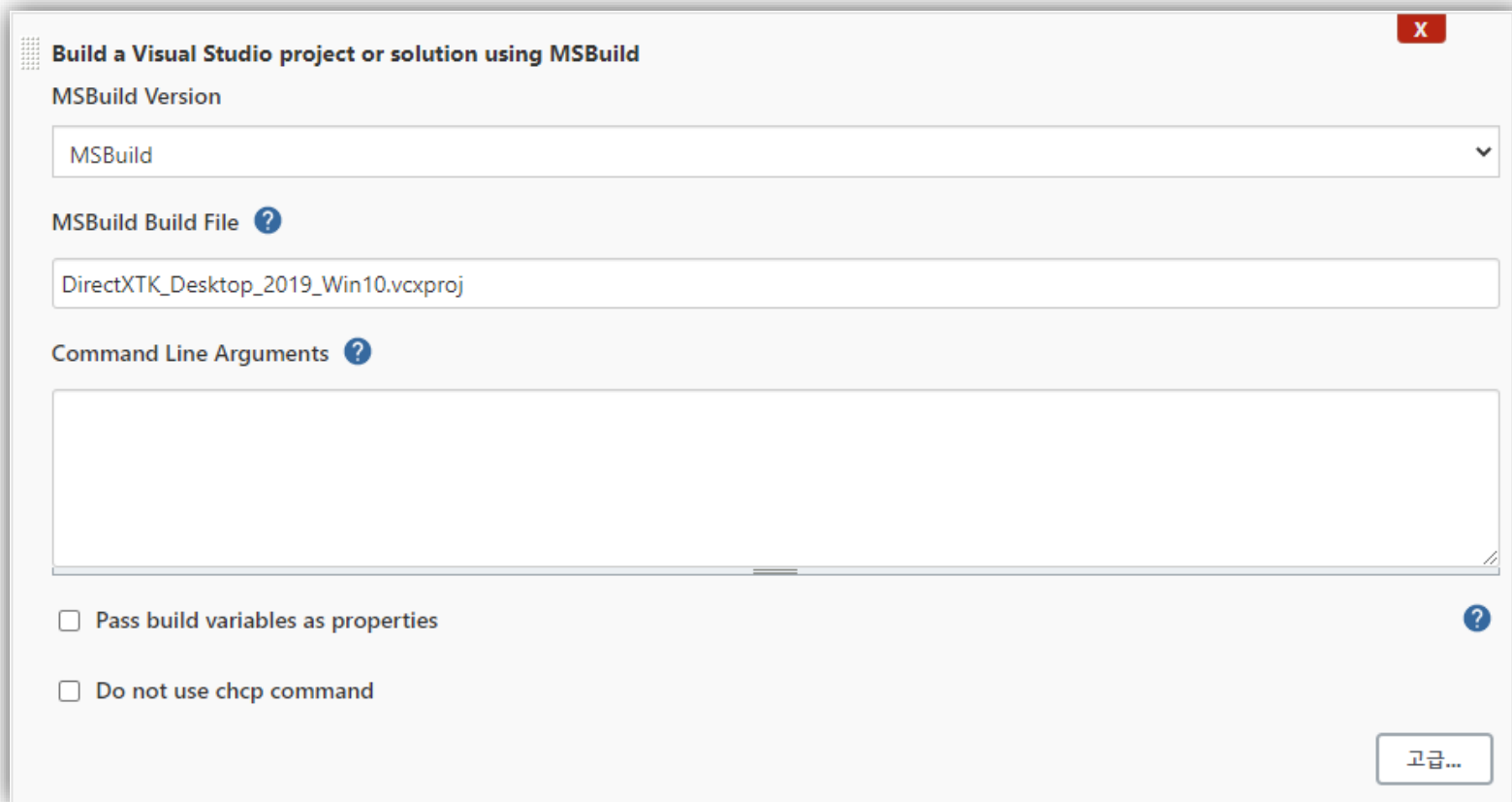
?

항목	설명
Build after other projects are built	다른 Job의 빌드가 완료되면 이 빌드를 시작한다. 예를 들어 Library란 Job의 빌드가 완료된 후 이 빌드를 시작하려면, 'Projects to watch' 항목에 Job 이름인 Library를 입력하면 된다.
Build periodically	정해진 일정에 따라 빌드를 진행한다. 일정 지정 형식은 cron 표기법과 유사하다.
Poll SCM	정해진 일정에 따라 소스 코드 저장소에서 변경 사항이 있는지 확인한다. 변경이 있을 경우 빌드를 진행한다. 일정 지정 형식은 cron 표기법과 유사하다.



- Build periodically과 Poll SCM의 일정 지정은 cron 표기법을 사용
  - TAB이나 공백으로 구분된 5자리
    - 분 시 일 월 요일순
  - 예) 매일 새벽 5시에 빌드를 진행
    - 0 5 \* \* \*
  - 예) 15분 단위로 소스 코드 저장소의 변경 사항을 확인
    - H/15 \* \* \* \*
  - cron 표기법 참고
    - <https://en.wikipedia.org/wiki/Cron>

- Build 에서 'MSBuild' 선택
- Build File에 프로젝트 파일 지정



Build a Visual Studio project or solution using MSBuild

MSBuild Version

MSBuild

MSBuild Build File ?

DirectXTK\_Desktop\_2019\_Win10.vcxproj

Command Line Arguments ?

☐ Pass build variables as properties ?

☐ Do not use chcp command

고급...

- CPP Build Job(Item) 생성

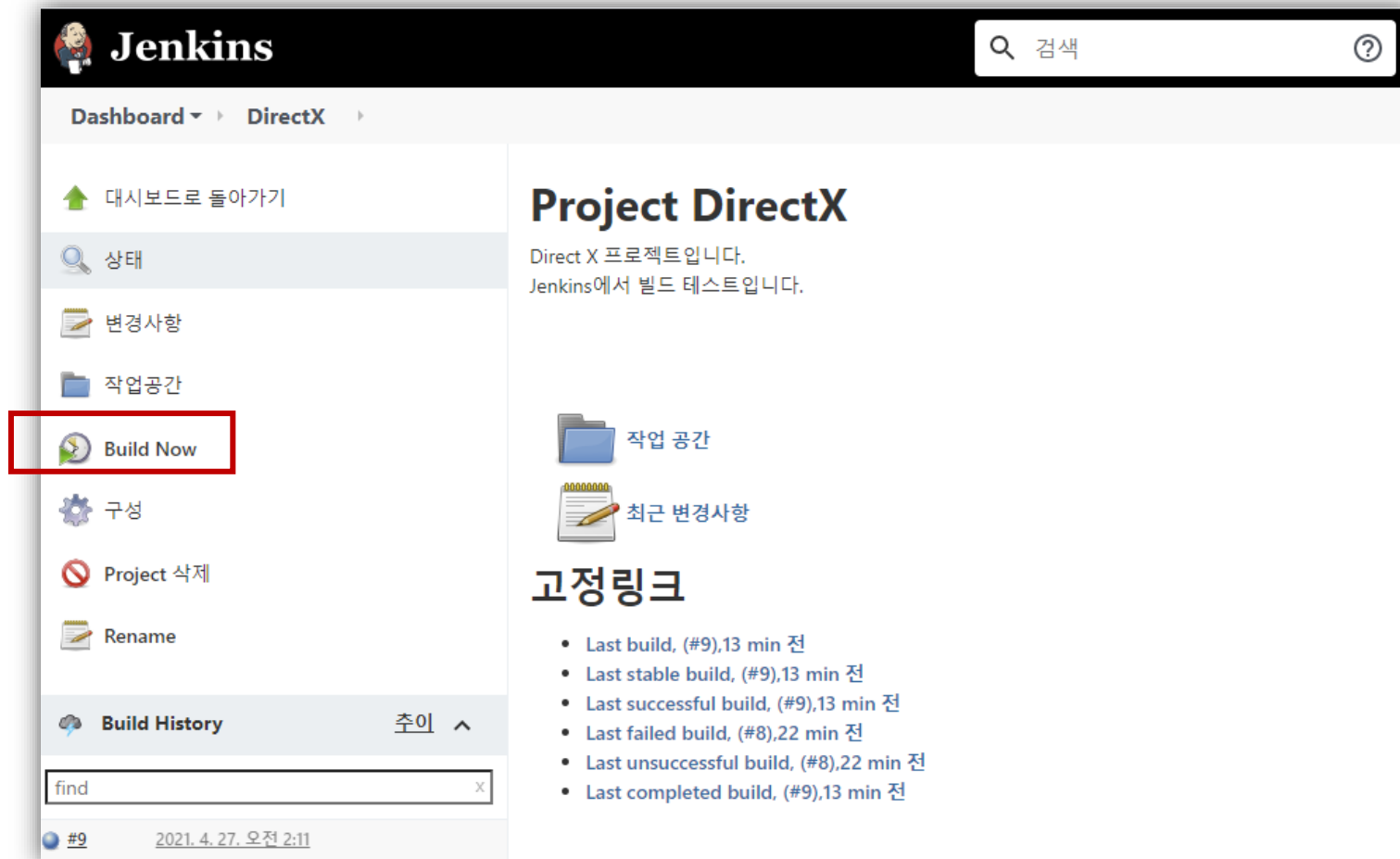
#PROFESSIONAL IT EDU-PLATFORM

code.presso()

CONTENTS / 015

# CPP 빌드를 위한 Item 생성 2

- 좌측 메뉴의 “Build Now” 클릭 시 바로 빌드 수행



- Console Output 을 통해 실시간 결과 확인 가능

```
Started by user admin
Running as SYSTEM
Building in workspace c:\jenkins\workspace\DirectX
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir c:\jenkins\workspace\DirectX\*.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/microsoft/DirectXTK12.git # timeout=10
Fetching upstream changes from https://github.com/microsoft/DirectXTK12.git
> git.exe --version # timeout=10
> git --version # 'git version 2.31.1.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/microsoft/DirectXTK12.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision fd27f5c1f61fc10ec25cec0e8b0560318ea3a358 (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f fd27f5c1f61fc10ec25cec0e8b0560318ea3a358 # timeout=10
Commit message: "Remove unused file"
> git.exe rev-list --no-walk fd27f5c1f61fc10ec25cec0e8b0560318ea3a358 # timeout=10
[DirectX] $ cmd /c call C:\windows\TEMP\jenkins5037040956938897937.bat

c:\jenkins\workspace\DirectX>"C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\MSBuild\Current\Bin\MSBuild"
DirectXTK_Desktop_2019_!\in10.vcxproj
Microsoft (R) Build Engine version 16.9.0+5e4b48a27 for .NET Framework
Copyright (C) Microsoft Corporation. All rights reserved.

Build started 4/27/2021 2:11:16 AM.
Project "c:\jenkins\workspace\DirectX\DirectXTK_Desktop_2019_!\in10.vcxproj" on node 1 (default targets).
ATGEnsureShaders:
  CompileShaders
```

- 빌드 수행
- 빌드 진행 실시간 확인

#PROFESSIONAL IT EDU-PLATFORM

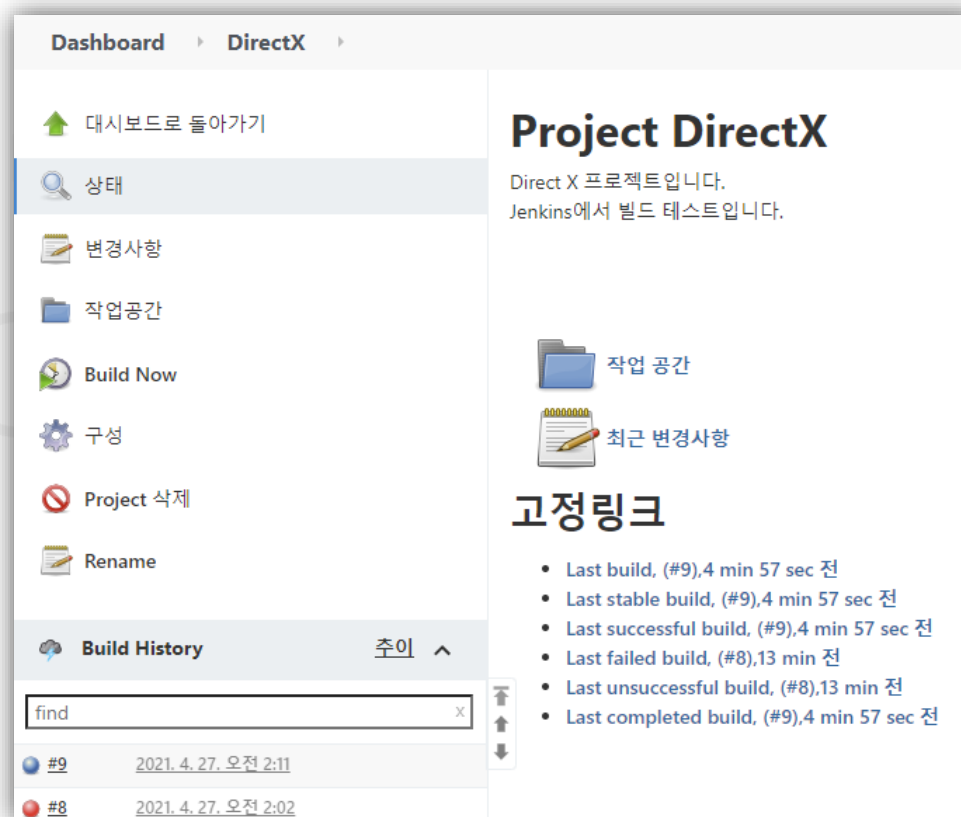
code.presso()

CONTENTS / 016

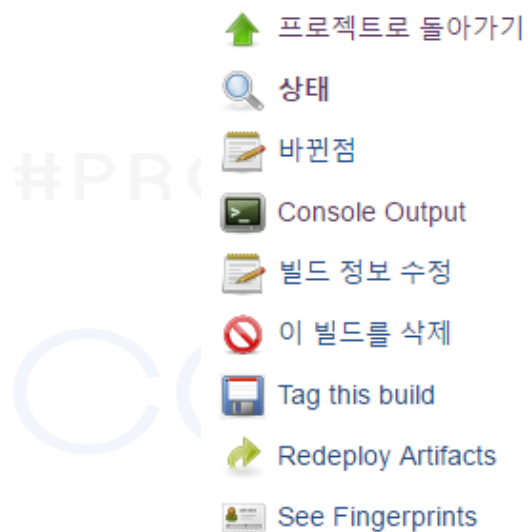
# Jenkins 빌드 결과 확인하기




- Jenkins 메인화면에서 각 Job을 클릭하면 Job 화면으로 이동
- Job이 정상적으로 빌드되었는지 결과만 확인하려면, 왼쪽 하단의 Build History를 확인
  - #1, #2 순으로 빌드가 표시
  - 파란색 공일 경우 빌드 성공
  - 빨간색 공일 경우 빌드 실패



- 각 빌드 #를 클릭하고, Console Output 확인



## 빌드 #1 (2016. 3. 29 오전 1:16:57)

 Revision: 3  
No changes.

 사용자 [관리자](#)에 의해 시작됨

### Module Builds

 [my\\_project](#) 1.6 sec

- 결과 확인

#PROFESSIONAL IT EDU-PLATFORM  
code.presso()

CONTENTS / 017

# CppCheck - 정적분석 도구 연동 1

## • 도구 개요

- 업계에서 "안좋은" 방식이라 알려진 코드의 발견
- 컴파일 에러는 아니지만, 나중에 문제를 발생시킬 수 있는 코드 탐지

## • 사용 방법

- 코딩하며, 컴파일 전에 사용

### [ Rule 예 ]

- Connection 열었으면 명시적으로 닫았는가?
- int i; 사용
- 매개변수 15개 사용
- case마다 brake; 있는가?
- final에서 return 금지
- 빈 if 문장은 피해야 함
- ...

※ 빨간색 표시: 보안 룰 검증 지원

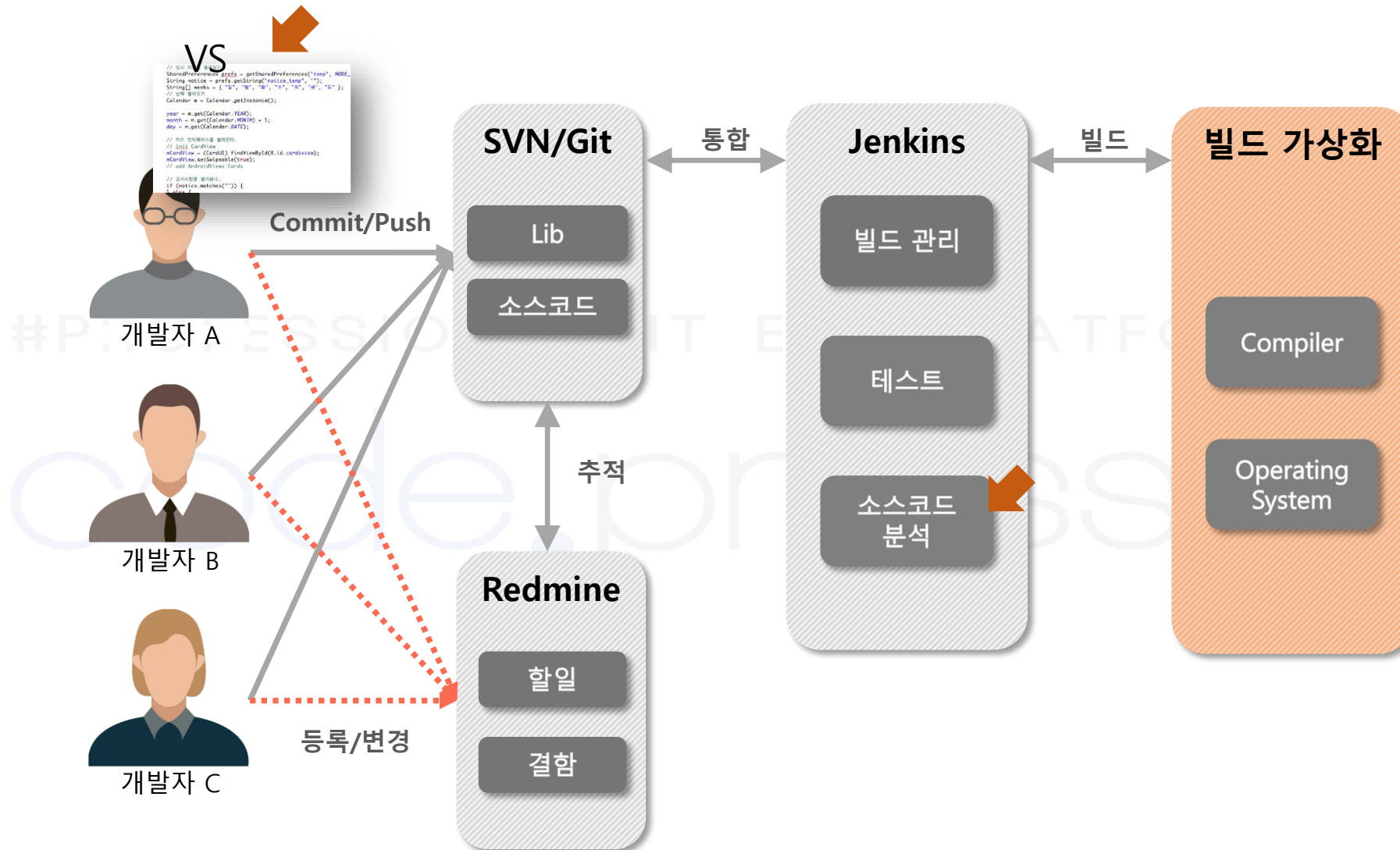
## • 대표 도구

언어	국산	외산	오픈소스	특징
<b>C/C++</b>	Sparrow(파수) CodeInspector(Suresoft) Resort(Soft4Soft)	<a href="#">QAC/QAF(PRQA)</a> <a href="#">Coverity(Synopsys)</a> <a href="#">Polyspace-Bug Finder(MathWorks)</a> <a href="#">Klockwork(Rogue Wave)</a>	CppCheck	<ul style="list-style-type: none"> <li>국산 도구는 행자부 개발 보안 검증</li> <li>ISO 26262 인증(오픈소스 제외)</li> </ul>
<b>Java</b>	Sparrow(파수) CodeInspector(Suresoft) Resort(Soft4Soft)	<a href="#">QAC/QAF(PRQA)</a> <a href="#">Coverity(Synopsys)</a>	PMD FindBug	<ul style="list-style-type: none"> <li>국산 도구는 행자부 개발 보안 검증</li> <li>보안의 경우 제외하고, 오픈소스 위주 사용</li> </ul>



## • 도구 선택 시 확인 사항

- 분석 대상 언어(C/C++ or JAVA)
- 1종, 2종 오류 발생률
  - 1종: 결함인데 못잡는 것
  - 2종: 결함이 아닌데 결함이라 하는 것



- 다운로드 위치

- <http://cppcheck.sourceforge.net/>

## Cppcheck

A tool for static C/C++ code analysis

[Home](#) | [Wiki](#) | [Forum](#) | [Issues](#) | [Developer Info](#) | [Online Demo](#) | [Project page](#)

[Download](#) | [Features](#) | [News](#) | [Documentation](#) | [Support](#) | [Contribute](#) | [Paying us to develop feature\(s\)](#)

**Cppcheck** is a [static analysis tool](#) for C/C++ code. It provides [unique code analysis](#) to detect bugs and focuses on detecting undefined behaviour and dangerous coding constructs. The goal is to have very few false positives. Cppcheck is designed to be able to analyze your C/C++ code even if it has non-standard syntax (common in embedded projects).

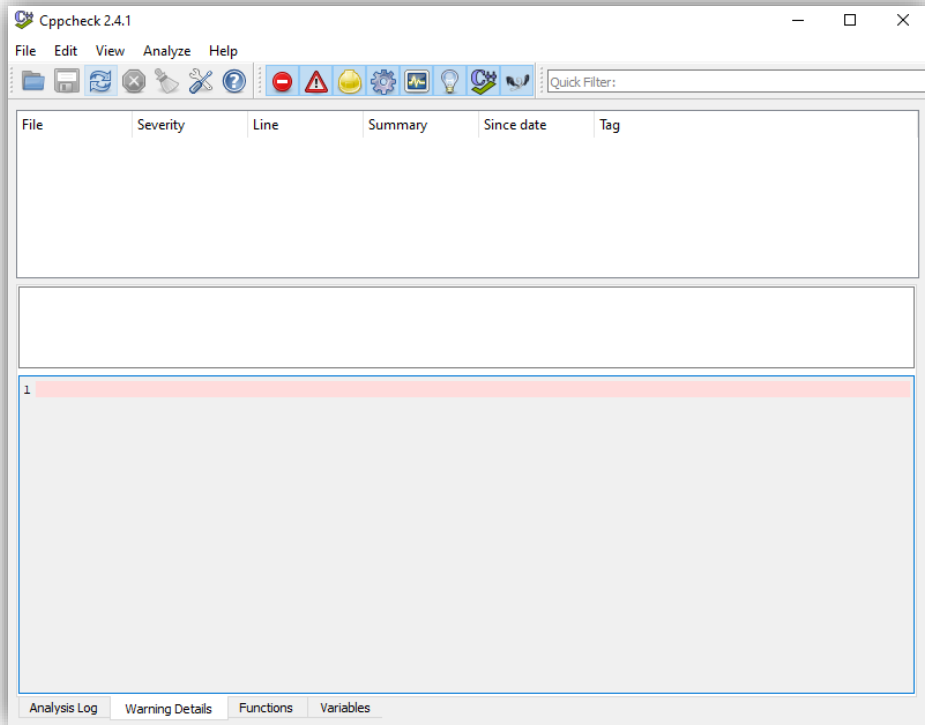
### Download

#### Cppcheck 2.4.1

Platform	File
Windows 64-bit (No XP support)	<a href="#">Installer</a>
Source code (.zip)	<a href="#">Archive</a>
Source code (.tar.gz)	<a href="#">Archive</a>



- UI로 사용할 수도 있으나, Jenkins에서는 커맨드라인으로 사용



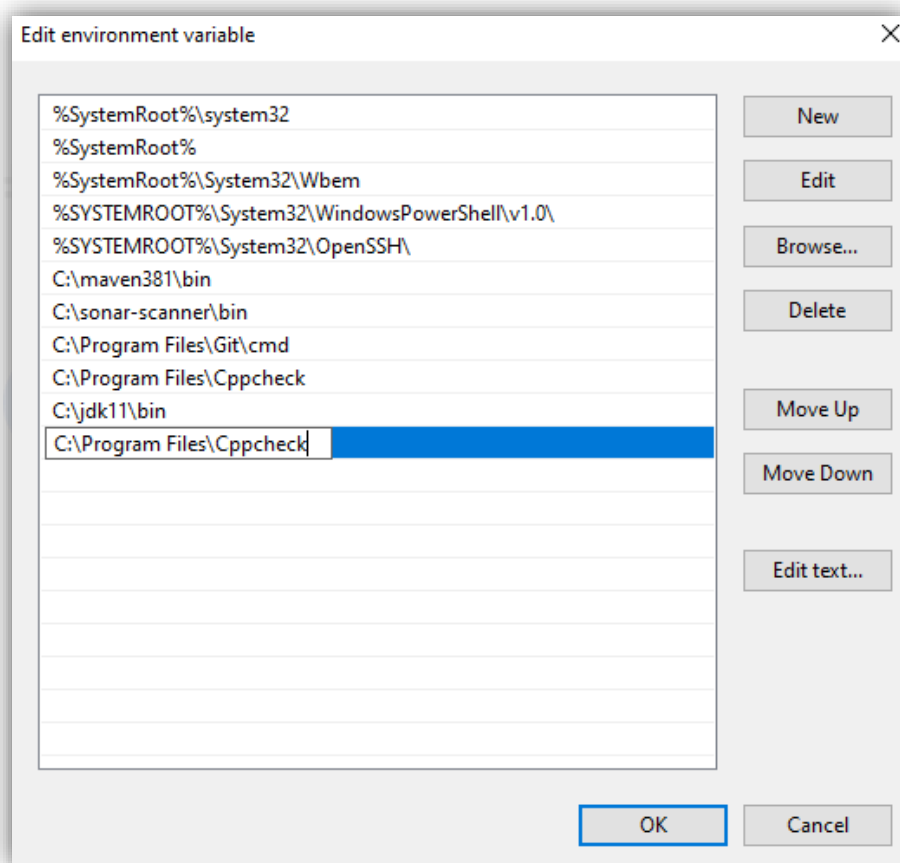
```
C:\Windows\system32\cmd.exe

C:\Program Files\Cppcheck>cppcheck --version
Cppcheck 2.4.1

C:\Program Files\Cppcheck>
```

code.presso()

- 쉬운 사용을 위해서라면, CppCheck 실행파일이 포함된 폴더를 환경변수 Path에 추가
  - 어느 위치에서도 CppCheck 실행하도록 설정



- `cppcheck --enable=all --inconclusive --xml --xml-version=2 src 2> cppcheck.xml`

```
C:\sourcecode\DirectXTK12>cppcheck --enable=all --inconclusive --xml --xml-version=2 src 2> cppcheck.xml
Checking Src\AlphaTestEffect.cpp ...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS2...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS2;NTDDI_WIN10_RS3...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS2;NTDDI_WIN10_RS3;NTDDI_WIN10_VB...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS2;NTDDI_WIN10_VB...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS3...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS5...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_RS5;NTDDI_WIN10_VB...
Checking Src\AlphaTestEffect.cpp: NTDDI_WIN10_VB...
Checking Src\AlphaTestEffect.cpp: WIN32...
Checking Src\AlphaTestEffect.cpp: _DEBUG...
Checking Src\AlphaTestEffect.cpp: _GAMING_XBOX_SCARLETT...
1/42 files checked 2% done
```

- IDE와 혼연일체가 되어야함

- Java 계열은 Eclipse, IntelliJ로 대동단결
  - 각 정적분석 도구는 이 2개 IDE 플러그인만 제공하면 됨
  - IDE가 아니더라도, Maven, Gradle 등 빌드 도구에 적용 가능(플러그인)
- C 계열은 컴파일러 자체 IDE 사용
  - 각 정적분석 도구가 각 IDE 별 플러그인 개발 필요 -> 플러그인이 거의 없음
  - 코딩을 위한 IDE 따로, 정적분석 도구 따로 사용

- 적용 룰을 최적화해야 함

- CppCheck의 모든 룰을 사용할 것인가? 필요한 룰만 사용할 것인가?
  - 필요한 룰은 어떻게 선정할 것인가?

- 적용 룰은 구현 전 배포 및 교육

- 최악: 코딩 후반부에 정적분석 도구 적용 및 강제화
- MISRA 적용 도구는 고가여서, 협력사에서 사용하기 어려우나 강제하는 경우 발생
  - CppCheck의 MISRA 룰셋 또는 SonarQube 사용 권장

- CI(Jenkins)에 적용은 일반화됨

- 거의 대부분의 유명 도구(상용/오픈소스 모두)는 Jenkins 플러그인 제공

- CppCheck 설치
- Path 설정
- 커맨드라인 실행 확인

#PROFESSIONAL IT EDU-PLATFORM

code.presso()

CONTENTS / 018

# CppCheck - 정적분석 도구 연동 2

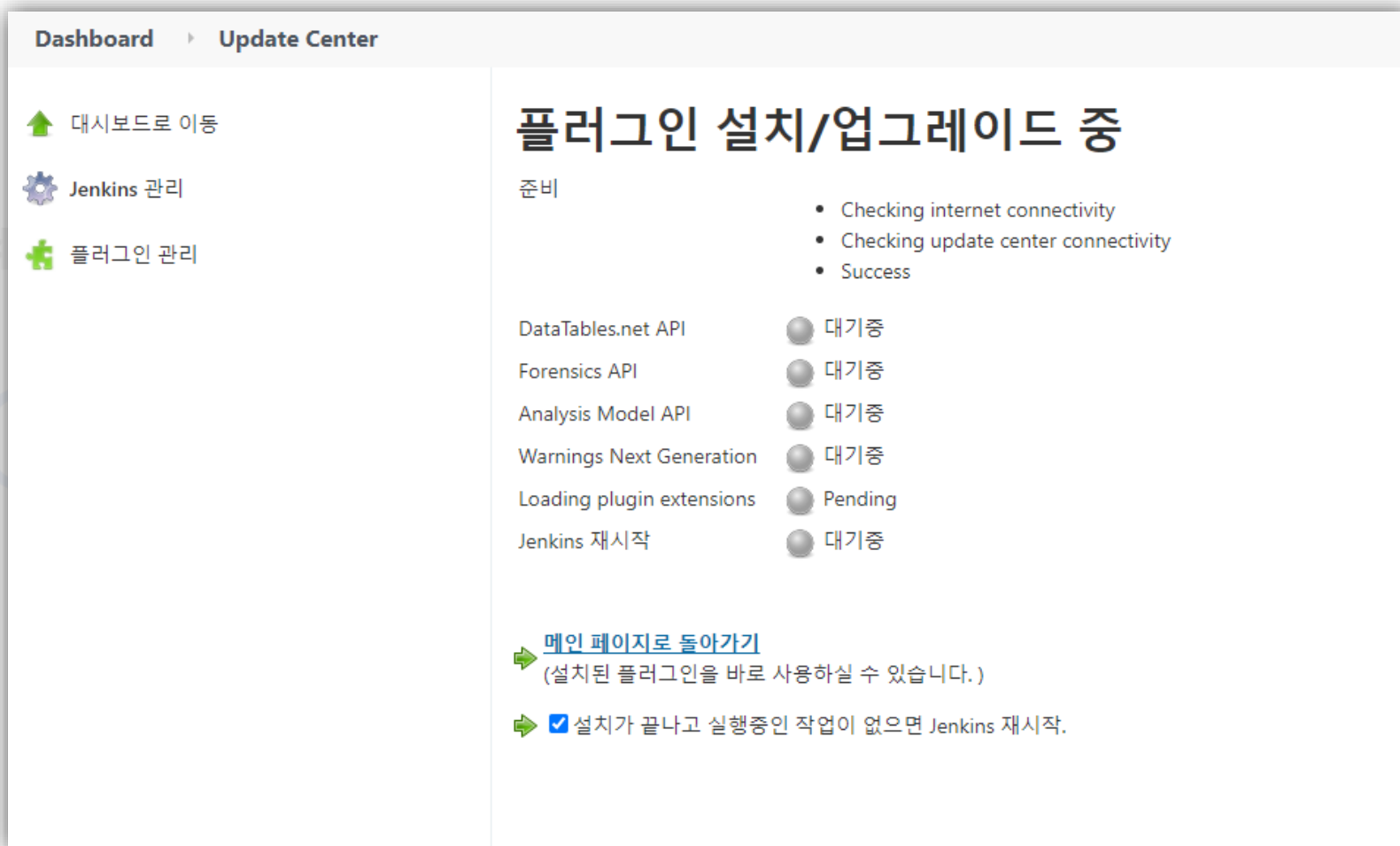
- 플러그인 설정에서 Warning Next Generation 검색
  - 설치 시 필요한 추가 플러그인을 자동 선택하여 설치함

The screenshot shows the CodePresto plugin marketplace interface. At the top, there is a search bar with the text 'warning next'. Below the search bar, there are tabs for '업데이트된 플러그인 목록', '설치 가능', '설치된 플러그인 목록', and '고급'. The '설치 가능' tab is selected. The main content area displays a list of plugins. The first plugin is 'Warnings Next Generation', which is highlighted. It has a checkbox to its left and a description: 'This plugin collects compiler warnings or issues reported by static analysis tools and visualizes the results. It has built-in support for numerous static analysis tools (including several compilers), see the list of [supported report formats](#).' To the right of the description, it shows the version '8.2.0' and the release time '8 days 12 hr ago'. Below this, there are several deprecated plugins: 'Checkstyle', 'Warnings', 'FindBugs', and 'PMD'. Each of these deprecated plugins has a checkbox, a description stating they have reached end-of-life and their functionality has been integrated into the 'Warnings Next Generation Plugin', and a version number and release time. A tooltip is visible over the 'FindBugs' plugin, showing the text 'functionality : 기능성, 상관성, 상관관계'.

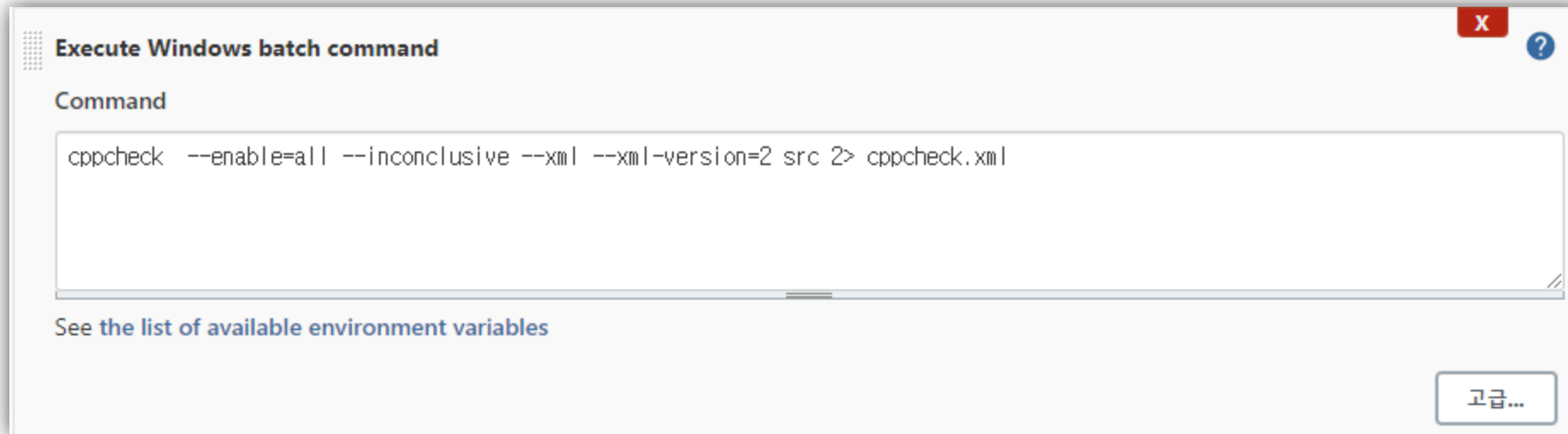
설치	이름	버전	Released
<input type="checkbox"/>	<b>Warnings Next Generation</b> This plugin collects compiler warnings or issues reported by static analysis tools and visualizes the results. It has built-in support for numerous static analysis tools (including several compilers), see the list of <a href="#">supported report formats</a> .	8.2.0	8 days 12 hr ago
<input type="checkbox"/>	<b>Checkstyle</b> deprecated Maven Build Reports The CheckStyle plug-in reached end-of-life. All functionality has been integrated into the <a href="#">Warnings Next Generation Plugin</a> .	4.0.0	1 yr 6 mo ago
<input type="checkbox"/>	<b>Warnings</b> deprecated Maven Build Reports The Warnings plug-in reached end-of-life. All functionality has been integrated into the <a href="#">Warnings Next Generation Plugin</a> .	5.0.1	1 yr 5 mo ago
<input type="checkbox"/>	<b>FindBugs</b> deprecated Maven Build Reports The FindBugs plug-in reached end-of-life. All functionality has been integrated into the <a href="#">Warnings Next Generation Plugin</a> .	5.0.0	1 yr 6 mo ago
<input type="checkbox"/>	<b>PMD</b> deprecated Maven Build Reports The PMD plug-in reached end-of-life. All functionality has been integrated into the <a href="#">Warnings Next Generation Plugin</a> .	4.0.0	1 yr 6 mo ago



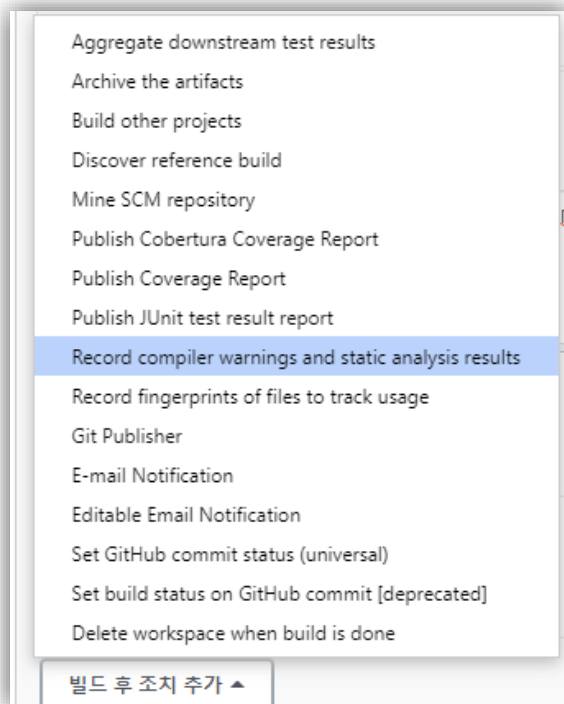
- “설치가 끝나고 실행중인 작업이 없으면 Jenkins 재시작” 선택



- Execute Windows Batch Command 를 선택하여, 명령어 입력



## • 보고서 생성 설정



**Record compiler warnings and static analysis results**

Static Analysis Tools

Tool ?

CPPCheck

Use options --xml --xml-version=2

Report File Pattern ?

\*\*/cppcheck.xml

Specifies the report files to scan for issues, such as '\*\*/target/checkstyle-results.xml'. If you leave this field blank, then the console log will be scanned for issues.

☐ Skip symbolic links when searching for files ?

Report Encoding ?

Encoding of your report files.

Custom ID ?

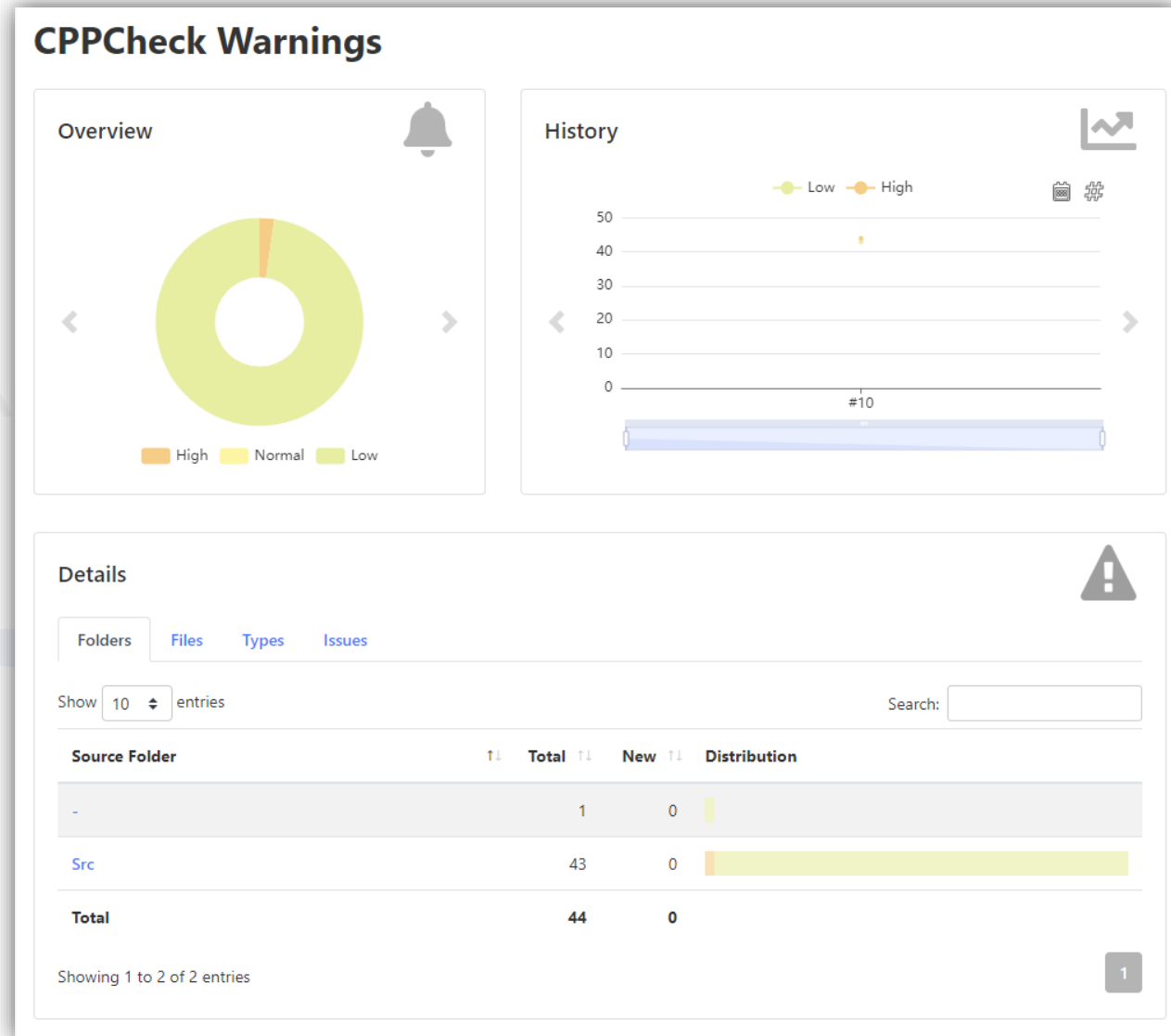
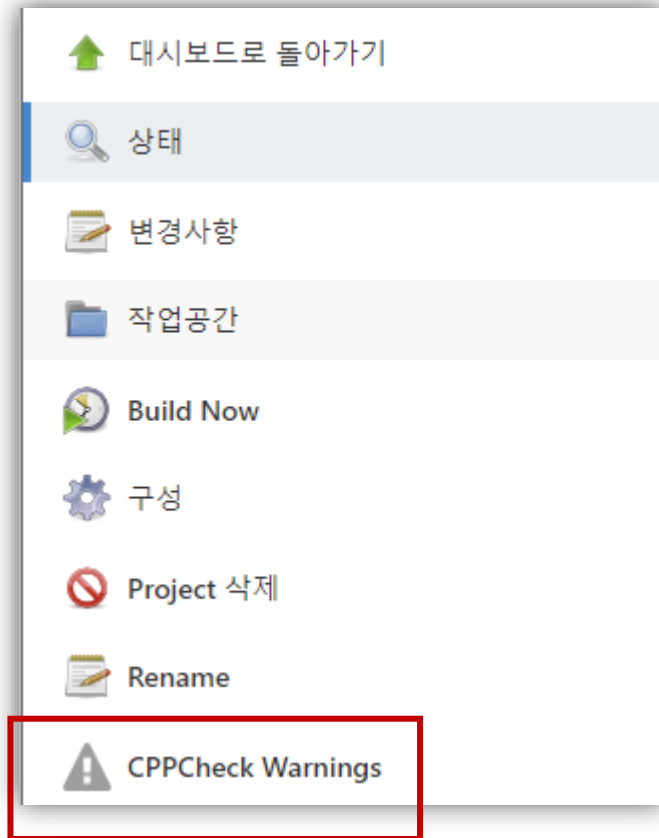
Optional custom ID (URL) of this tool, overwrites the built-in ID 'cppcheck'.

Custom Name ?

Optional custom display name of the tool, overwrites the built-in name 'CPPCheck'.

Add Tool

고급...



## Content of file BasicPostProcess.cpp



Too many #ifdef configurations - cppcheck only checks 12 of 22 configurations. Use --force to check all configurations.. The checking of the file will be interrupted because there are too many #ifdef configurations. Checking of all #ifdef configurations can be forced by --force command line option or from GUI preferences. However that may increase the checking time.

```
//-----  
// File: BasicPostProcess.cpp  
//  
// Copyright (c) Microsoft Corporation.  
// Licensed under the MIT License.  
//  
// http://go.microsoft.com/fwlink/?LinkID=615561  
//-----  
  
#include "pch.h"  
#include "PostProcess.h"  
  
#include "AlignedNew.h"  
#include "CommonStates.h"  
#include "DemandCreate.h"  
#include "DirectXHelpers.h"  
#include "EffectPipelineStateDescription.h"  
#include "GraphicsMemory.h"  
#include "SharedResourcePool.h"
```

- CppCheck 실행 설정
- CppCheck 보고서 설정
- 빌드 실행
- 결과 확인

#PROFESSIONAL IT EDU-PLATFORM  
code.presso()

CONTENTS / 019

# Doxygen - 의존성 도구 연동 1

- 개요

- 함수, 변수의 호출관계를 분석하는 도구
- 도구가 추구하는 방향에 따라 패키지/클래스(파일)/함수 단위로 표현

- 목적

- 아키텍처 구조에서 서브 시스템 간의 의존이 적절한지 확인
  - 서브 시스템 레이어에서 각 레이어 간 호출 관계 (예. Autosar)
  - 서브 시스템 레벨에서 각 레벨 간 호출 관계 (자식과 부모 패키지)
  - 원형 의존성(상호 참조) 관계



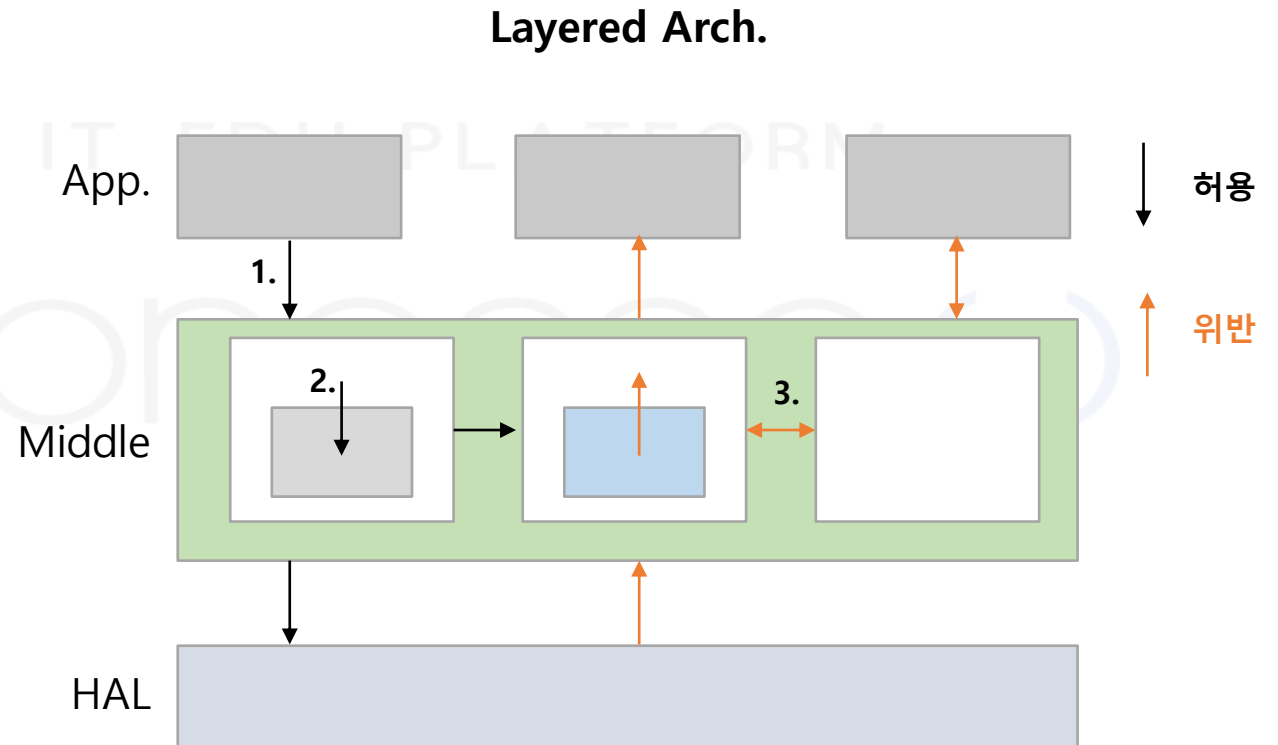
- 대표 도구

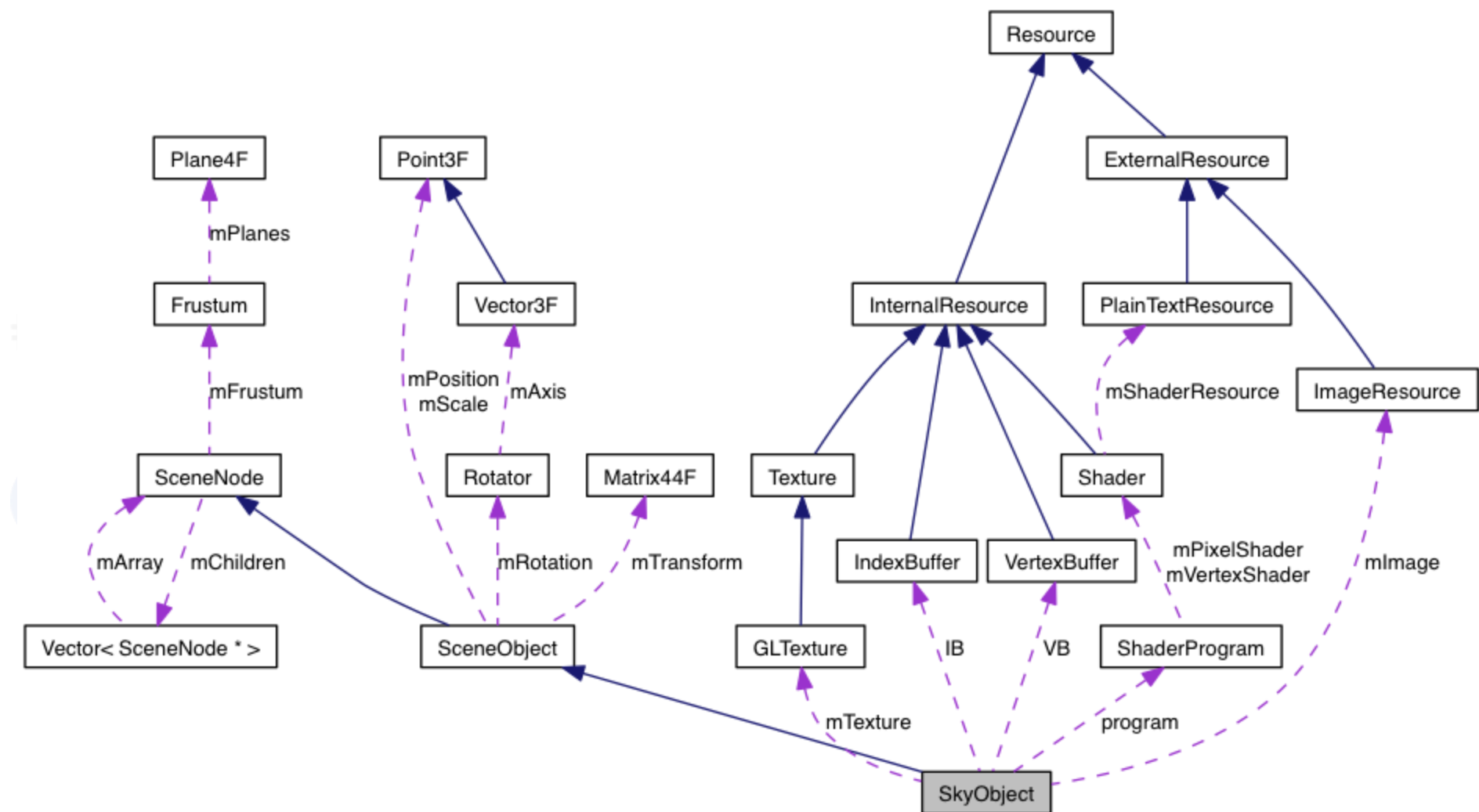
언어	외산		오픈소스
C/C++	Lattix Imagix4D	CppDepend	Doxygen
Java		JArchtect	JDepend Doxygen

- 의존성 분석 표현 종류

- DSM: Dependency Structure Matrix로, X/Y축의 형태로 서브 시스템 간의 의존성 확인에 유리
- 지표: 호출 대상을 지표로 표현
  - 특히, 객체지향 관련 도구는 Robert C. Martin의 논문을 근거로, 의존성 관련 지표를 정의하고 숫자로 표현
- 다이어그램: 호출 관계를 다이어그램으로 표현

1. 각 레이어 간 호출 관계 (예. Autosar)
2. 동일 레벨 간 호출 관계 (자식과 부모 패키지)
3. 원형 의존성(상호 참조) 관계





- 개요

- 소스코드의 의존 관계를 분석하는 오픈소스 도구
- Diagram으로 관계를 표시하여, 시각적으로 판단 가능
- C/C++외 Java, PHP 등 지원

- Diagram 제작 도구 필요

- 기본 기능으로 Diagram을 생성하나, 전문 도구를 활용하는 것이 좋음
- Graphviz 설치 시, 이 도구를 활용하도록 설정할 수 있음

- 다운로드 위치

- <https://www.doxygen.nl/download.html>

## Sources and Binaries

### Latest release

The latest version of doxygen is 1.9.1 (release date January 8th 2021).

#### The **source** distribution

[doxygen-1.9.1.src.tar.gz](#) (4.9MB)

#### A binary distribution for **Linux x86-64**

[doxygen-1.9.1.linux.bin.tar.gz](#) (11.5MB)

Compiled using Ubuntu 20.04 and dynamically linked against libclang version 9. This archive includes the HTML version of the manual, and the GUI frontend compiled against Qt5.

#### A binary distribution for **Windows**. All versions of Windows since Vista are supported.

[doxygen-1.9.1-setup.exe](#) (47.1MB)

This is a self-installing archive that includes the HTML and compressed HTML versions of the manual and the GUI frontend. It bundles 32-bit and 64-bit versions of doxygen.exe, and will install the right one based on the OS.

If you are allergic to installers and GUIs, haven't sufficient bandwidth, or don't have administrator privileges you can also download the [32-bit doxygen binary in a zip](#) (18.8MB) or the [64-bit version](#) (22.6MB).

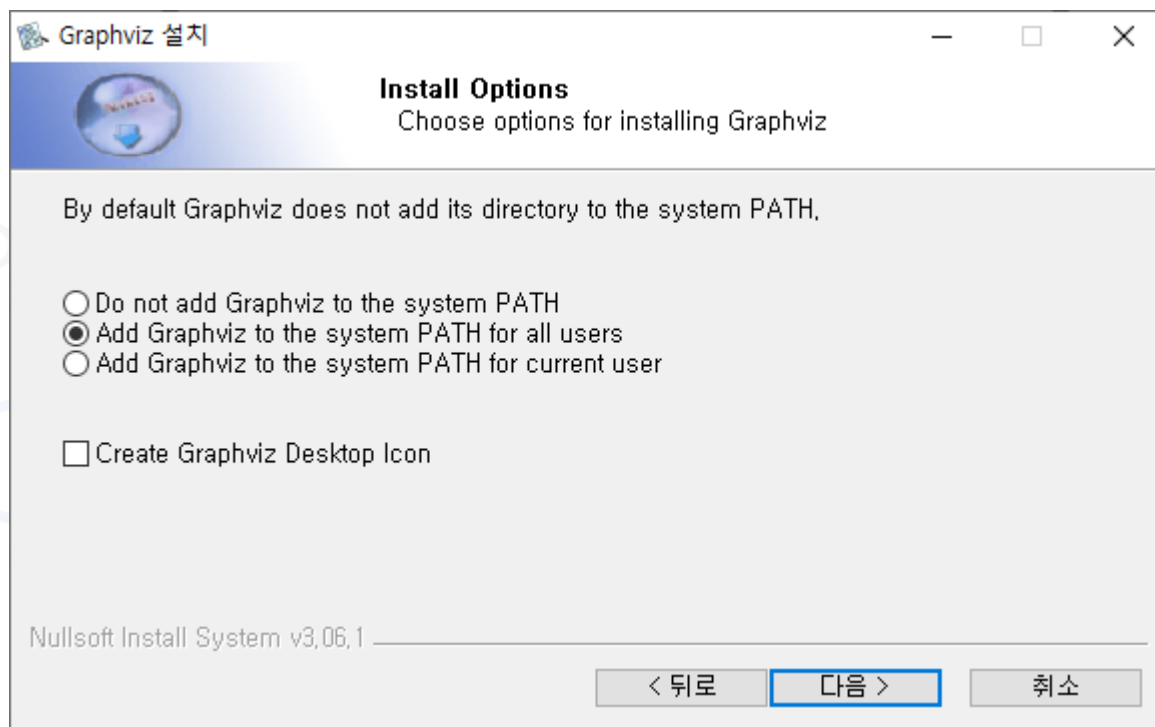
- 다운로드 위치

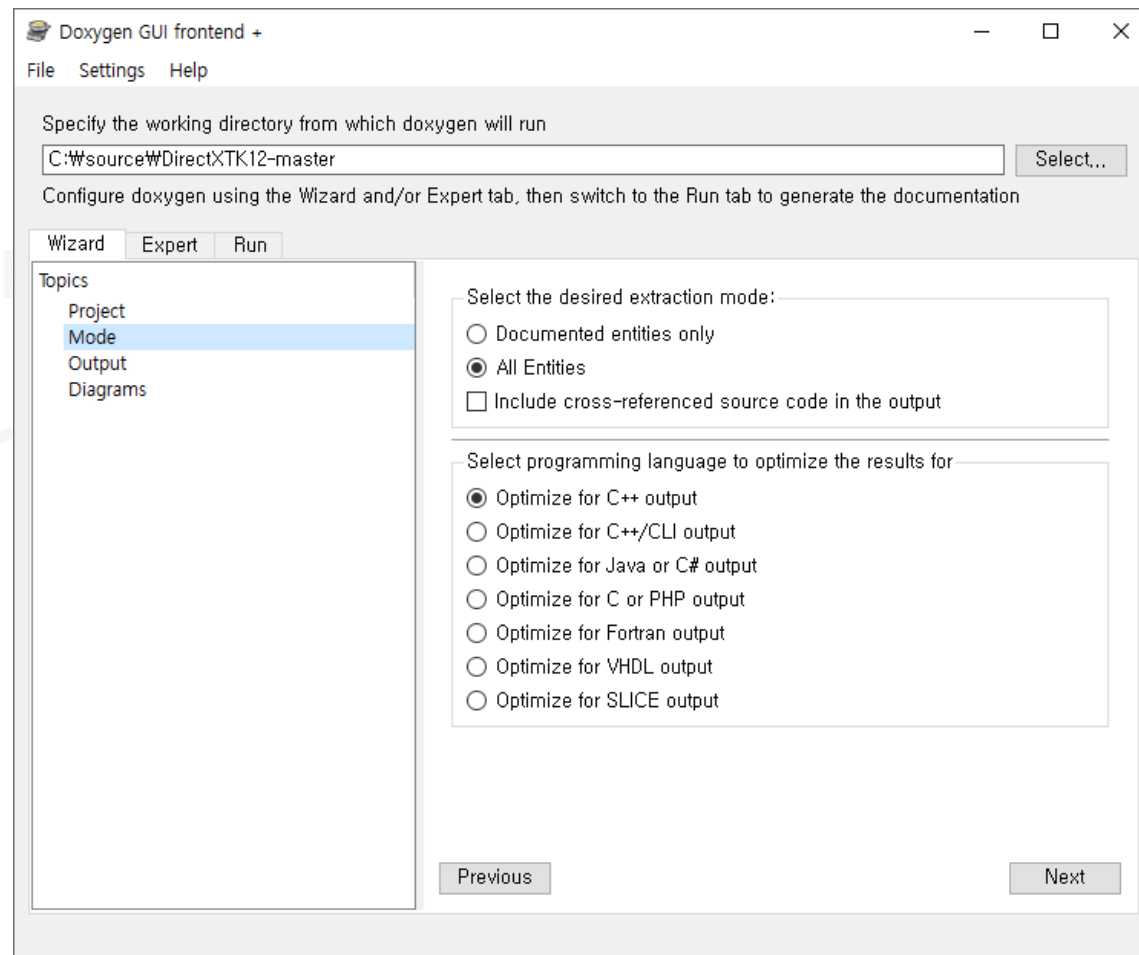
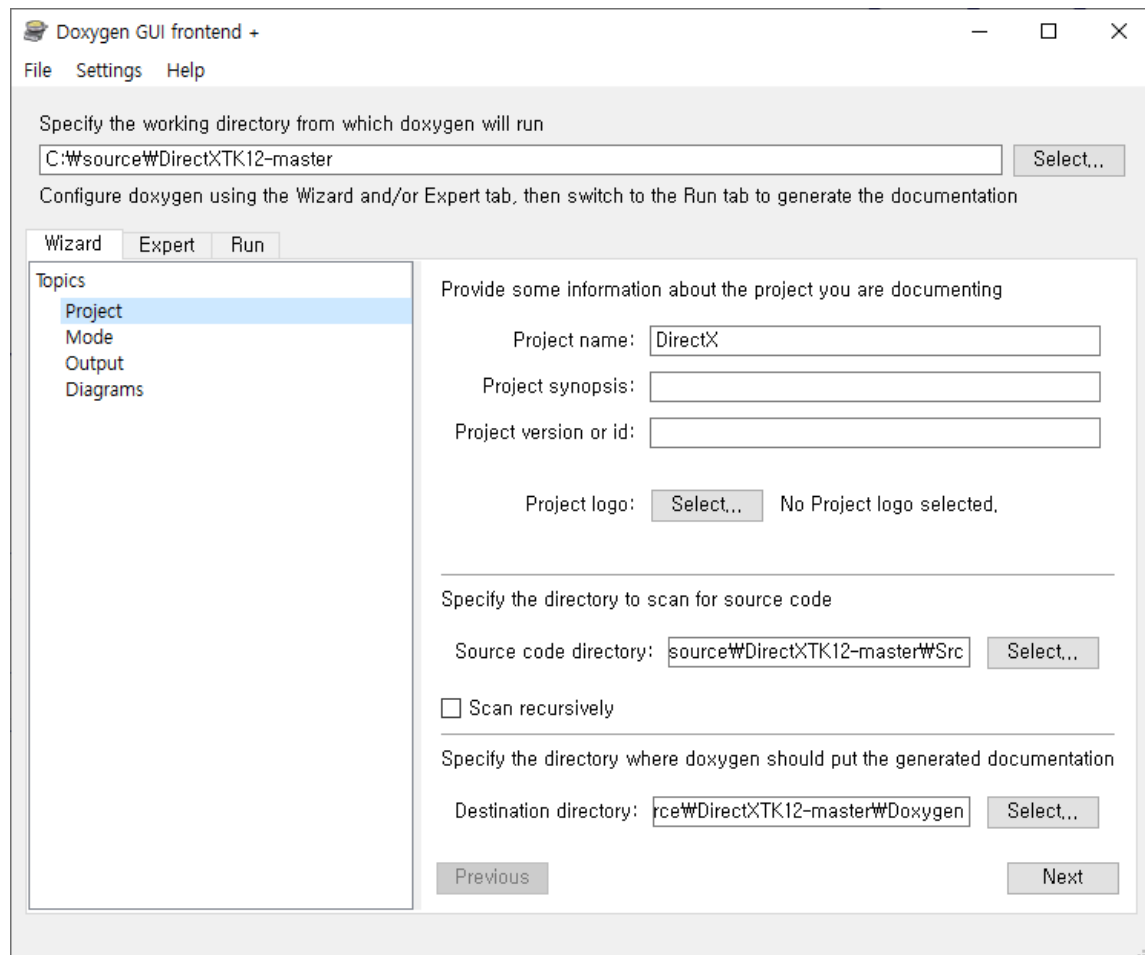
- <https://graphviz.org/download/>

## Windows

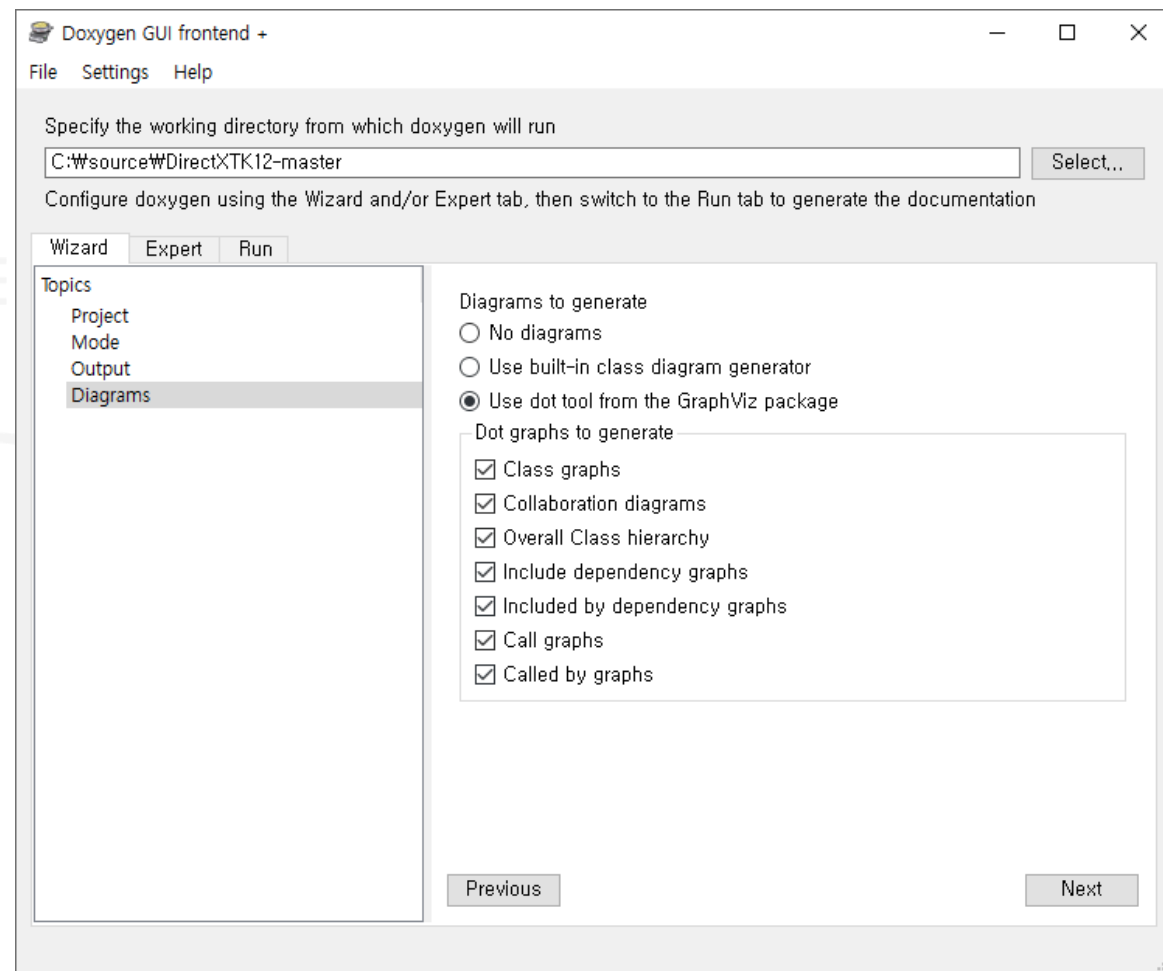
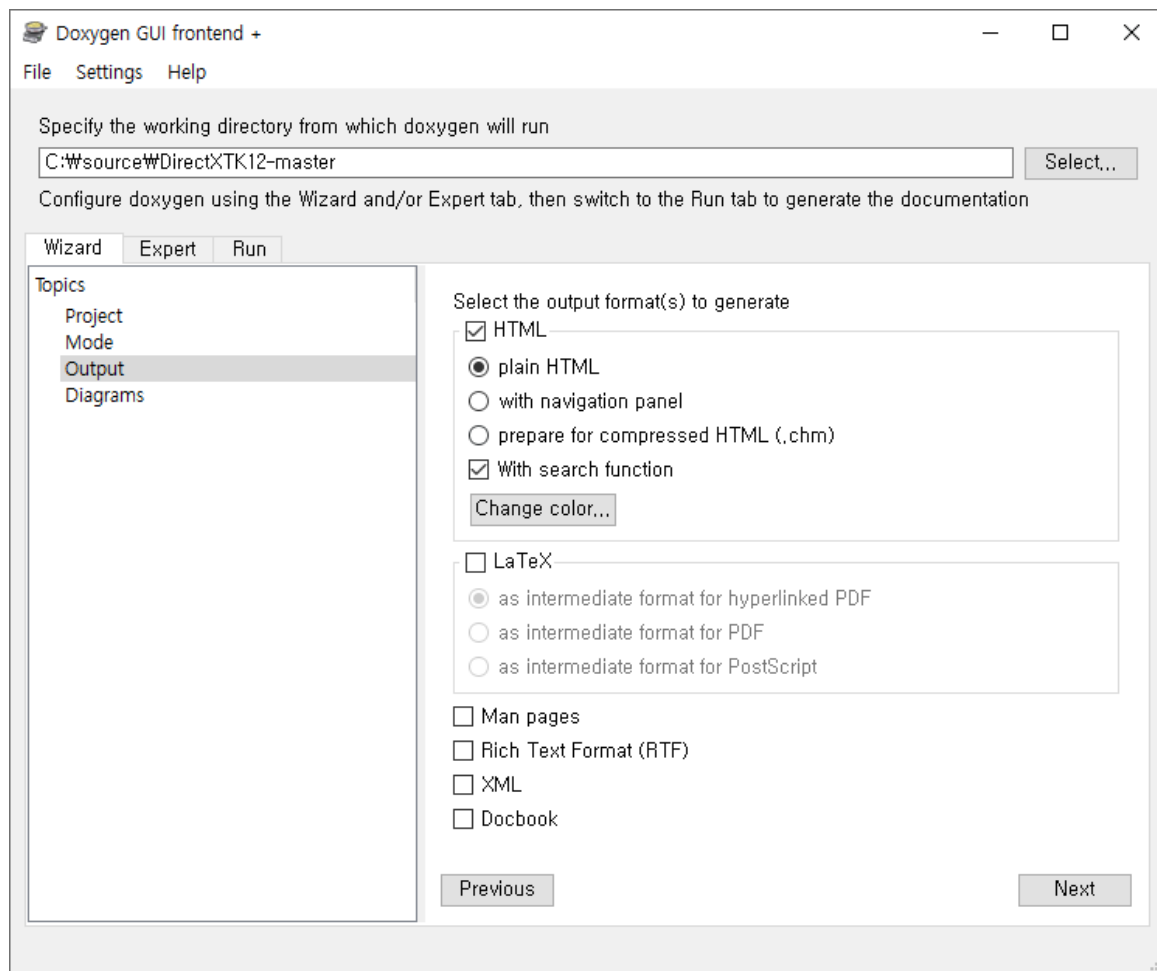
- Stable Windows install packages:
  - 2.47.1 EXE installer for Windows 10 (64-bit):  
[stable windows 10 cmake Release x64 graphviz-install-2.47.1-win64.exe](#) (not all tools and libraries are included)
  - 2.47.1 EXE installer for Windows 10 (32-bit):  
[stable windows 10 cmake Release Win32 graphviz-install-2.47.1-win32.exe](#) (not all tools and libraries are included)
  - 2.47.1 ZIP archive for Windows 10 (32-bit):  
[stable windows 10 msbuild Release Win32 graphviz-2.47.1-win32.zip](#)
  - checksums: [stable windows 10 cmake Release x64 graphviz-install-2.47.1-win64.exe.sha256](#) | [stable windows 10 cmake Release Win32 graphviz-install-2.47.1-win32.exe.sha256](#) | [stable windows 10 msbuild Release Win32 graphviz-2.47.1-win32.zip.sha256](#)
  - Further 2.47.1 variants available on [Gitlab](#)
  - [Prior to 2.46 series](#)

- Doxygen이 어느 위치에서나 Graphviz를 사용하기 위함

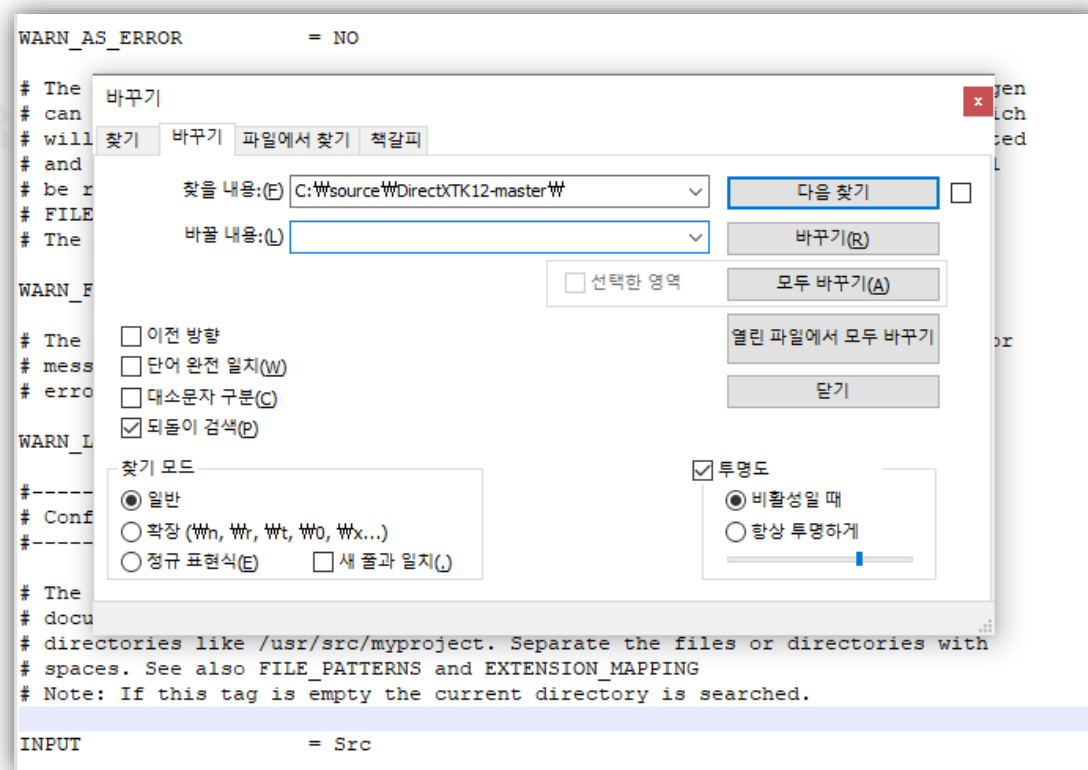


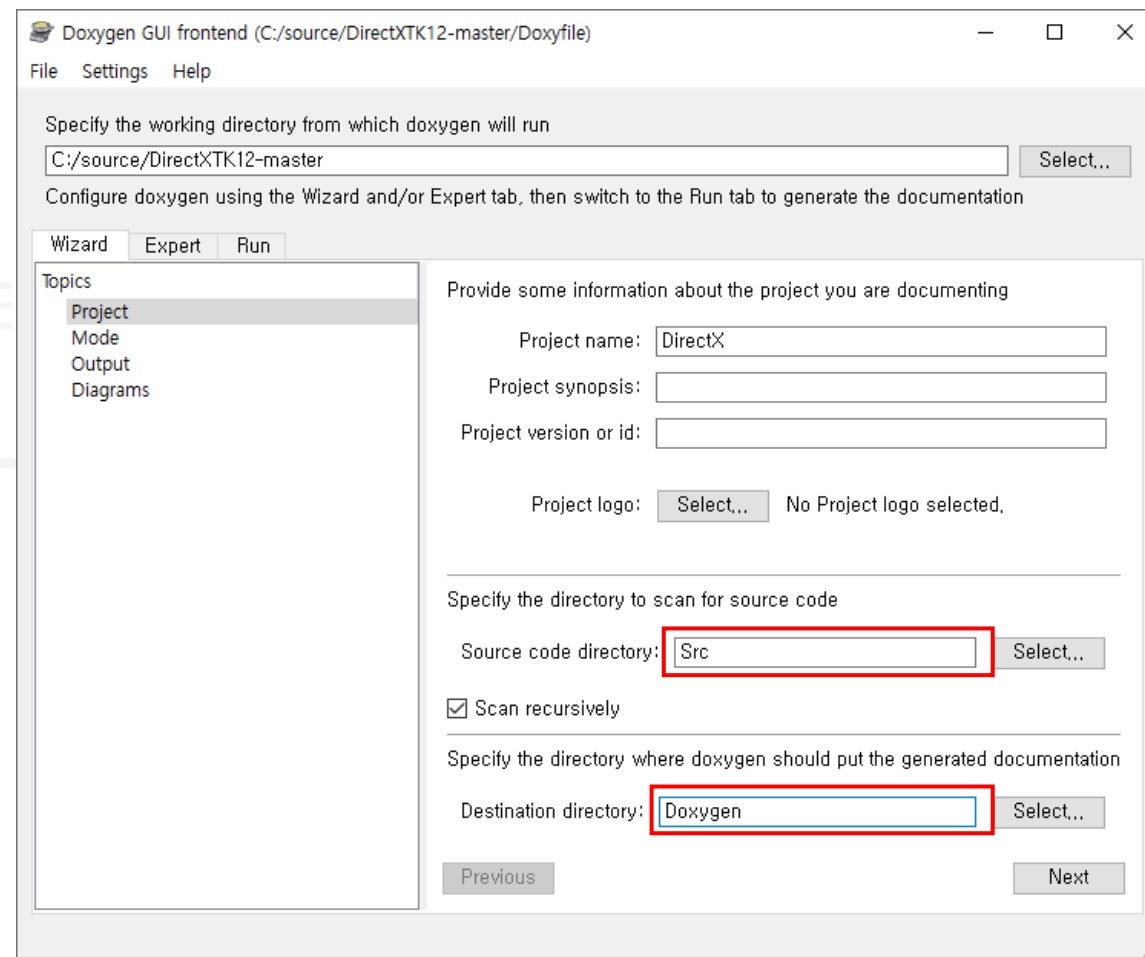
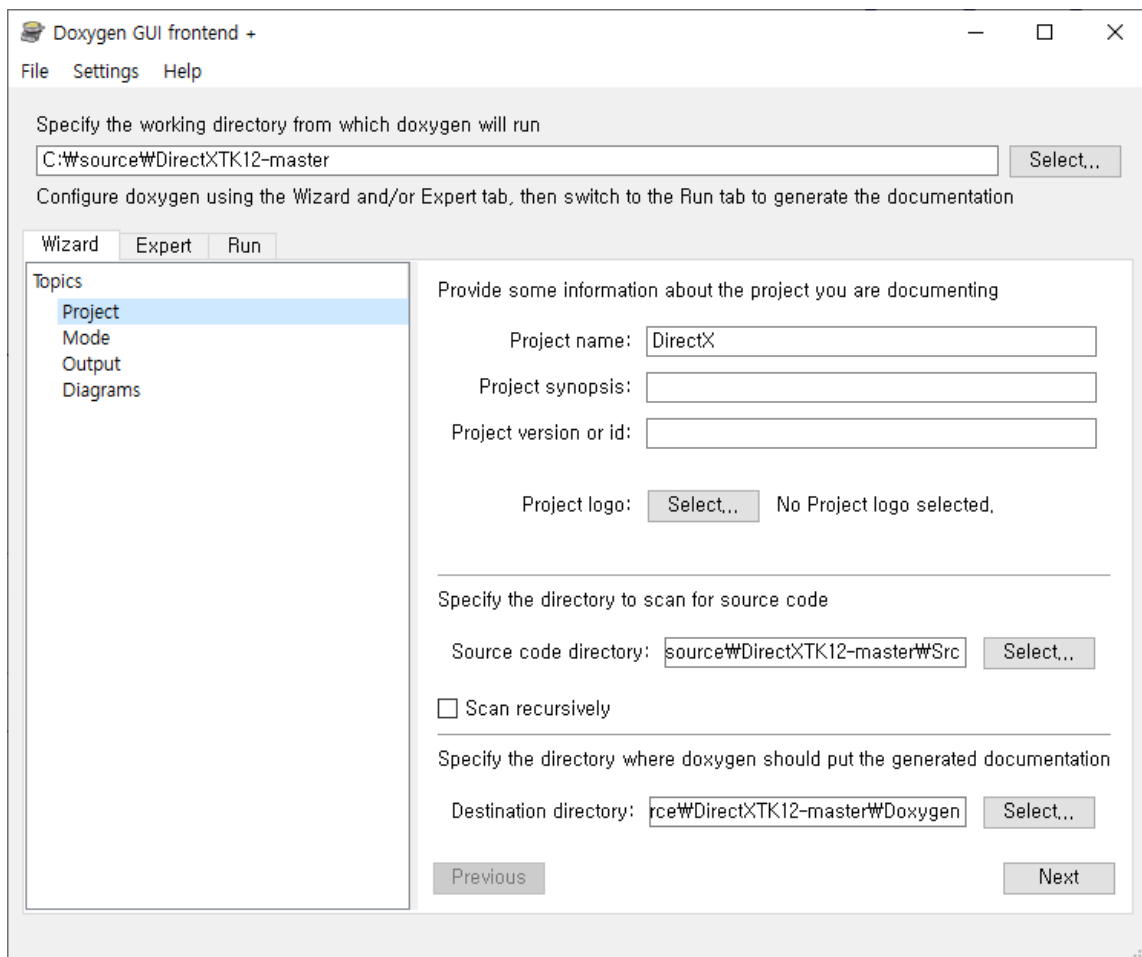






- Jenkins 및 모든 개발 참여자가 동일한 설정 파일을 사용하기 위함(정적분석 도구 동일)
  - 절대경로라면?
    - 모든 개발자가 자신의 설정 파일을 Git에 올리거나 개별로 가져야 함

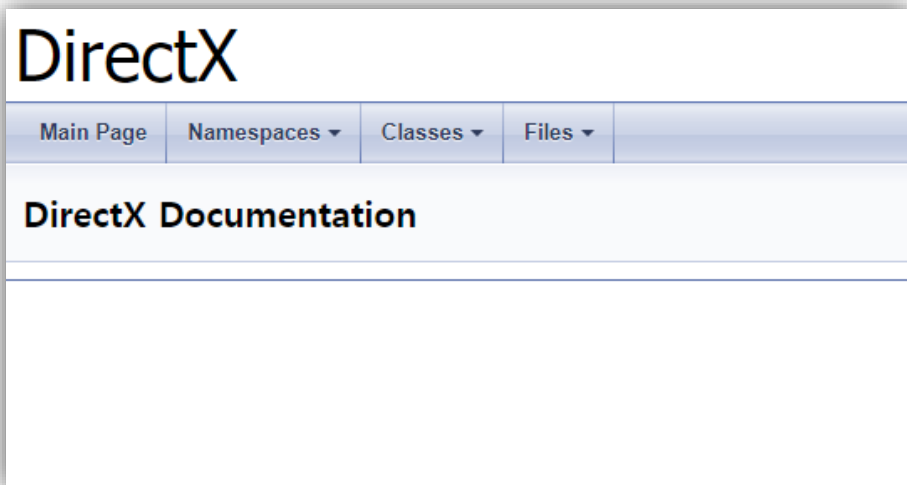




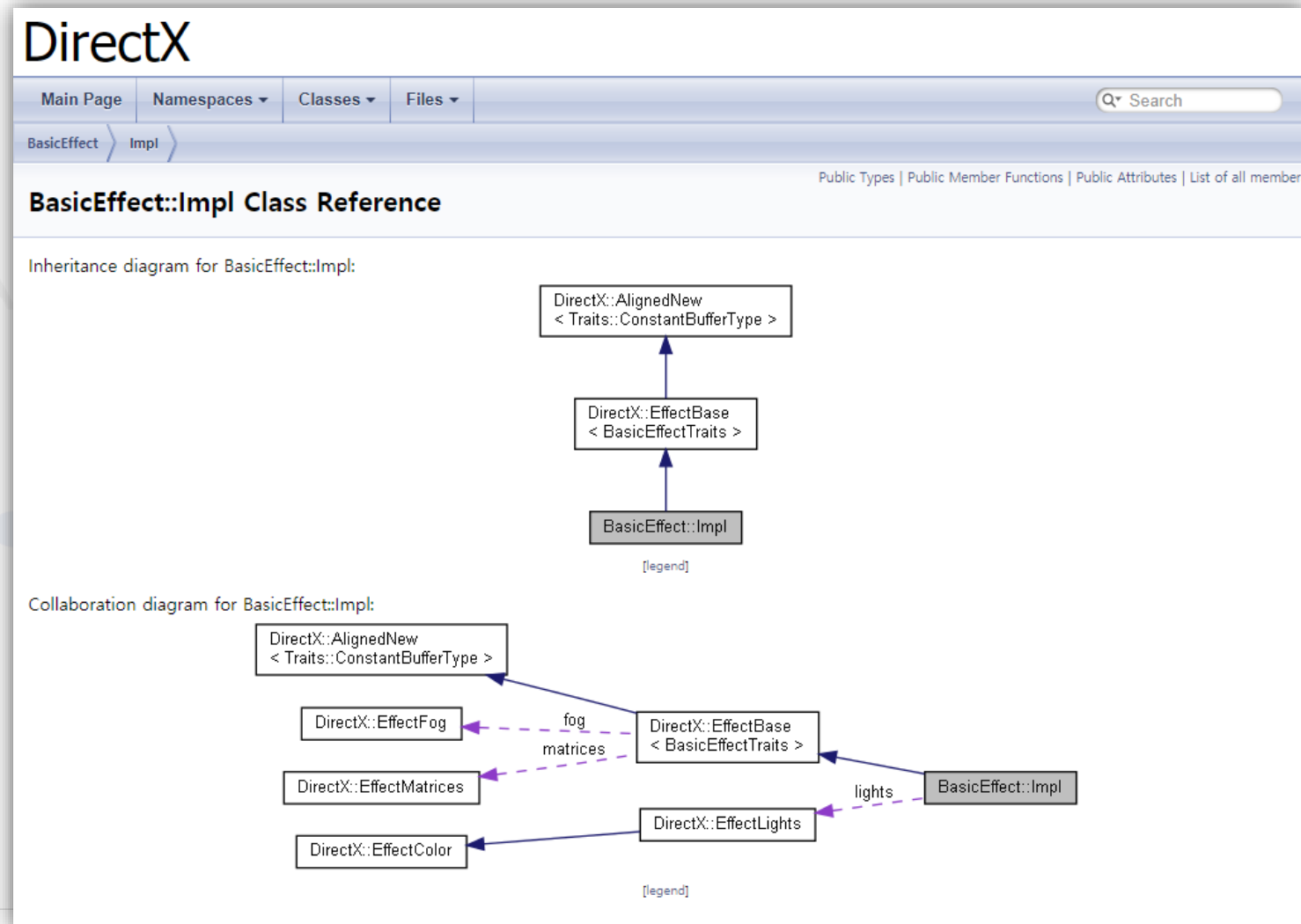
- "c:\Program Files\doxygen\bin\doxygen.exe" Doxyfile

```
Generating docs for namespace VBO
Generating docs for compound VBO::header_t...
Generating graph info page...
Generating directory documentation...
Generating index page...
Generating page index...
Generating module index...
Generating namespace index...
Generating namespace member index...
Generating annotated compound index...
Generating alphabetical compound index...
Generating hierarchical class index...
Generating graphical class hierarchy...
Generating member index...
Generating file index...
Generating file member index...
Generating example index...
finalizing index lists...
writing tag file...
Running plantuml with JAVA...
Running dot...
lookup cache used 3554/65536 hits=19165 misses=3860
finished...
```

[첫 화면]



[상세 화면]



- Doxygen, Graphviz 설치
- Doxyfile 생성
- 커맨드라인 실행

#PROFESSIONAL IT EDU-PLATFORM

code.presso()

CONTENTS / 020

# Doxygen - 의존성 도구 연동 2

- Doxyfile의 Git Commit/Push

- Jenkins와 모든 참여자가 동일한 Doxygen 결과를 생성하기 위함
- 이를 위해 "상대 경로" 사용

#PROFESSIONAL IT EDU-PLATFORM

code.presso()



- 플러그인 관리에서 doxygen 검색

The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and name are on the left, and a search bar with the text '검색' is on the right. Below the search bar, there are notification icons and the user name 'admin'. The main content area is titled 'Plugin Manager' and shows a search for 'doxygen'. The search results are displayed in a table with columns 'Install', 'Name', 'Version', and 'Released'. The 'Doxygen' plugin is listed with version '0.18' and released '5 yr 5 mo ago'. A yellow warning box is present for this plugin, stating 'This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.' At the bottom of the table, there are buttons for 'Install without restart' and 'Download now and install after restart', along with the text 'Update information obtained: 8 min 44 sec ago' and a '지금 확인' button.

Install	Name	Version	Released
<input type="checkbox"/>	<b>Doxygen</b> Build Tools This plugin publishes the reports generated by the <a href="#">Doxygen</a> tool.	0.18	5 yr 5 mo ago

**This plugin is up for adoption!** We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

[Install without restart](#) [Download now and install after restart](#) Update information obtained: 8 min 44 sec ago [지금 확인](#)

- Doxygen bin 폴더 정보 입력
  - 노란색 경고가 표시되도 무시
  - doxygen.exe의 위치를 알려줘야 함

**Doxygen**

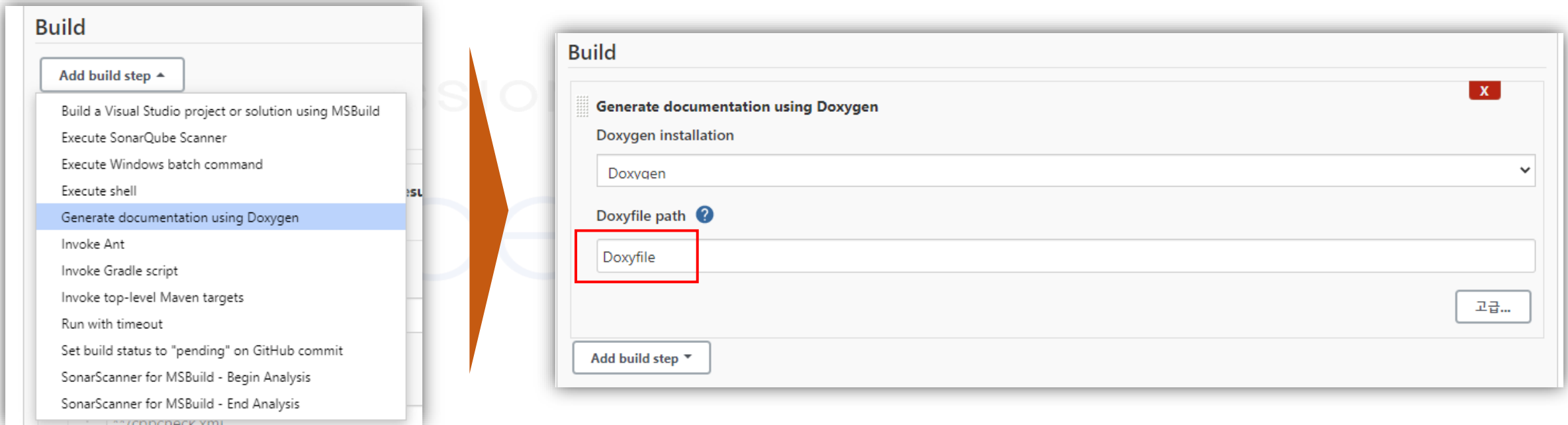
Doxygen installations

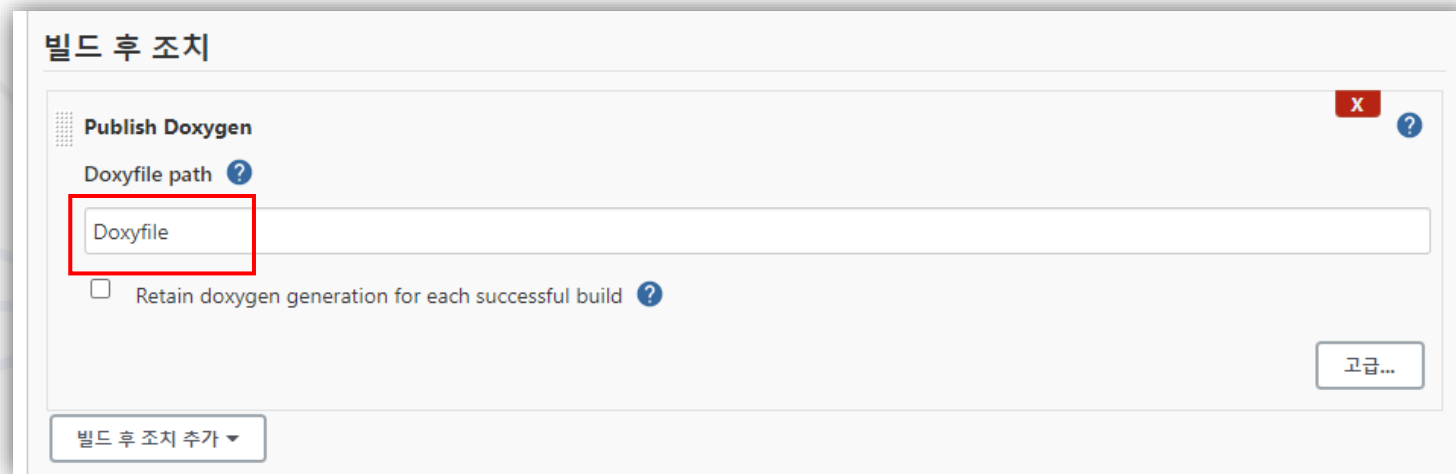
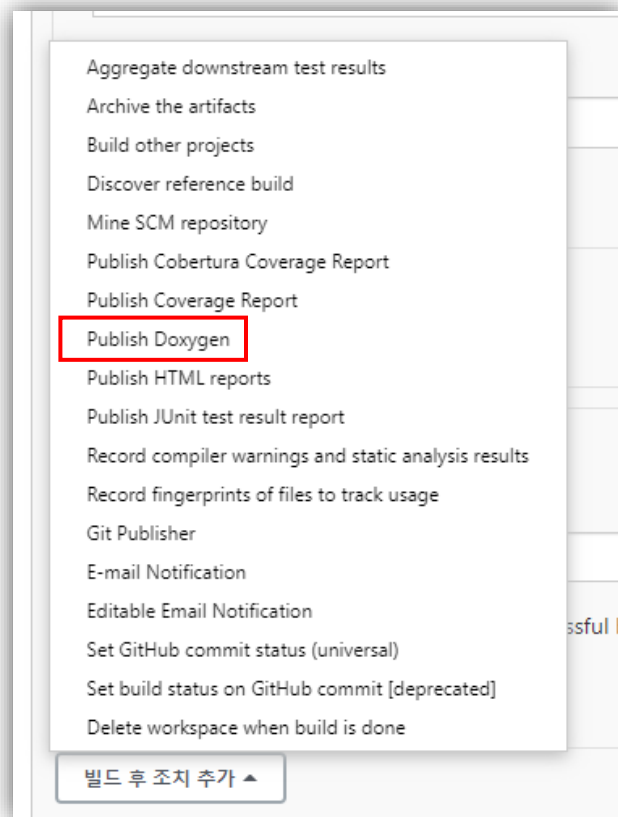
[Add Doxygen](#)

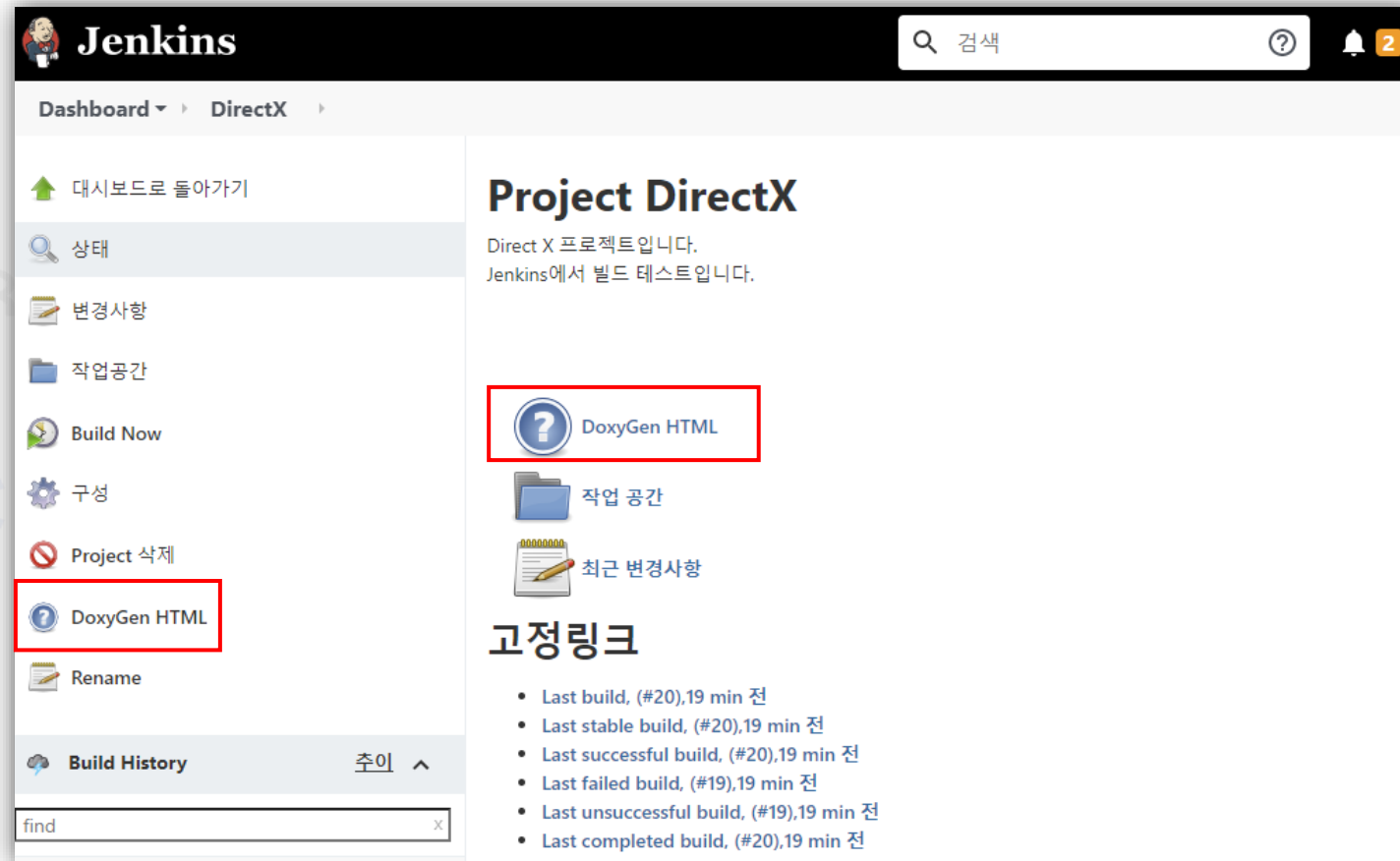
Doxygen
<b>Name</b> <span>?</span> <input type="text" value="Doxygen"/>
<b>Path to Doxygen</b> <span>?</span> <input type="text" value="C:\Program Files\doxygen\bin\doxygen.exe"/> <b>⚠ C:\Program Files\doxygen\bin\doxygen.exe is not a directory on the Jenkins master (but perhaps it exists on some agents)</b>
<input type="checkbox"/> <b>Install automatically</b> <span>?</span>

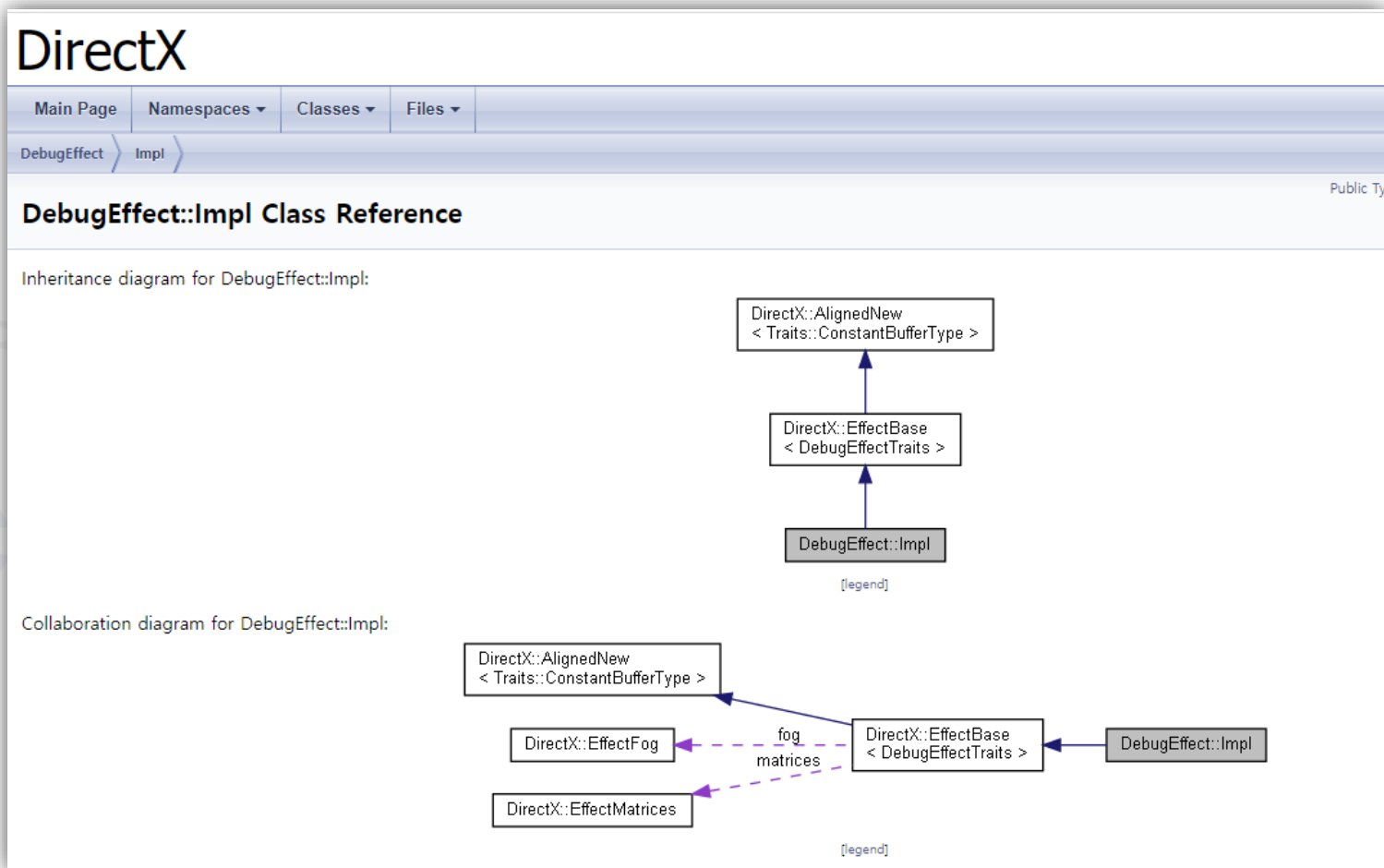
[Delete Doxygen](#)

- Build에서 Doxygen 실행 설정
  - Doxyfile Path는 상대경로에 따라 Doxyfile만 명시

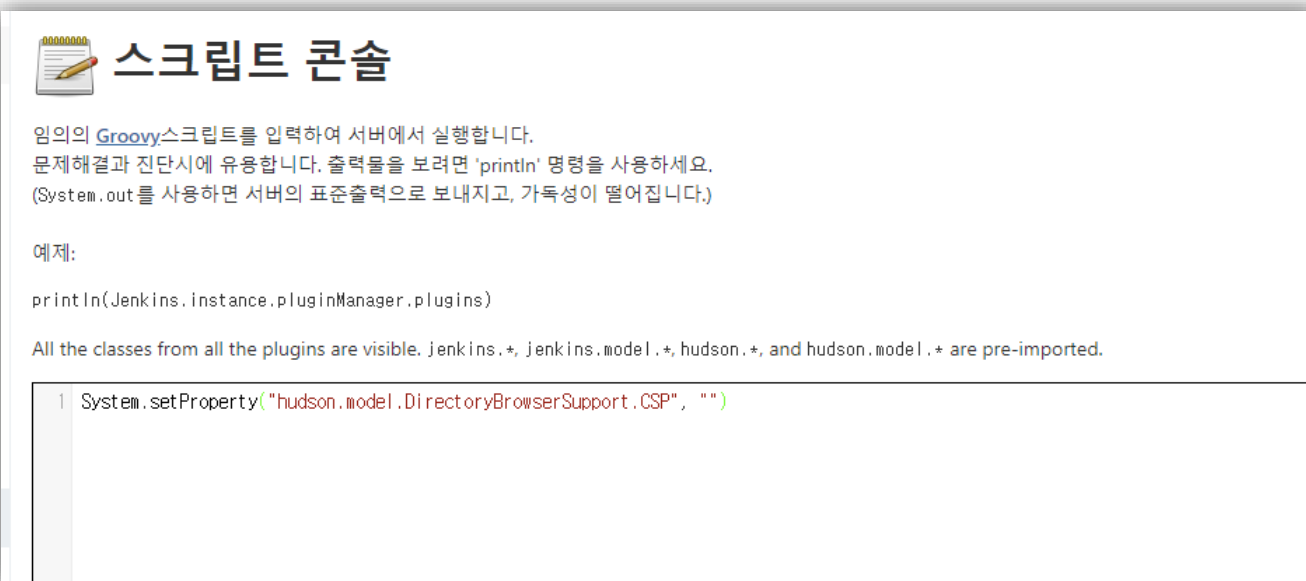








- Jenkins의 정책으로 CSS 파일 사용에 대한 제약(기본값)
- CSS 적용 기본 값 변경 방법
  - Jenkins 관리 -> Script Console 이동
  - 다음 명령 실행  
`System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")`



- Doxyfile Commit/Push
- Jenkins Item에 Doxygen 실행 설정
- 빌드
- Doxygen 결과 확인

#PROFESSIONAL IT EDU-PLATFORM  
code.presso()



CONTENTS / 021

# Lizard를 활용한 함수 라인수 및 복잡도 분석

- Lizard (1.17.7)

- 코드의 순환 복잡도를 분석하는 Python 기반의 도구 <Python이 설치되어야 함>
- 분석 지표: 순환 복잡도, 주석 제외 라인수, 함수 매개변수 수
- 홈페이지: <https://pypi.org/project/lizard/>
- 설치방법: pip로 설치
  - `pip install lizard`

- 언제 사용하는가?

- 복잡해서 이해도 및 유지보수성을 저해시키는 코드 확인 시

## • 사전 조건

- Python 설치 / pip 설치

## • 고려사항

- pip로 설치한 Lizard의 실행 위치 확인
- 필요 시 Path 추가 고려

```
C:\sourcecode\DirectX\TK12>C:\Users\dongjoonhan\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\Scripts\lizard.exe -C 10
=====
NLLOC  CCN  token  PARAM  length  location
=====
      8      2     63      0      8 EngineCallback::EngineCallback@30-37@.\\Audio\\AudioEngine.cpp
      1      1      5      0      1 EngineCallback::EngineCallback::STDMETHOD_( void , OnProcessingPassStart)@47-47@.\\Audio\\AudioEngine.cpp
      1      1      5      0      1 EngineCallback::EngineCallback::STDMETHOD_( void , OnProcessingPassEnd)@48-48@.\\Audio\\AudioEngine.cpp
      6      1     35      1      8 EngineCallback::EngineCallback::STDMETHOD_( void , OnCriticalError)@50-57@.\\Audio\\AudioEngine.cpp
      8      2     63      0      8 VoiceCallback::VoiceCallback@64-71@.\\Audio\\AudioEngine.cpp
      3      1      5      0      3 VoiceCallback::~VoiceCallback@79-81@.\\Audio\\AudioEngine.cpp
      1      1      6      1      1 VoiceCallback::VoiceCallback::STDMETHOD_( void , OnVoiceProcessingPassStart)@83-83@.\\Audio\\AudioEngine.cpp
      1      1      5      0      1 VoiceCallback::VoiceCallback::STDMETHOD_( void , OnVoiceProcessingPassEnd)@84-84@.\\Audio\\AudioEngine.cpp
      1      1      5      0      1 VoiceCallback::VoiceCallback::STDMETHOD_( void , OnStreamEnd)@85-85@.\\Audio\\AudioEngine.cpp
```

## • 명령줄 실행

- 순환 복잡도 10 이상을 분석
  - lizard.exe -C 10
- 순환 복잡도 10 이상을 분석, 엑셀에서 확인하기 위해 csv로 출력해서 저장
  - lizard.exe -C 10 --csv > lizard\_result.csv

		순환 복잡도				함수 정보	
	A	B	C	D	E	F	G
1		8	2	63	0	8	EngineCallback::EngineCall .#Audio#AudioEngine.cp
2		1	1	5	0	1	EngineCallback::EngineCall .#Audio#AudioEngine.cp
3		1	1	5	0	1	EngineCallback::EngineCall .#Audio#AudioEngine.cp
4		6	1	35	1	8	EngineCallback::EngineCall .#Audio#AudioEngine.cp
5		8	2	63	0	8	VoiceCallback::VoiceCallba .#Audio#AudioEngine.cp

- 실행 방법

- `lizard ./src -C 10 -L 80 --xml > lizard.xml`
- `"C:\Users\cypark\AppData\Local\Programs\Python\Python310\Scripts\lizard" ./src -C 10 -L 80 --xml > lizard.xml`

- 연동 방법

- CppNCSS 플러그인 설치
- CppNCSS 플러그인에 xml 파일 지정

## 빌드 후 조치

### ☰ Publish Cpp NCSS Report

#### CppNCSS xml report pattern

This is a file name pattern that can be used to locate the CppNCSS xml report files (for example with Maven2 use **\*\*/target/cppncss-raw-report.xml**). The path is relative to **the module root** unless you are using Subversion as SCM and have configured multiple modules, in which case it is relative to the workspace root.

CppNCSS must be configured to generate XML reports for this plugin to function.

#### Function CCN violation threshold

If the Cyclomatic Complexity Number of a function larger than this number, it will be counted on the trend chart.

CONTENTS / 022

# CPD를 이용한 중복 코드 분석

- CPD (6.34.0)

- 코드 내에 지정한 Token 이상 중복이 있는지 분석
- Java 룰 기반 정적 분석 도구인 PMD의 부가 도구. 단, 대부분의 개발 언어 지원
- 분석 지표: 중복 코드
- 홈페이지: <https://pmd.github.io/>
- 설치방법: 다운로드 후 압축 해제

- 언제 사용하는가?

- 중복 코드 분석 시
- 불필요한 재사용 확인 시

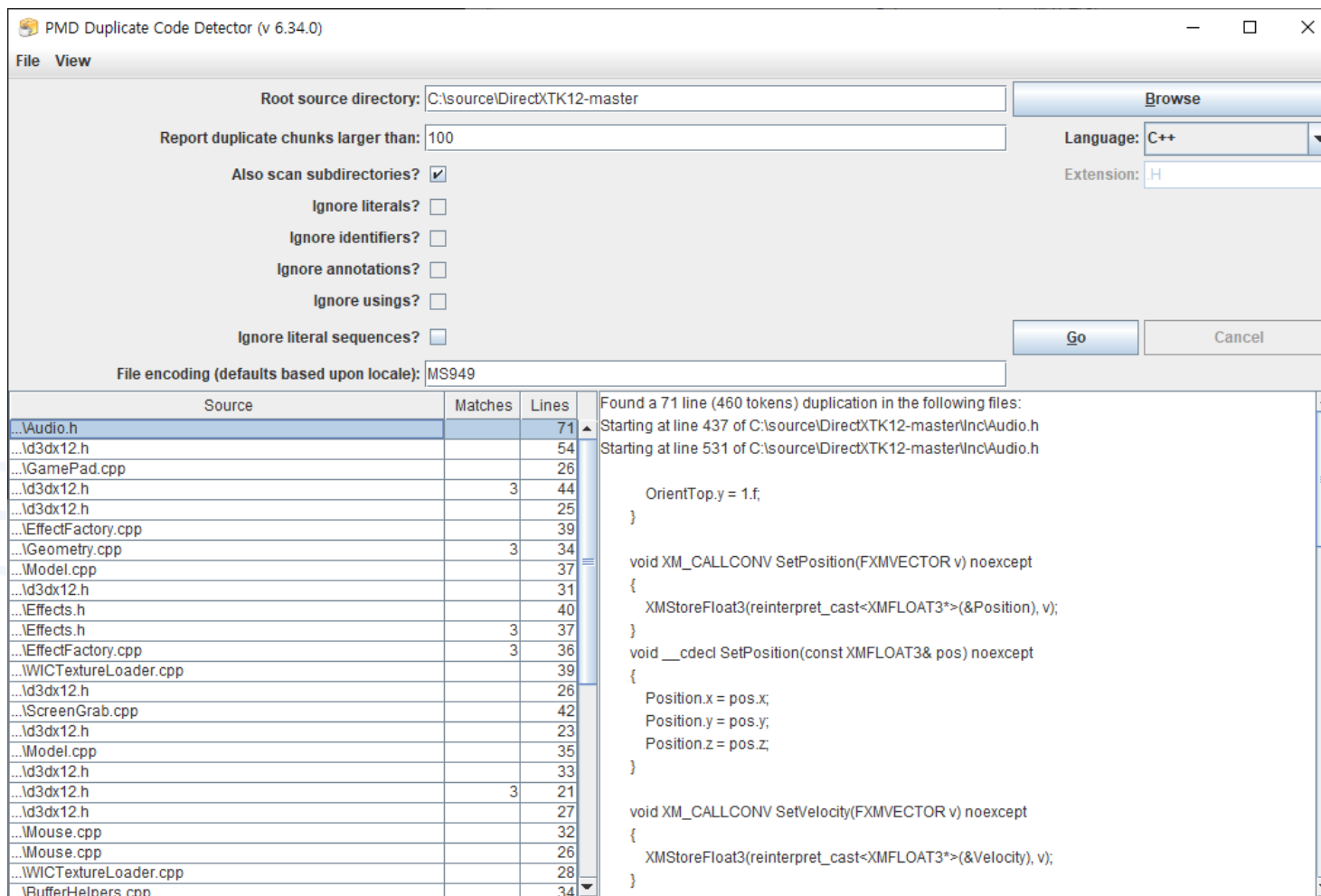


## • 사전 조건

- Java 설치

## • GUI 사용

- Wbin\Wcpdgui.bat 실행



## • 명령줄 실행

- 100 Token 이상, C++ 을 대상으로 분석
  - cpd.bat --minimum-tokens 100 --files . --language cpp
- 100 Token 이상, C++ 을 대상으로 분석, 엑셀에서 확인하기 위해 csv로 출력해서 저장
  - cpd.bat --minimum-tokens 100 --files . --language cpp --format csv > result.csv

lines	tokens	occurrences										
71	460	2	437	C:\source	531	C:\source\DirectX\TK12-master\W\Inc\Audio.h						
54	296	2	2641	C:\source	2759	C:\source\DirectX\TK12-master\W\Src\d3dx12.h						
26	281	2	502	C:\source	971	C:\source\DirectX\TK12-master\W\Src\GamePad.cpp						
44	266	3	2651	C:\source	2769	C:\source	2922	C:\source\DirectX\TK12-master\W\Src\d3dx12.h				
25	244	2	2965	C:\source	3030	C:\source\DirectX\TK12-master\W\Src\d3dx12.h						
39	231	2	152	C:\source	402	C:\source\DirectX\TK12-master\W\Src\EffectFactory.cpp						
34	231	3	800	C:\source	866	C:\source	1084	C:\source\DirectX\TK12-master\W\Src\Geometry.cpp				
37	227	2	551	C:\source	589	C:\source\DirectX\TK12-master\W\Src\Model.cpp						
31	224	2	2990	C:\source	3053	C:\source\DirectX\TK12-master\W\Src\d3dx12.h						
40	215	2	397	C:\source	459	C:\source\DirectX\TK12-master\W\Inc\Effects.h						

- 실행 방법

- `cpd --minimum-tokens 100 --files ./src --language cpp --format xml > cpd.xml || exit 0`
- `C:\WDevTools\Wpmd\Wbin\Wcpd --minimum-tokens 100 --files ./src --language cpp --format xml > cpd.xml || exit 0`

- 연동 방법

- Next Warning Generation에 적용

Tool ?

CPD

Report File Pattern ?

\*\*/\*.cpd.xml

Fileset 'includes' syntax specifying the files to scan for issues. If you leave this field empty then the default file pattern '\*\*/\*.cpd.xml' will be used.

☐ Skip symbolic links when searching for files ?

Report Encoding ?

Encoding of your report files.

High severity threshold

50

Minimum number of duplicated lines for high severity warnings.

Normal severity threshold

25

Minimum number of duplicated lines for normal severity warnings.

Custom ID ?

Optional custom ID (URL) of this tool, overwrites the built-in ID 'cpd'.

Custom Name ?

Optional custom display name of the tool, overwrites the built-in name 'CPD'.

CONTENTS / 023

# Pipeline 개념 이해

- Pipeline이란?

- 소스코드 버전 관리 저장소에서 최종 사용자까지의 지속적 배포(Continuous Delivery) 절차를 자동화 한 것
- 전반적인 배포 절차를 Pipeline domain-specific language(DSL) 문법을 활용해 "code"로 구성 한 것

- 방법

- jenkinsfile 이라 부르는 text 파일에 정의
  - 소스코드 버전 관리 저장소에 등록 후 활용
- Web UI 상에서 정의
  - Jenkins 내에서 활용

```
pipeline {  
    agent any  
  
    stages {  
        stage('Hello') {  
            steps {  
                echo 'Hello World'  
            }  
        }  
    }  
}
```

*Pipeline임을 의미*

*실행할 Node 설정*

*Pipeline의 각 단계. 예를 들어 Build, Test, Deploy 등*

*Stage 내의 단일 Task*

*실행 명령*

```
pipeline {  
  agent any  
  stages {  
    stage('SCM') { "SCM" 단계  
      steps {  
        git 'https://github.com/DongJoonHan/DirectXTK12.git' Git에서 소스코드 가져오기  
      }  
    }  
  
    stage('VS Build') { "VS Build" 단계  
      steps {  
        bat 'msbuild.exe DirectXTK_Desktop_2019_Win10.vcxproj' Windows 명령어 실행  
      }  
    }  
  }  
}
```



```
stage('Verification') {  
  steps {  
    bat 'cppcheck --enable=all --inconclusive --xml --xml-version=2 src 2> cppcheck.xml'  
    recordIssues enabledForFailure: true, aggregatingResults: true, tool: cppCheck(pattern: 'cppcheck.xml')
```

*CppCheck 실행*  
*결과 게시(Warning Next Generation 플러그인)*

#PROFESSIONAL IT EDU-PLATFORM

# Any Question?

code.presso()