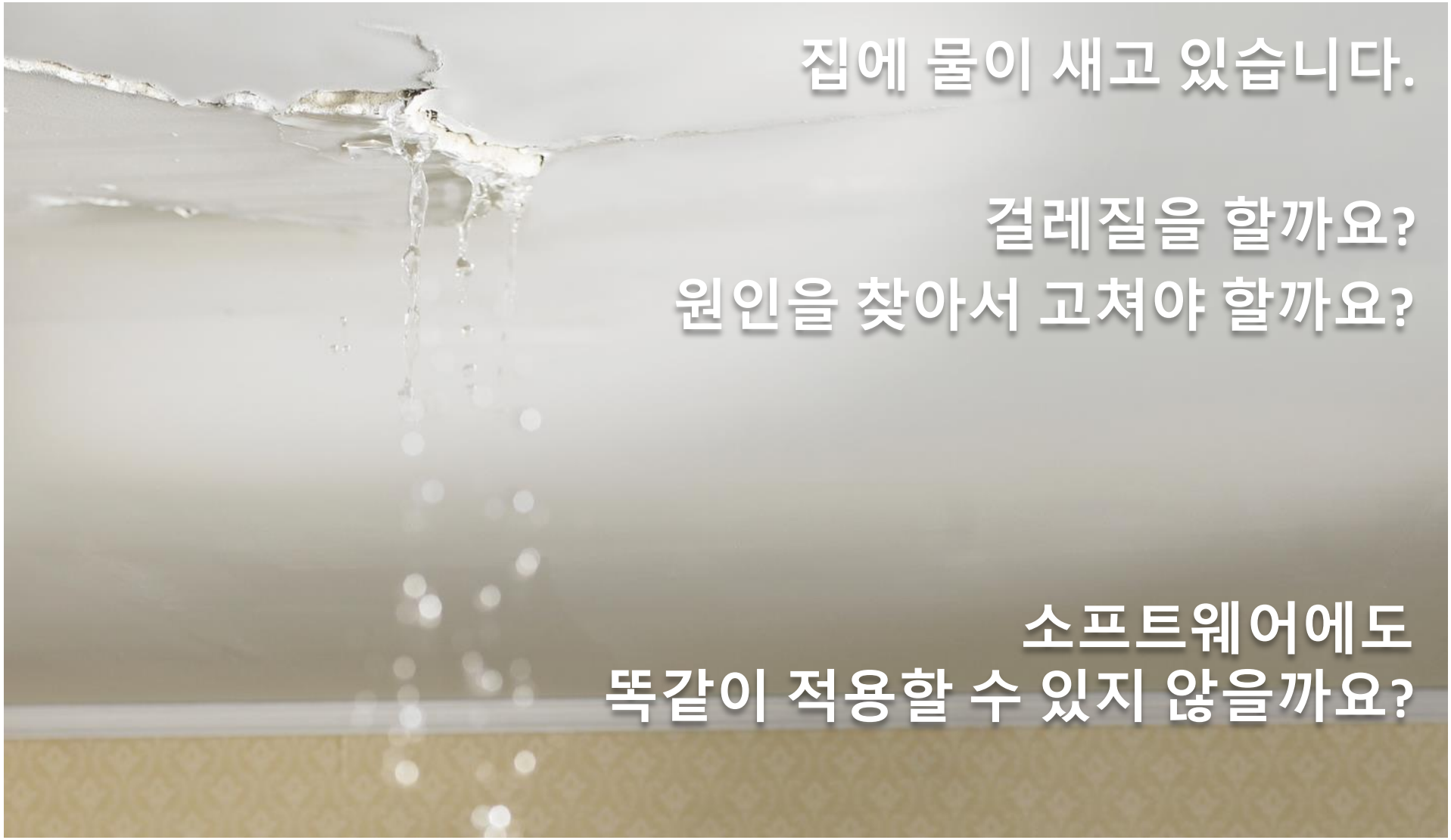
지속적인 코드 품질 관리

SonarSource SonarQube 소개

Version	Version 2.2
Date	2020.02
Author	CURVC Corp.
Type	Customer Presentation

Water Leak

sonarqube

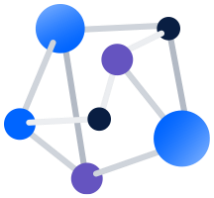


curvc

Traditional approach

sonarqube

전형적인 개발에서는 릴리즈 직전에 정기적인 코드 품질 감사를 수행하고 개발자가 조치를 취합니다. 이러한 접근은 단기간에 강력한 관리로 효과적일 수 있지만, 중장기적으로 일관되게 실패합니다.



너무 늦은

정시 감사 일정으로 늦은 문제 발견, 문제에 따라 프로젝트 일정 변경 필요

개발팀의 무관심

팀 외부에서 생성된 새로운 업무, 늦은 스케줄로 개발자의 학습 필요

오너쉽의 부재

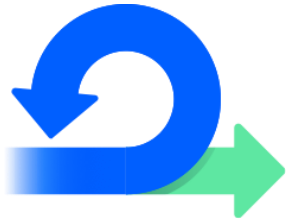
개발 프로세스는 개발자 품질 프로세스는 감사자

이질적인 요구사항

다른 요구사항으로 각 어플리케이션 측정, 공통적 품질 요구사항 적용이 불가능

SonarQube Concept

SonarQube의 지속적인 인스펙션(Continuous Inspection)은 소프트웨어 개발 라이프사이클의 완전한 부분으로 소프트웨어 품질을 향상시키기 위해 설계된 코드 품질 관리를 위한 새로운 패러다임입니다.



지속적인 검사

지속적인 통합처럼 지속적인 검사를 수행

품질 가시화

프로젝트 이해관계자를 위한 품질 가시화

조기에 문제 해결

개발 초기부터 문제를 해결하여 ROI 향상

문제 해결의 적시성

알림을 통한 최대한 빠른 해결과 개발 훈련

4

SonarQube Overview

SonarQube는 20개 이상의 프로그래밍 언어의 버그, 코드 스멜, 보안 취약성과 같은 정적 분석과 자동화된 리뷰를 수행하여 코드 품질의 지속적인 인스펙션을 지원하는 플랫폼입니다.

프로젝트 기반 품질 측정

- 프로젝트의 홈에서 품질 게이트, 버그, 취약성, 중복코드, 단위 테스팅 정보를 한눈에 보여주며 현재 코드의 품질 측면에서 위치를 확인합니다.
- 빌드와 통합을 통해 소스 코드 품질의 변화를 즉각적으로 확인할 수 있습니다.

다중 언어의 Coding Rules

- SonarQube는 Java, C/C++과 같은 인기있는 20개 이상의 언어의 코딩 룰을 제공합니다.
- Checkstyle, Clover, Findbugs, PMD 등 다양한 외부 분석기에 대한 플러그인을 지원합니다.

Quality Profiles

- SonarQube는 프로젝트 분석 시 사용되는 규칙(Rulesets)을 정의할 수 있습니다.
- 이 룰셋은 품질 프로파일로 구성되고 조직의 모든 구성원은 프로젝트에 적용되는 규칙을 확인할 수 있습니다.

Quality Gates

- 품질 게이트(Quality Gate)를 통해 다양한 조직의 품질 요구사항을 설정할 수 있습니다.
- 복잡도, 커버리지, 중복, 보안성, 유지보수성, 사이즈, 신뢰성, 이슈 등 다양한 항목에 대한 품질 요구사항을 설정합니다.

5

SonarQube Support Languages

sonarqube

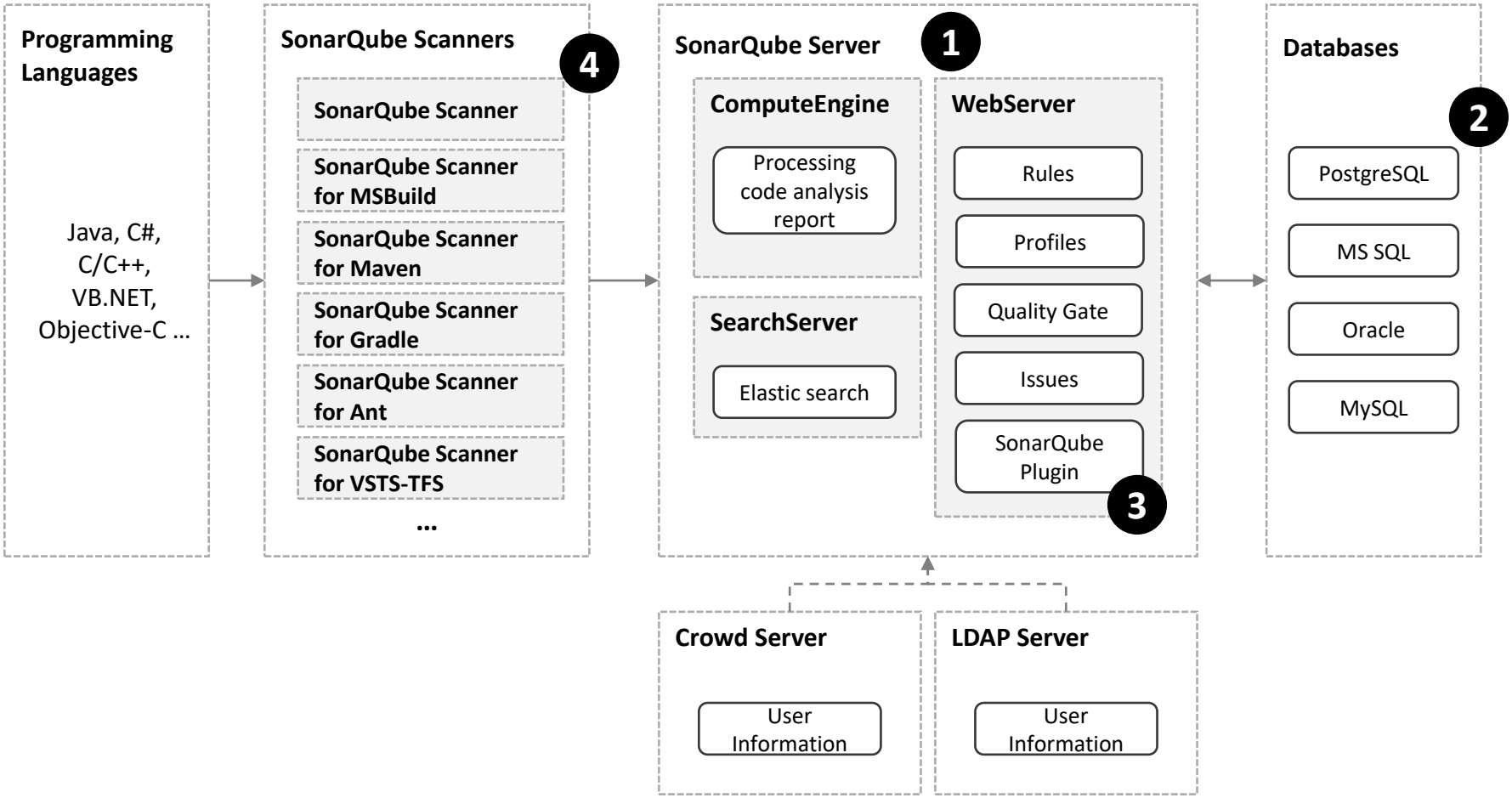
JAVA 540	Java Script 213	C# 372	TypeScript 123	Kotlin 43	
Go 40	Scala 41	Flex 73	Python 64	PHP 186	15 Languages
CSS 24	XML 10	VB.NET 132	Ruby 42	HTML 56	
C 256	C++ 377	Objective-C 254	T-SQL 79	ABAP 86	21 Languages
PL/SQL 186	Swift 116				
COBOL 182	RPG 57	PL/I 26	VB6 35	Apex 44	26 Languages

Community Edition | Developer Edition | Enterprise Edition

https://rules.sonarsource.com

SonarQube Architecture

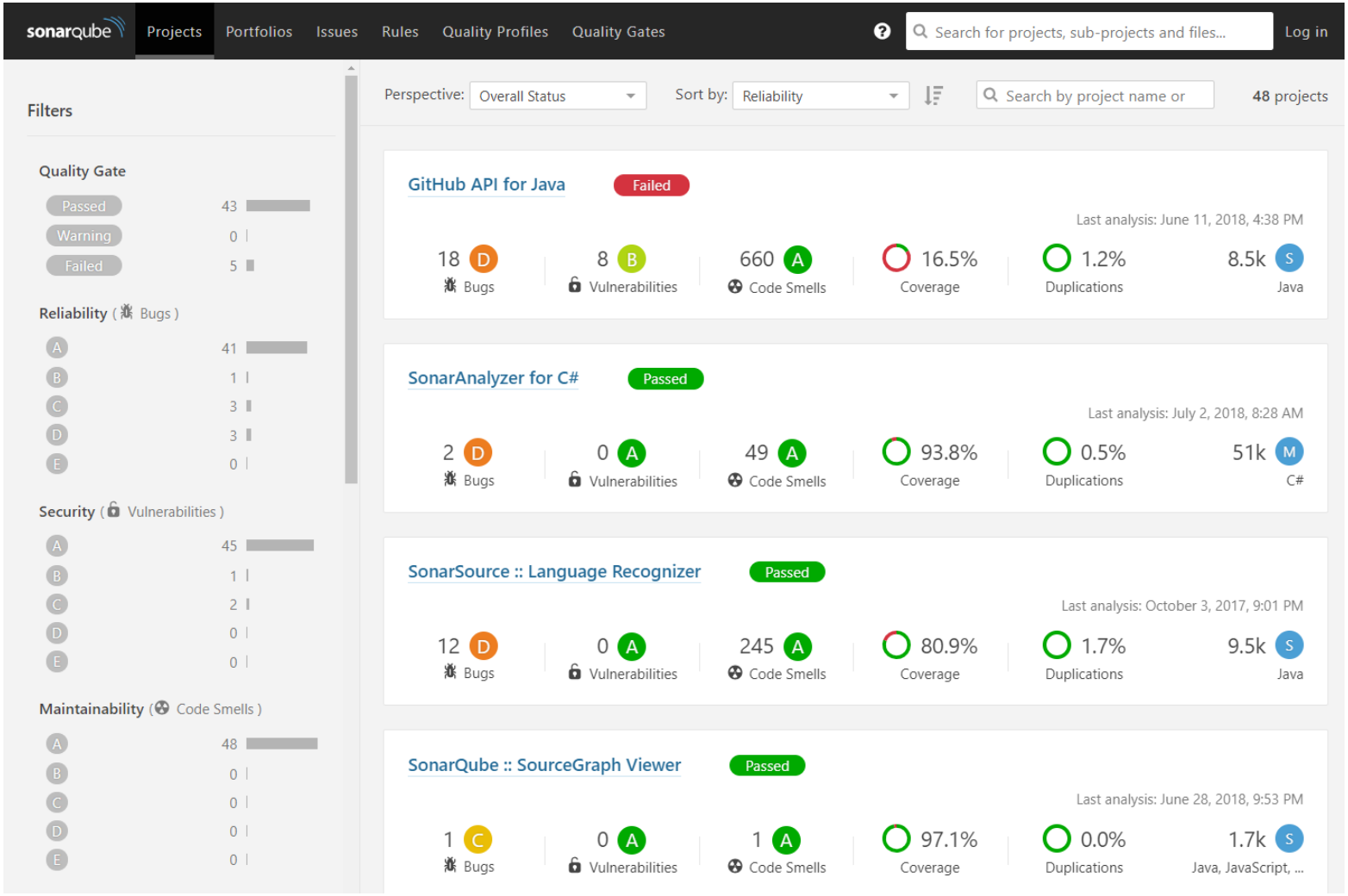
SonarQube는 4개의 컴포넌트로 구성됩니다. 첫번째는 SonarQube 서버, 두번째는 데이터베이스, 세번째는 SonarQube 플러그인, 마지막은 SonarQube Scanner로 구성됩니다.



7

SonarQube > Projects

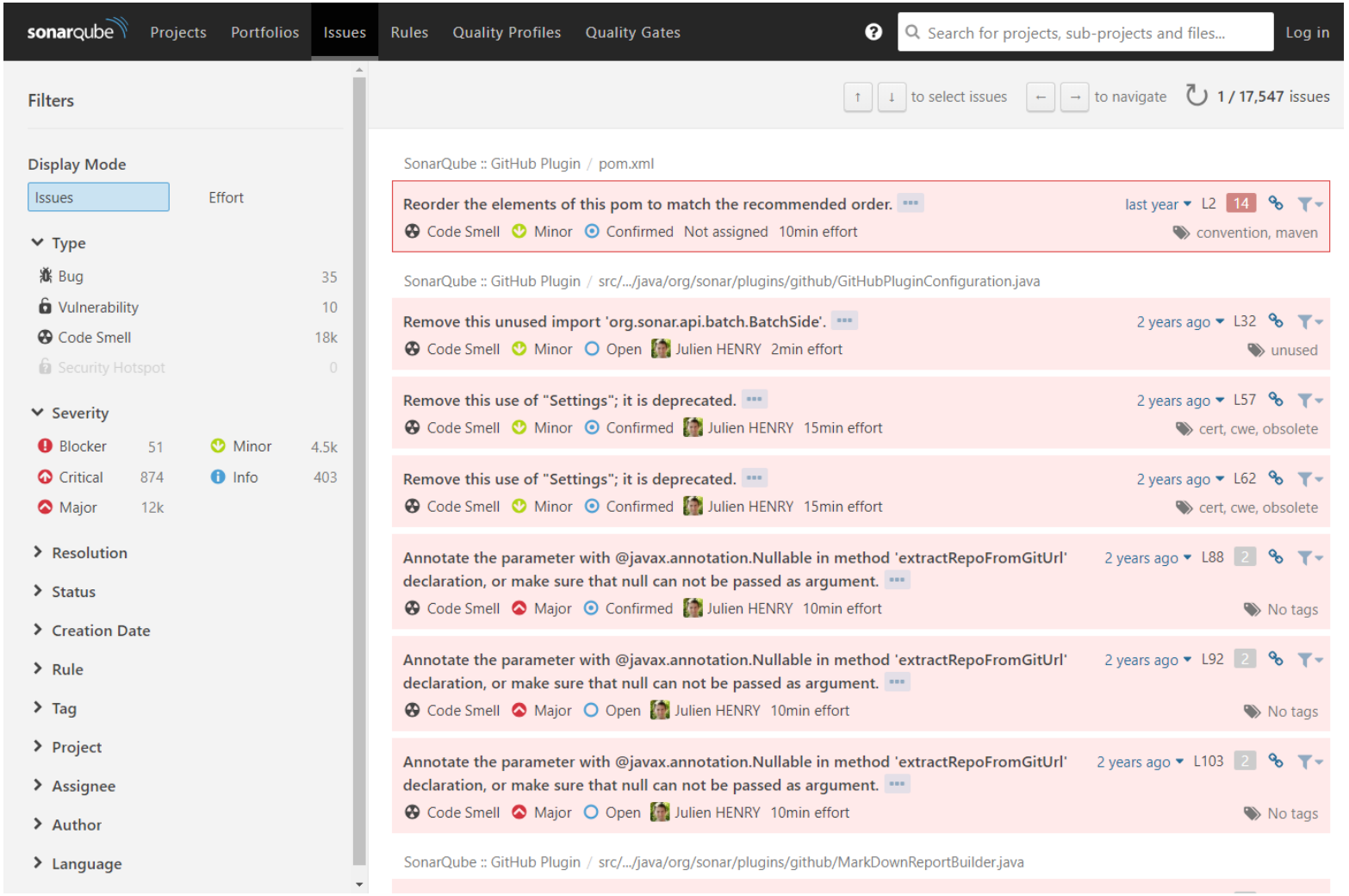
프로젝트 페이지에서는 전체 프로젝트의 현황을 빠르게 파악할 수 있게 해줍니다. 품질 요구사항을 만족하지 못하는 프로젝트 혹은 특정 이슈들이 발견된 프로젝트를 필터할 수 있습니다.



8

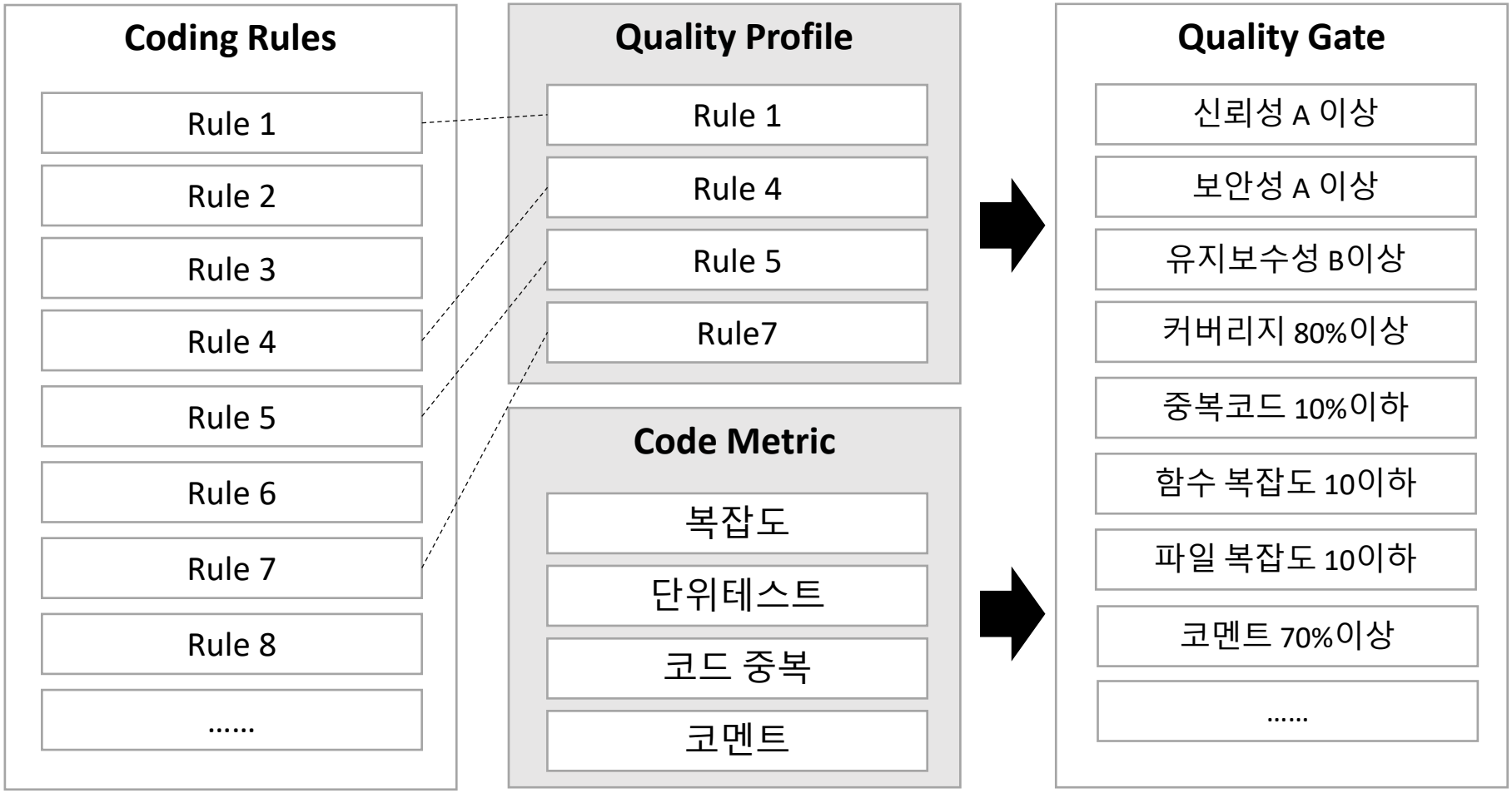
SonarQube > All Issues

SonarQube는 교차 프로젝트 서비스를 제공합니다. 개발자들은 프로젝트와 상관없이 자신에게 할당된 이슈리스트를 확인할 수 있습니다.



SonarQube > Quality Management

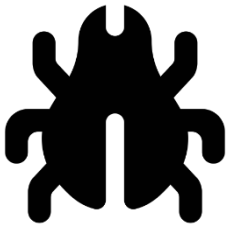
SonarQube는 코드 품질 관리를 위해 다음과 같이 Rules, Quality Profiles, Code Metric, Quality Gate의 개념이 포함되어 있습니다.



10

SonarQube > Rule Type

Poor한 코드 품질은 팀 속도 저하, 어플리케이션 폐기, 프로덕션에서 크래쉬, 나쁜 회사 평판과 같은 다양한 문제를 일으킵니다. SonarQube는 신뢰성, 보안성, 유지보수성을 향상시키는 솔루션을 제공합니다.



Bugs
Reliability

잠재적인 버그, 런타임에 예상되는 동작을 가지지 않는 코드

Vulnerabilities
Security

어플리케이션이 잘못된 방식으로 사용될 수 있는 취약점

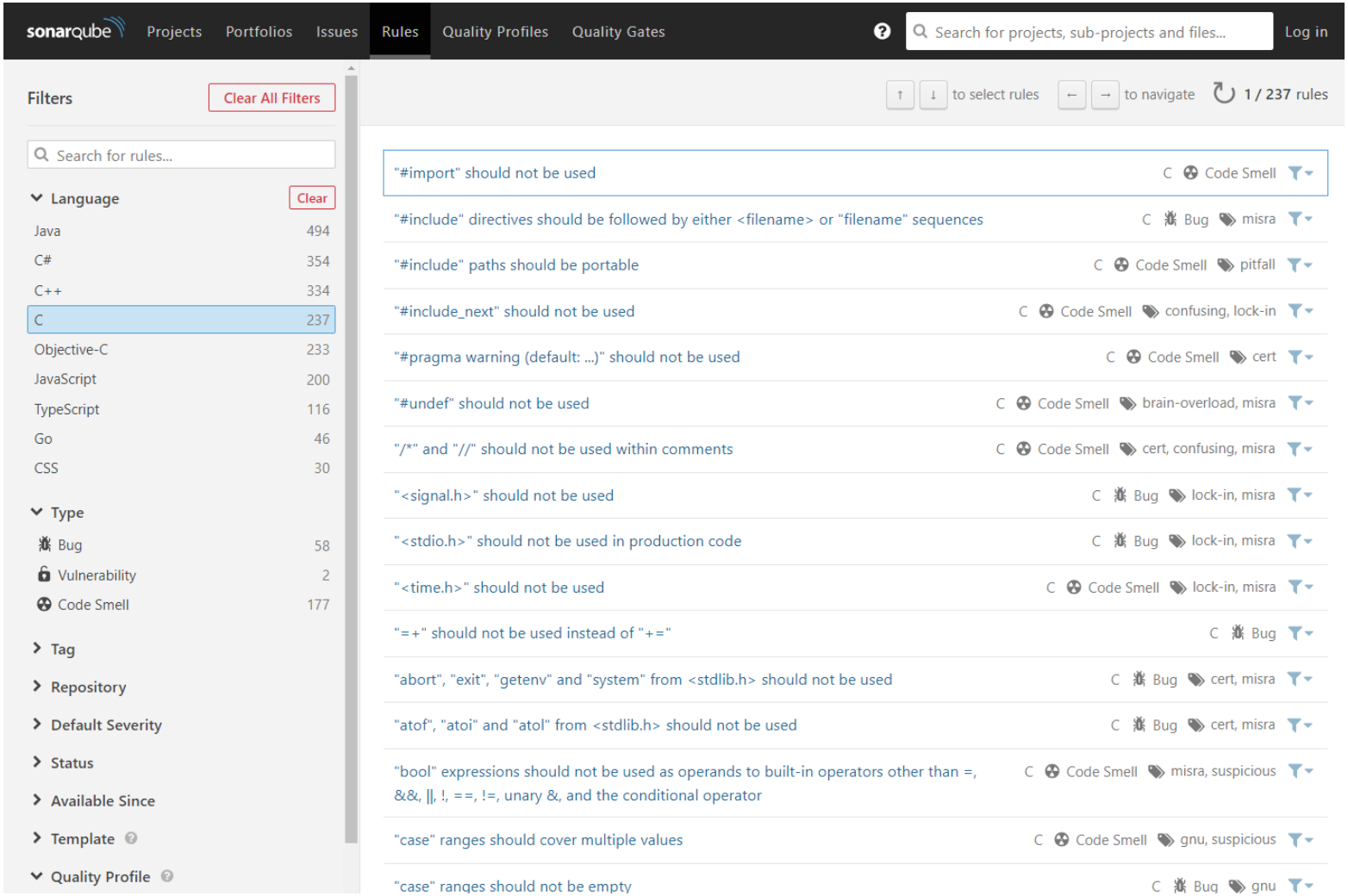
Code Smells
Maintainability

모듈성, 이해가능성, 변경가능성, 테스트 가능성, 재사용성

11

SonarQube > Rules

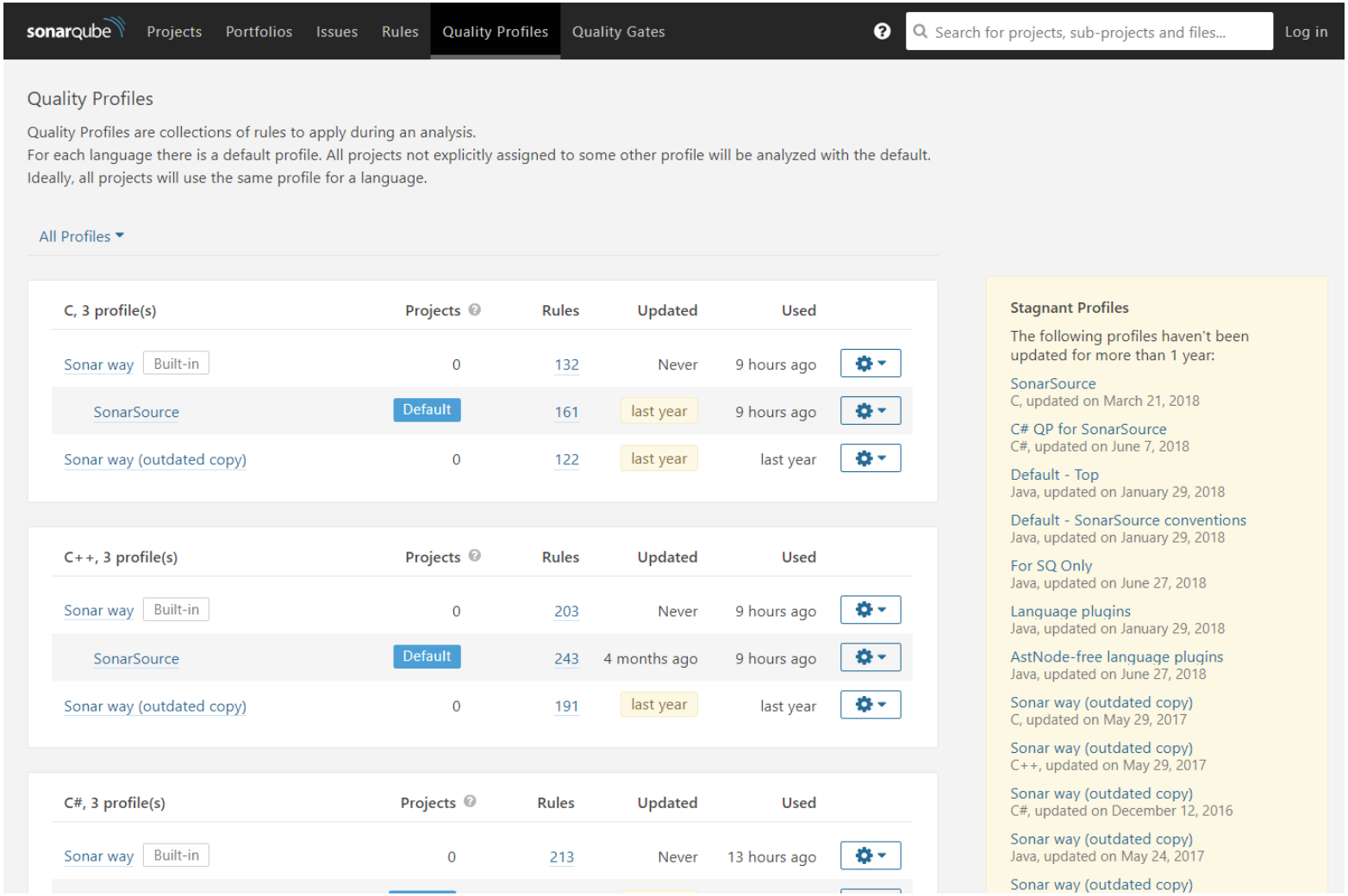
SonarQube의 Rules 페이지에서는 SonarQube에 존재하는 모든 룰을 확인할 수 있습니다. 지원 언어별, 타입별, Tag별, Third Party Rule 등 모든 룰을 확인할 수 있습니다.



12

SonarQube > Profiles 1/2

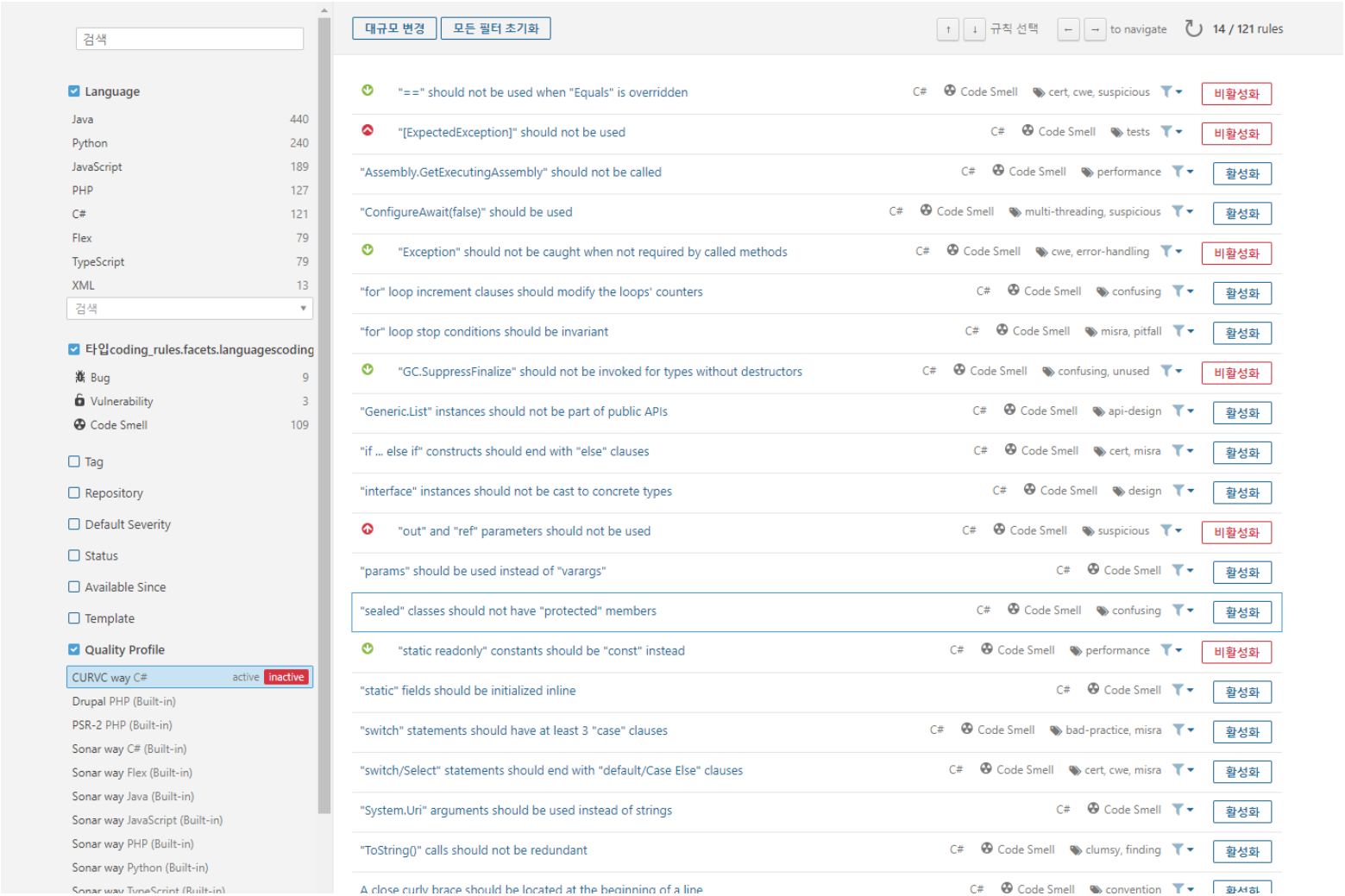
Quality Profiles은 Rules의 집합을 정의함으로써 조직이 원하는 코딩 룰에 대한 요구사항을 정의할 수 있기 때문에 SonarQube의 핵심입니다.



13

SonarQube > Profiles 2/2

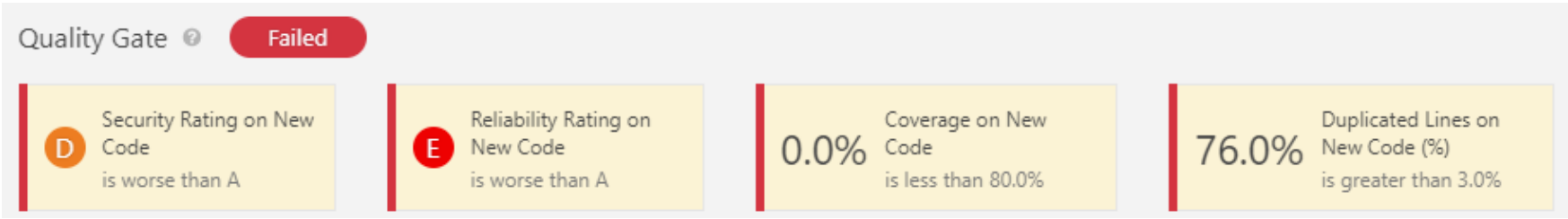
Quality Profiles은 Rules의 집합을 정의함으로써 조직이 원하는 코딩 룰에 대한 요구사항을 정의할 수 있기 때문에 SonarQube의 핵심입니다. 이를 통해 원하는 Rules을 추가하거나 제외할 수 있습니다.

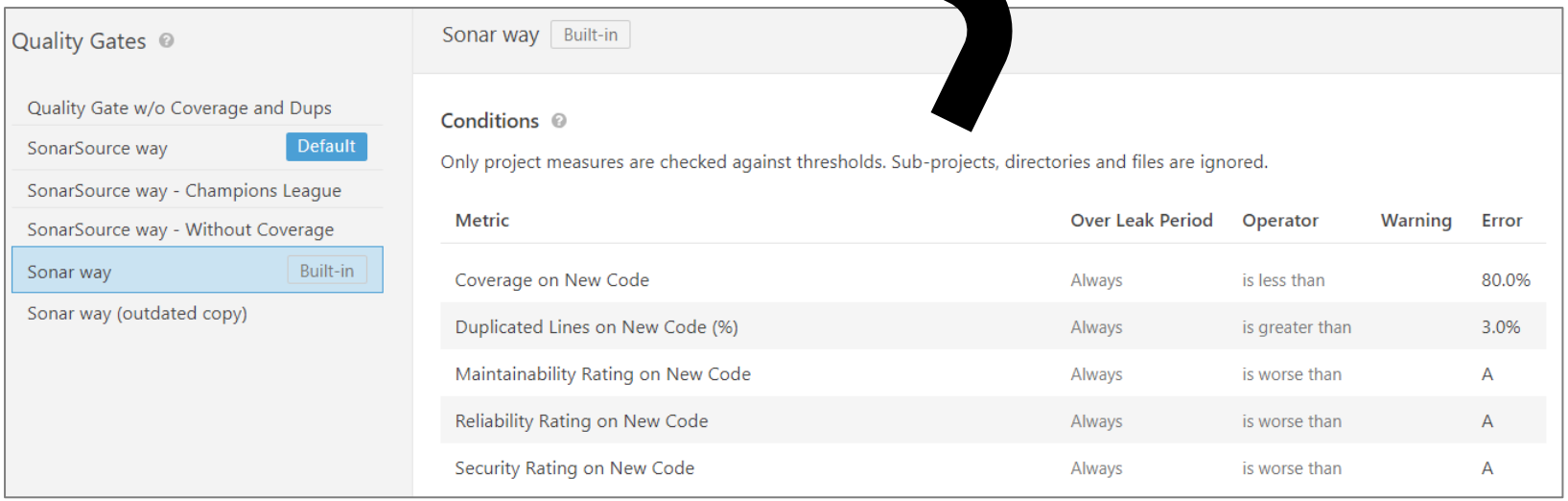


14

SonarQube > Quality Gate

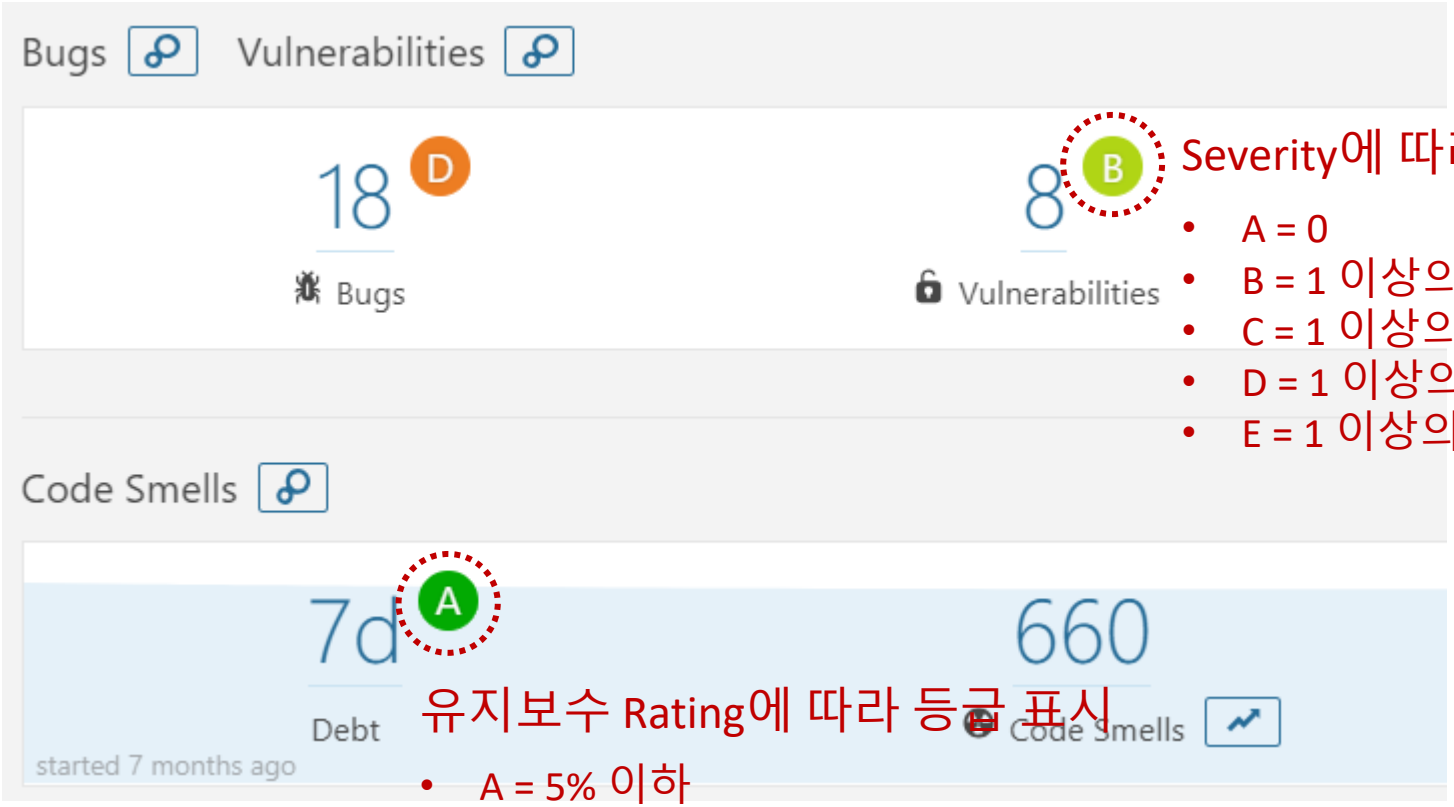
Quality Gate는 어플리케이션을 릴리즈 하기 전에 어플리케이션이 품질 요구사항을 지키고 있는지 확인하는 기준 목록입니다. Quality Gate를 통해 어플리케이션의 출시 여부를 결정할 수 있습니다.





15

SonarQube > Rating Model



Severity에 따라 등급 표시

- A = 0
- B = 1 이상의 Minor 존재 시
- C = 1 이상의 Major 존재 시
- D = 1 이상의 Critical 존재 시
- E = 1 이상의 Blocker 존재 시

유지보수 Rating에 따라 등급 표시

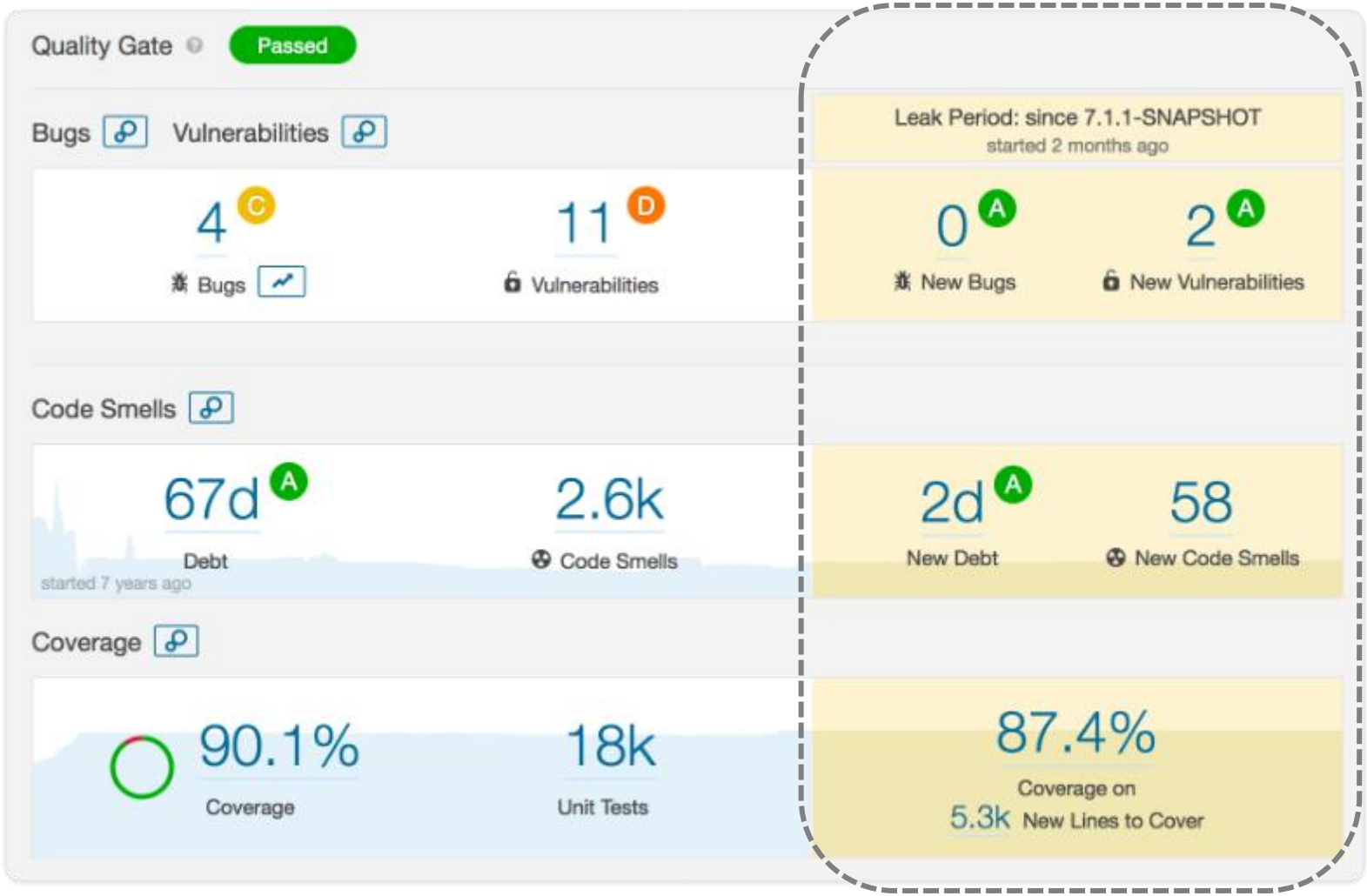
- A = 5% 이하
- B = 6~10%
- C = 11~20%
- D = 21~50%
- E = 50%~

수정비용 / 개발비용

16

SonarQube > The Water Leak Paradigm

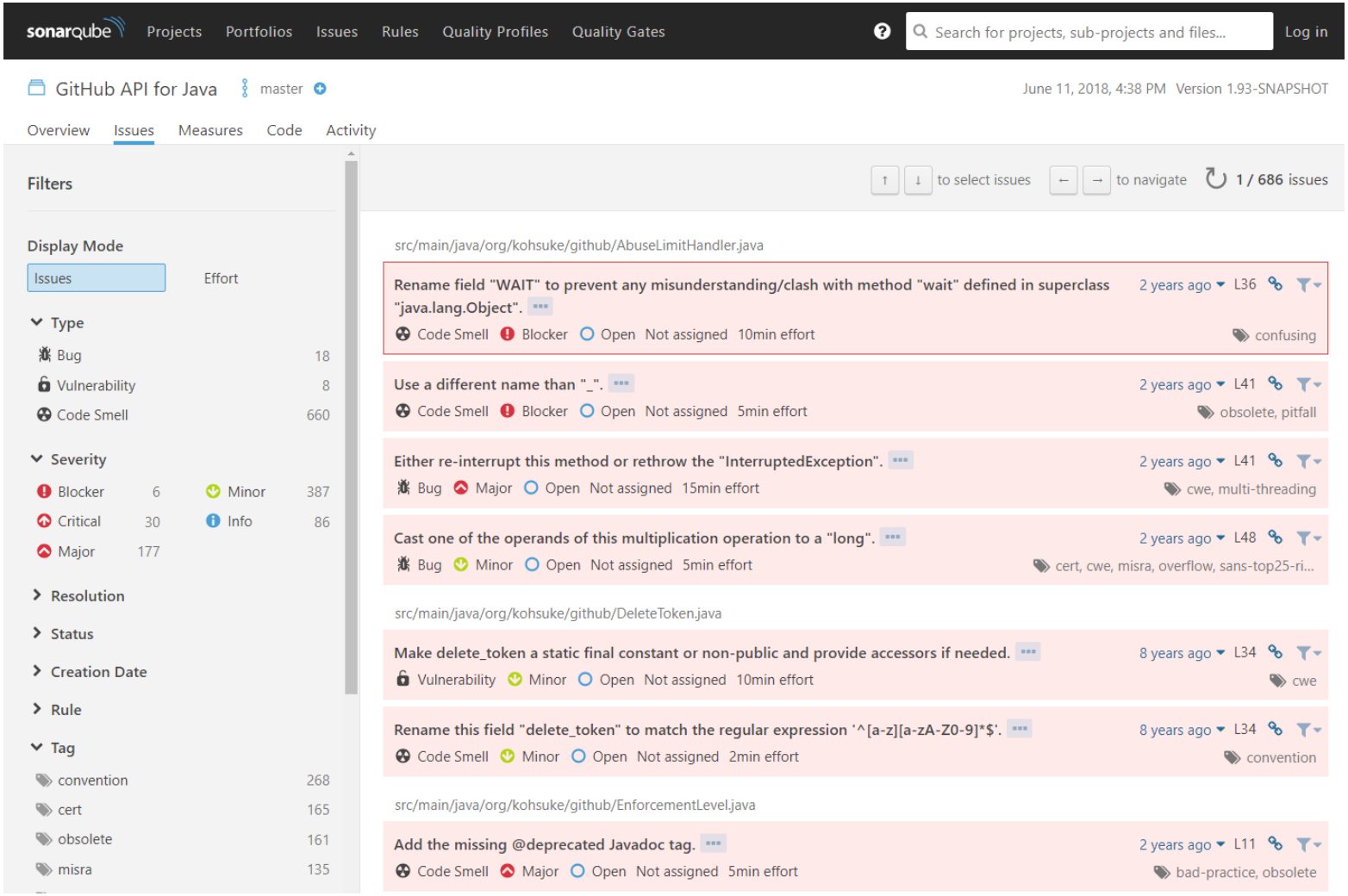
코드 품질 관리를 위해 권장되는 접근방법은 무엇일까요?
물이 새는 것을 관리하고 바닥을 청소하기 전에 물이 새는 것을 수리하세요.



17

SonarQube > Project Issues

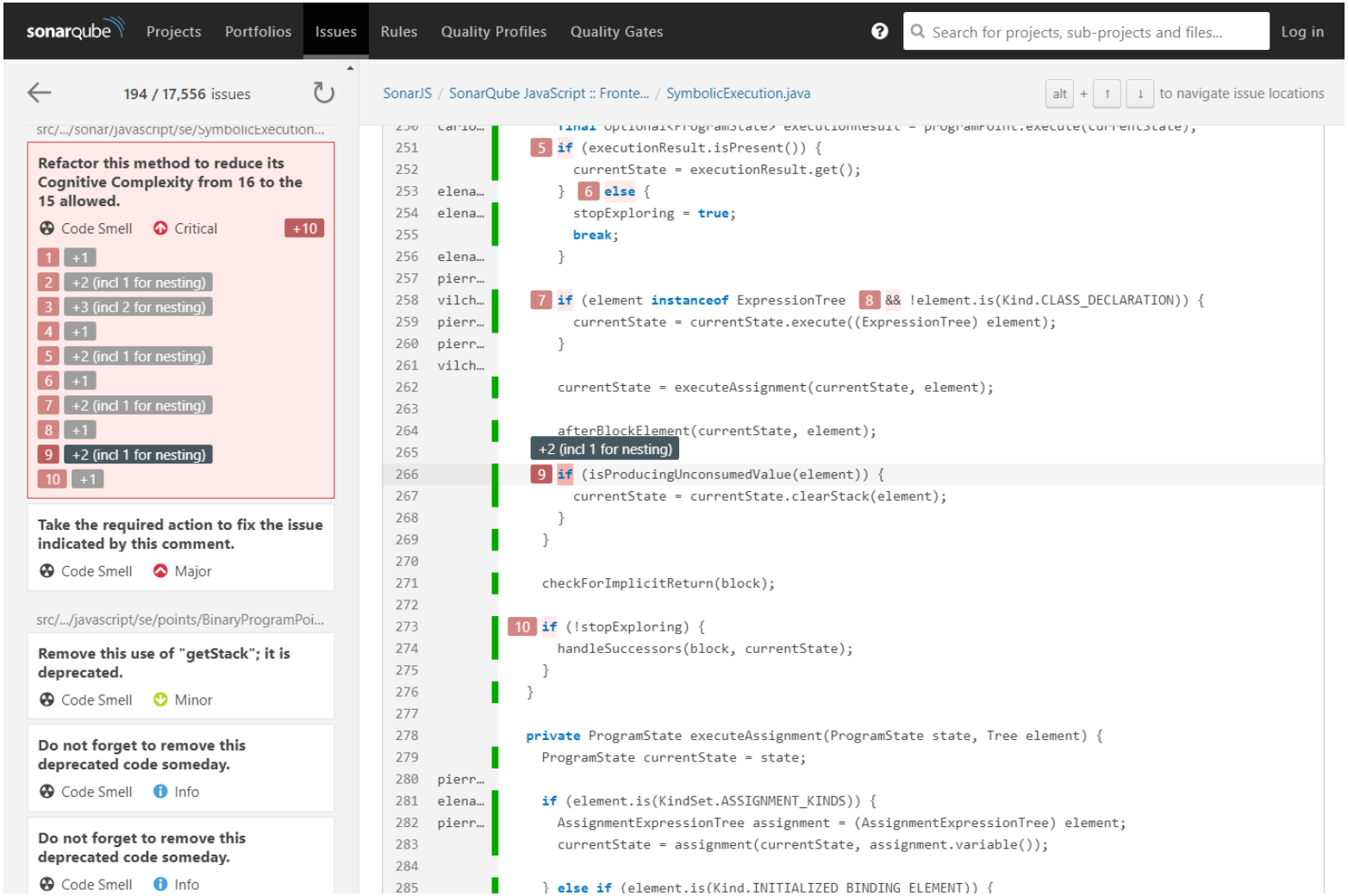
Project의 Issues 페이지는 프로젝트에 존재하는 이슈들을 보여줍니다. 개발자들은 자신에게 할당된 프로젝트 이슈를 확인할 수 있습니다.



18

SonarQube > Issues

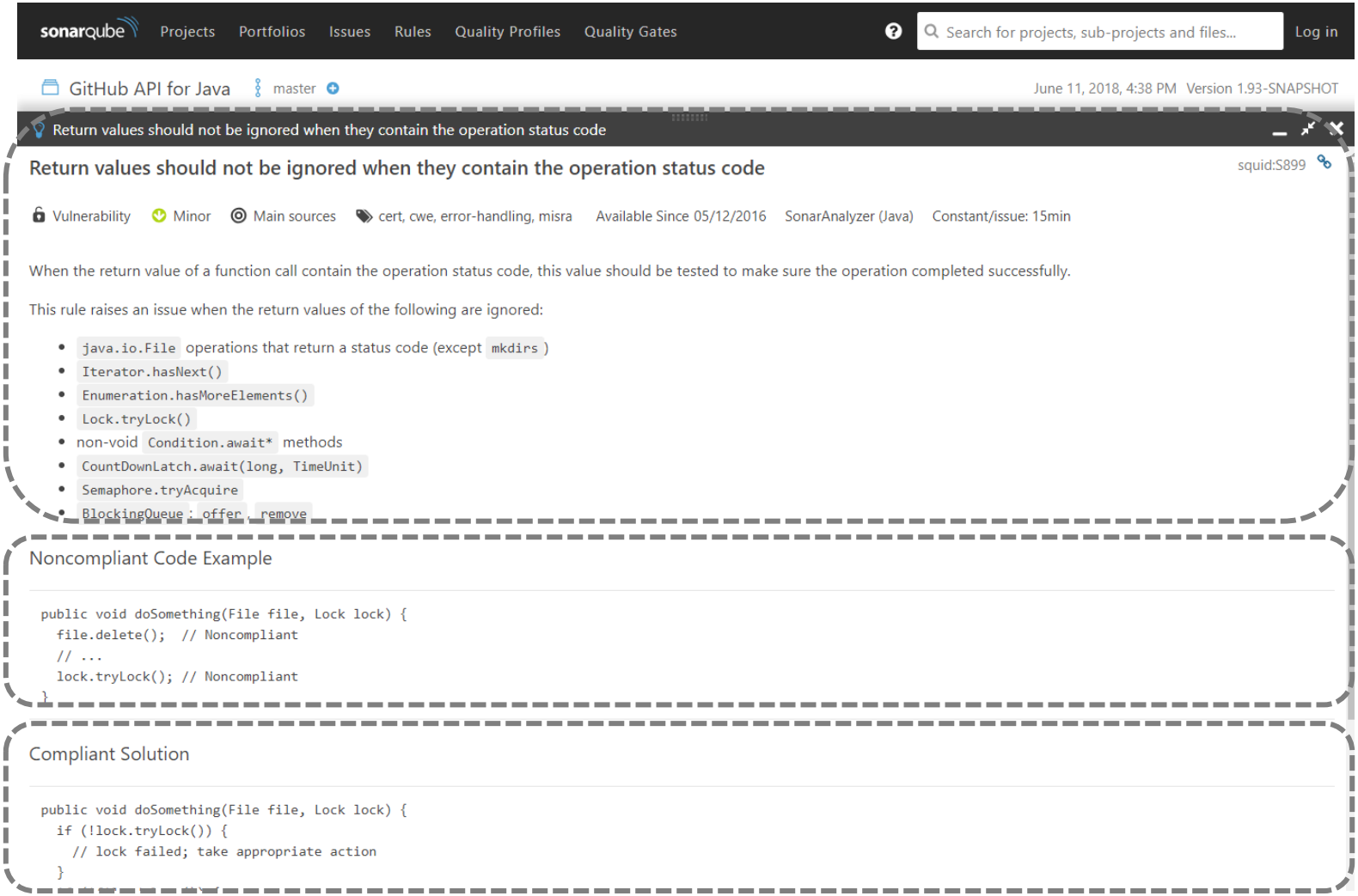
SonarQube는 여러 위치에서 발견된 이슈를 코드 경로 상에 표시하고 쉽게 네비게이션하여 찾을 수 있게 도와줍니다.



19

SonarQube > Issues

SonarQube는 발견된 이슈의 세부적인 정보를 제공합니다. 이슈에 대한 설명과 함께 잘못된 예제와 코딩 룰을 준수하는 예제를 확인할 수 있습니다.



20

SonarQube > Metric 1/3

SonarQube의 Measure에서는 Reliability, Security, Maintainability, Tests, Duplications, Complexity, Size, Issues 등 다양한 Metric을 정의하고 측정하고 있습니다.

Reliability

Name	Description
Bugs	버그 수
New Bugs	새로운 버그 수
Reliability Rating	A = 0 버그 B = 1 이상의 Minor 버그 C = 1 이상의 Major 버그 D = 1 이상의 Critical 버그 E = 1 이상의 Blocker 버그
Reliability remediation effort	버그 이슈를 수정하는데 소요되는 시간(분 단위 저장)
Reliability remediation effort on new code	새로운 코드에서 버그 이슈를 수정하는데 소요되는 시간

Security

Name	Description
Vulnerabilities	취약점 수
New Vulnerabilities	새로운 취약점 수
Security Rating	A = 0 취약성 B = 1 이상의 Minor 취약성 C = 1 이상의 Major 취약성 D = 1 이상의 Critical 취약성 E = 1 이상의 Blocker 취약성
Security remediation effort	취약성 이슈를 수정하는데 소요되는 시간(분 단위 저장)
Security remediation effort on new code	새로운 코드에서 취약성 이슈를 수정하는데 소요되는 시간

21

SonarQube > Metric 2/3

SonarQube의 Measure 에서는 Reliability, Security, Maintainability, Tests, Duplications, Complexity, Size, Issues 등 다양한 Metric을 정의하고 측정하고 있습니다.

Maintainability

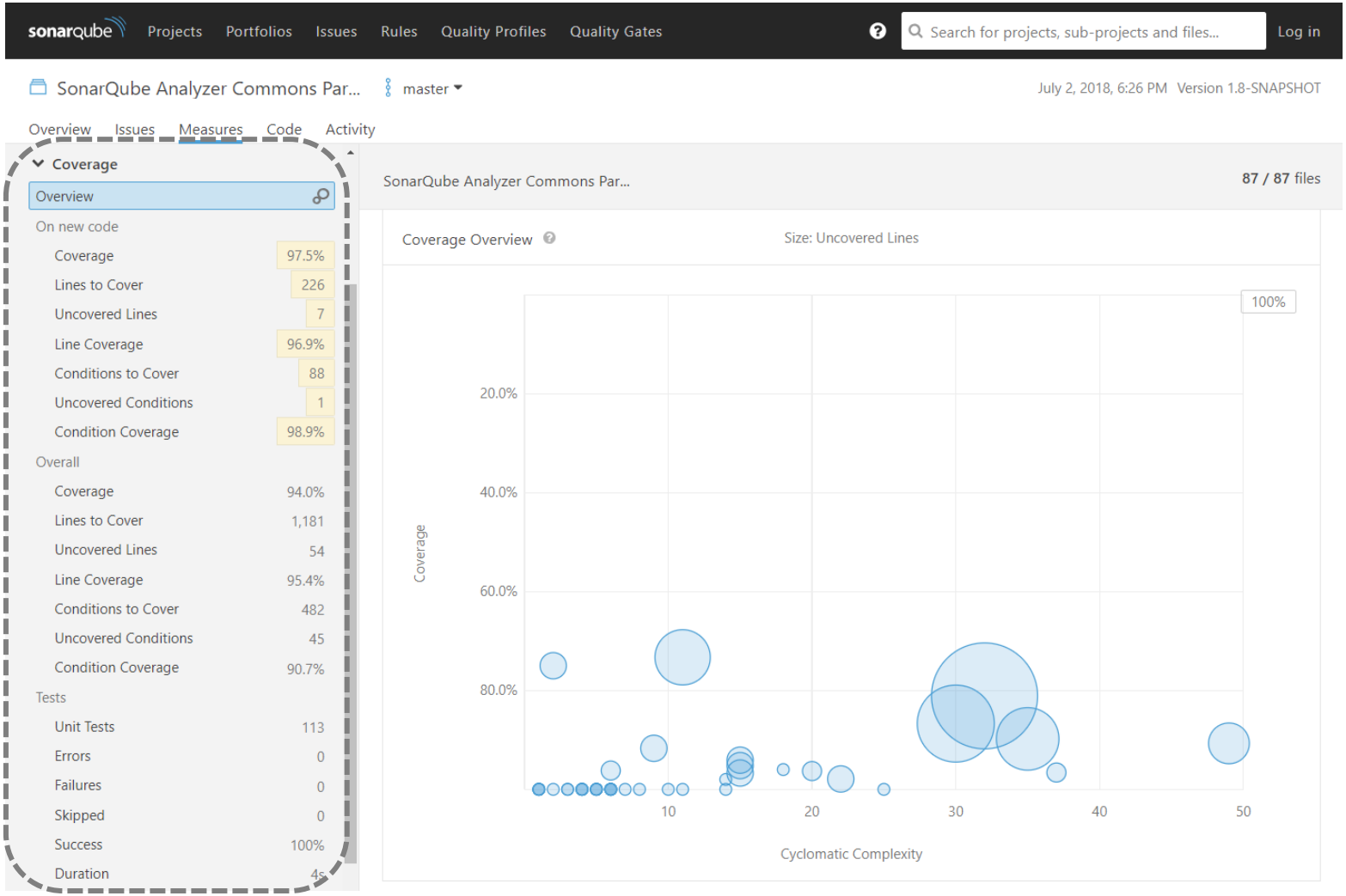
Name	Description
Code Smells	코드 스멜 수
New Code Smells	새로운 코드 스멜 수
Maintainability Rating	A = 5% 이하 B = 6~10% C = 11~20% D = 21~50% E = 50%~
Technical Debt	유지보수 이슈를 수정하는데 소요되는 시간(분 단위 저장)
Technical Debt on new code	새로운 코드에서 유지보수 이슈를 수정하는데 소요되는 시간
Technical Debt Ratio	수정 비용 / 개발 비용 코드 한 줄 개발 비용은 0.06일
Technical Debt Ratio on new code	새로운 코드에서 Technical Debt Ratio

Tests

Name	Description
Condition Coverage	Condition coverage = (CT + CF) / (2*B)
Coverage	Coverage = (CT + CF + LC)/(2*B + EL)
Line Coverage	Line coverage = LC / EL
Unit tests	단위 테스트 수
Unit test duration	단위 테스트 수행 시간
Unit test errors	실패된 단위 테스트 수
Unit test failures	예상치 못한 실행으로 실패된 단위 테스트 수
Unit test success density(%)	테스트 성공 밀도 = (Unit tests - (Unit test errors + Unit test failures)) / Unit tests * 100

SonarQube > Metric 2/3

SonarQube의 Measure 에서는 Reliability, Security, Maintainability, Tests, Duplications, Complexity, Size, Issues 등 다양한 Metric을 정의하고 측정하고 있습니다.



23

SonarQube > Metric 3/3

SonarQube의 Measure 에서는 Reliability, Security, Maintainability, Tests, Duplications, Complexity, Size, Issues 등 다양한 Metric을 정의하고 측정하고 있습니다.

Size

Name	Description
Classes	클래스의 수
Comment lines	주석 혹은 주석 처리된 코드 수
Comments (%)	주석 밀도 = 주석 줄 / (코드 줄 + 주석 줄) * 100
Directories	디렉토리 수
Files	파일 수
Lines	전체 라인수
Lines of code	문자가 하나이상 포함된 물리적 라인 수
Functions	함수 수
Statements	문장 수

Complexity

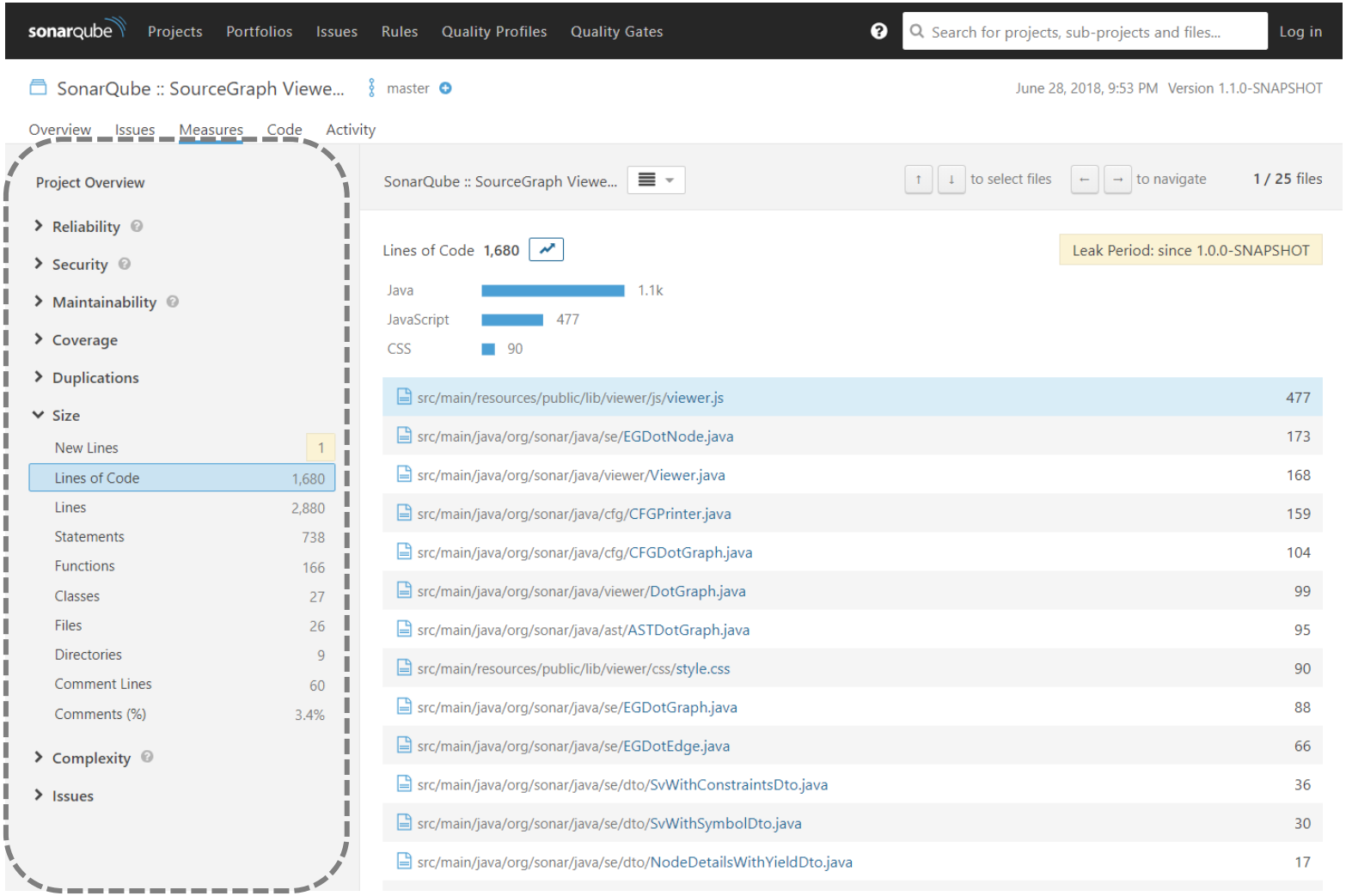
Name	Description
Complexity	코드의 경로 수를 기반으로 계산된 복잡도
Cognitive Complexity	더 좋은 복잡도 측정법

Duplications

Name	Description
Duplicated blocks	중복된 라인 블록 수
Duplicated files	중복된 파일 수
Duplicated lines	중복된 라인 수
Duplicated lines(%)	중복 밀도 = 중복된 라인 / 라인 수 * 100

SonarQube > Metric 3/3

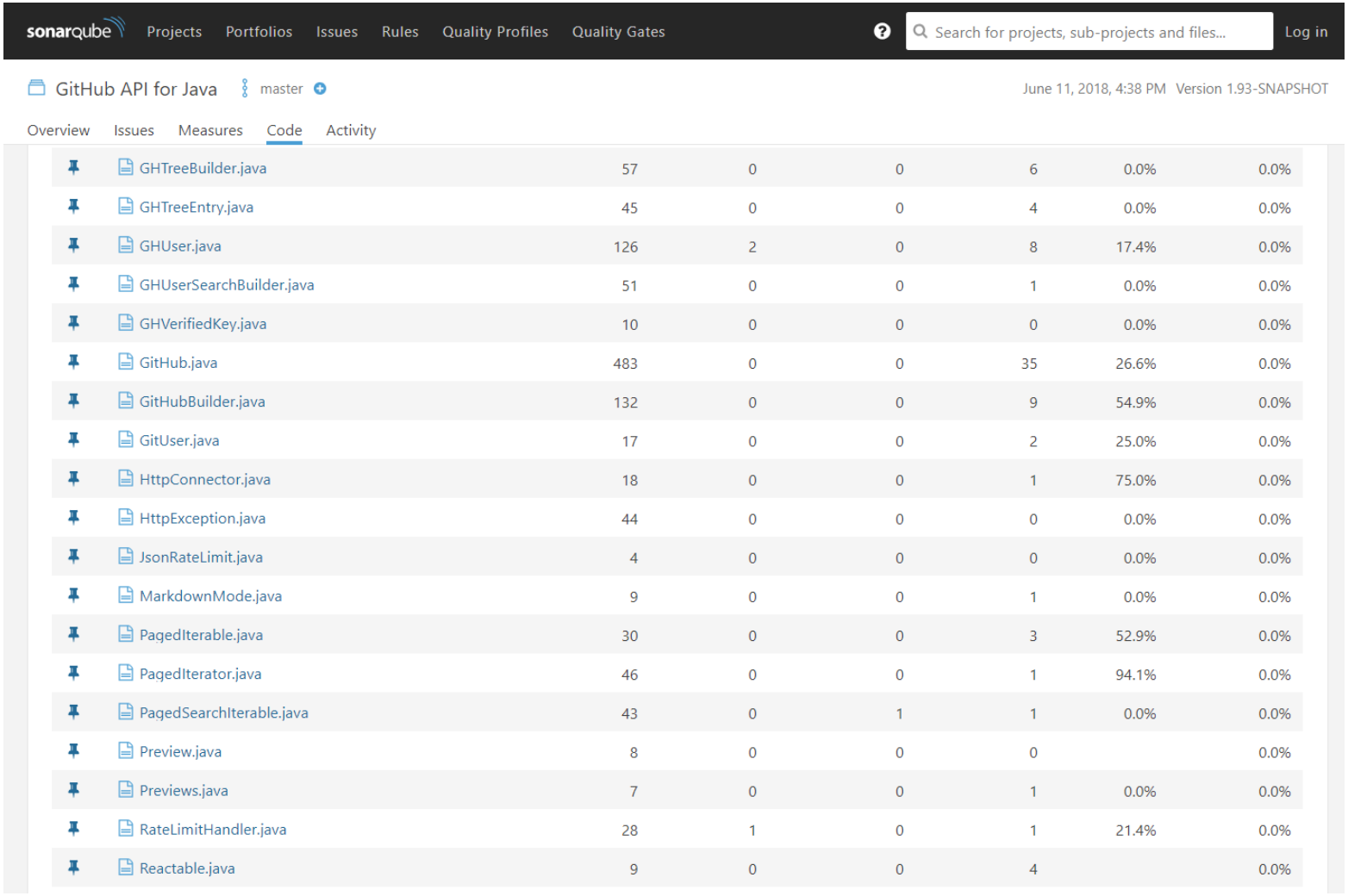
SonarQube의 Mesure에서는 Reliability, Security, Maintainability, Tests, Duplications, Complexity, Size, Issues등 다양한 Metric을 정의하고 측정하고 있습니다.



25

SonarQube > Code

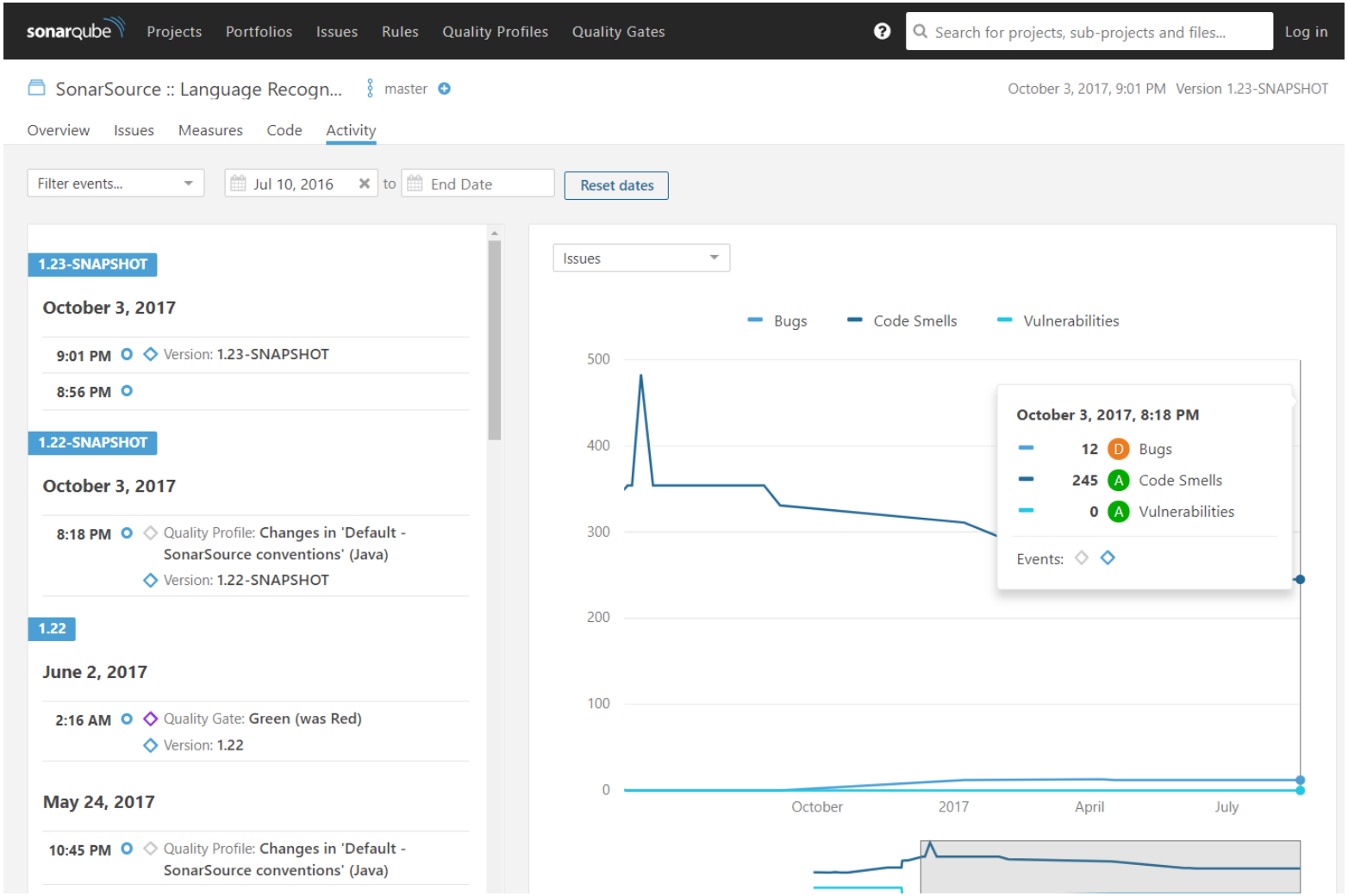
SonarQube의 Project Code에서는 폴더 구조에 해당하는 코드의 LOC, Bugs, Vulnerabilities, Code Smells, Coverage, Duplication을 확인할 수 있습니다.



GHTreeBuilder.java	57	0	0	6	0.0%	0.0%
GHTreeEntry.java	45	0	0	4	0.0%	0.0%
GHUser.java	126	2	0	8	17.4%	0.0%
GHUserSearchBuilder.java	51	0	0	1	0.0%	0.0%
GHVerifiedKey.java	10	0	0	0	0.0%	0.0%
GitHub.java	483	0	0	35	26.6%	0.0%
GitHubBuilder.java	132	0	0	9	54.9%	0.0%
GitUser.java	17	0	0	2	25.0%	0.0%
HttpConnector.java	18	0	0	1	75.0%	0.0%
HttpException.java	44	0	0	0	0.0%	0.0%
JsonRateLimit.java	4	0	0	0	0.0%	0.0%
MarkdownMode.java	9	0	0	1	0.0%	0.0%
PagedIterable.java	30	0	0	3	52.9%	0.0%
PagedIterator.java	46	0	0	1	94.1%	0.0%
PagedSearchIterable.java	43	0	1	1	0.0%	0.0%
Preview.java	8	0	0	0		0.0%
Previews.java	7	0	0	1	0.0%	0.0%
RateLimitHandler.java	28	1	0	1	21.4%	0.0%
Reactable.java	9	0	0	4		0.0%

SonarQube > Activity

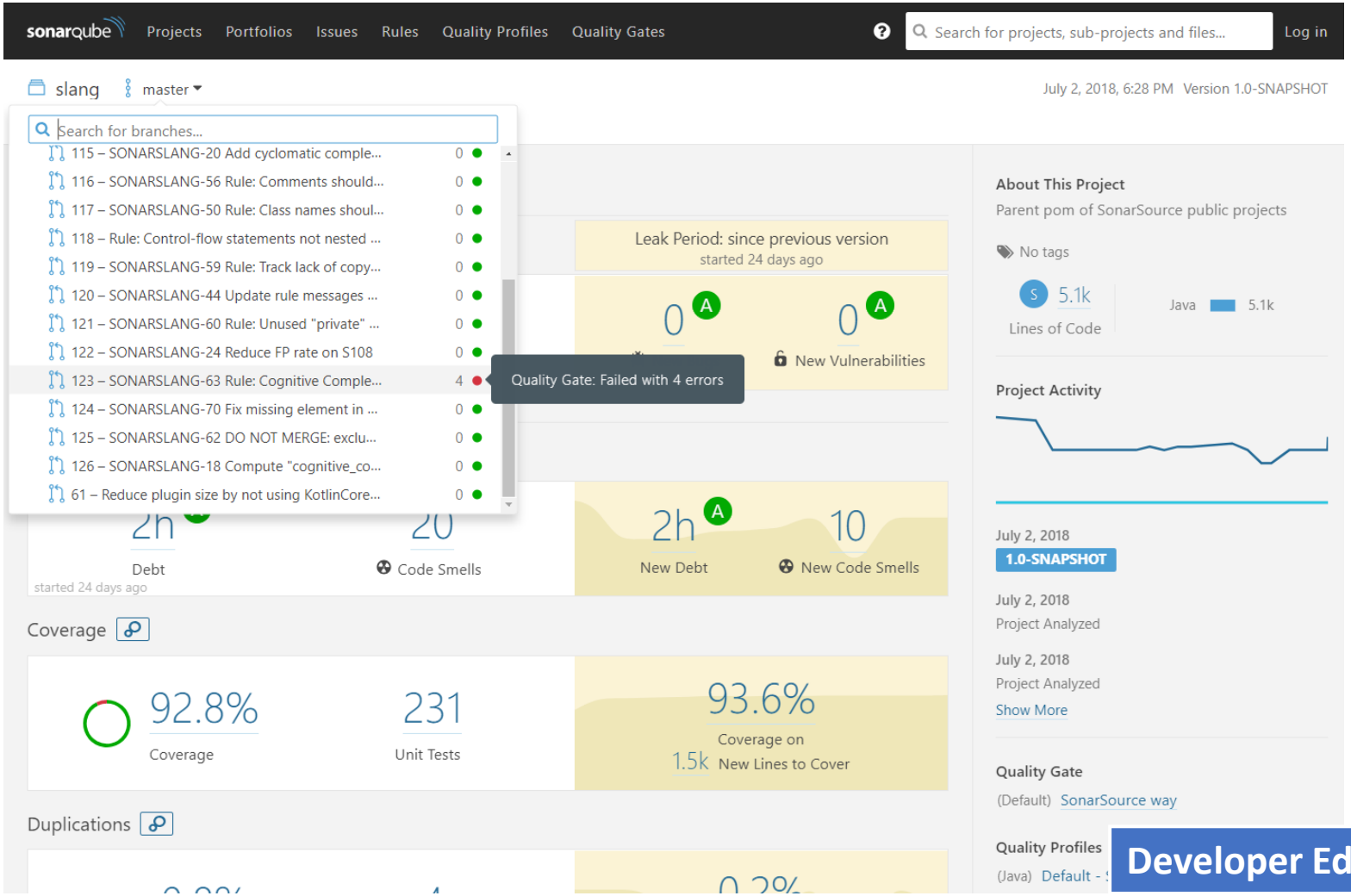
SonarQube의 Project Activity는 SonarQube 측정 지표의 트랜드를 확인할 수 있습니다. 기본적으로 Issues, Coverage, Duplication을 보여주며, Custom을 통해 측정한 모든 데이터의 트랜드를 확인할 수 있습니다.



27

SonarQube > Branch analysis

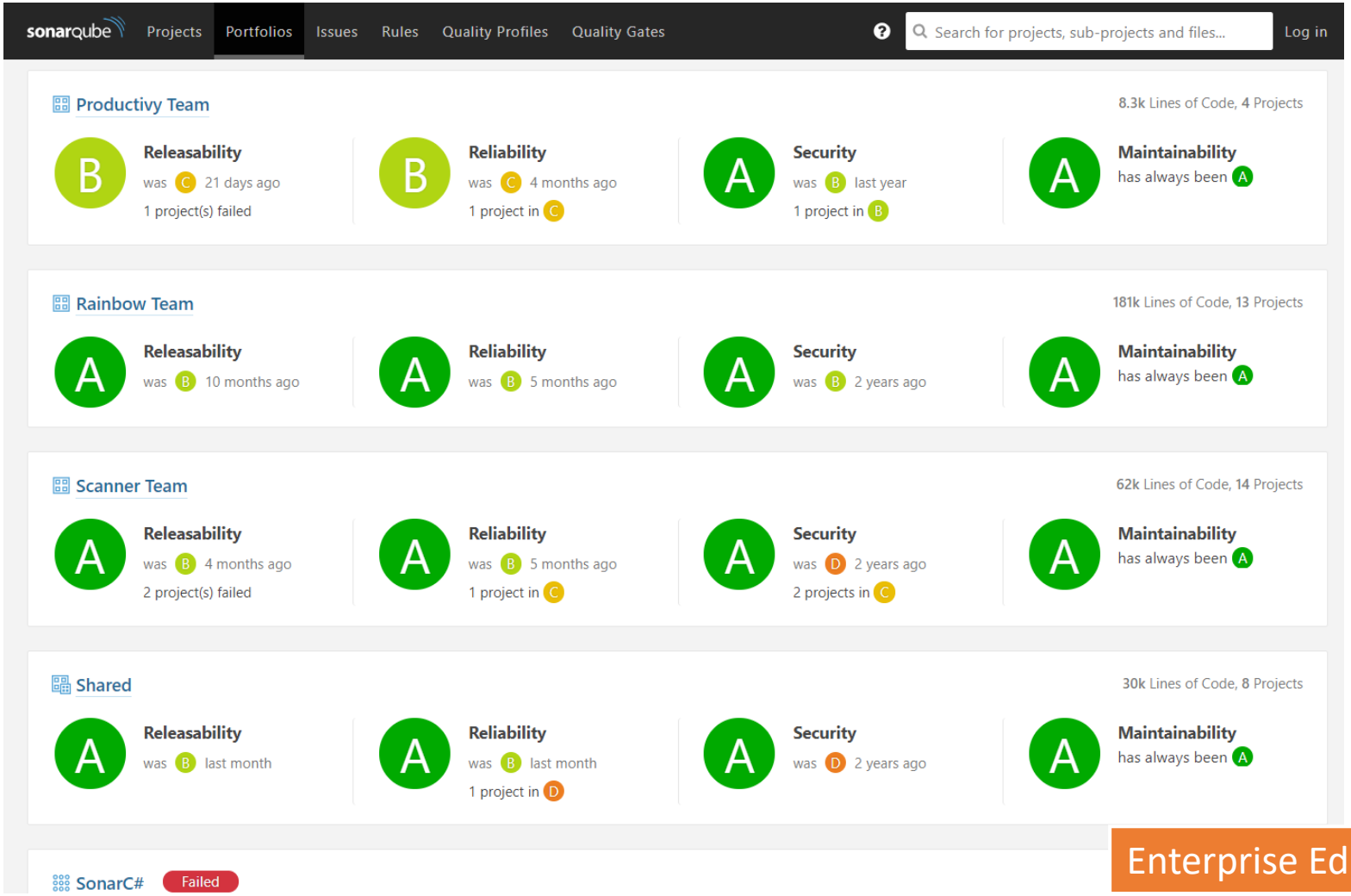
Branch 기반으로 개발을 한다면 SonarQube에서 Short-lived와 Long-lived 코드 브랜치의 품질을 추적할 수 있습니다. 안정성이 보장된 코드만이 master로 merge될 수 있습니다.



28

SonarQube > Portfolios 1/2

SonarQube Portfolios를 통해 팀별 혹은 제품별로 원하는 형태의 포트폴리오를 구성할 수 있습니다. 품질 게이트를 실패한 프로젝트의 비율은 Releaseability로 표시됩니다. 이를 통해 릴리즈를 결정할 수 있습니다.

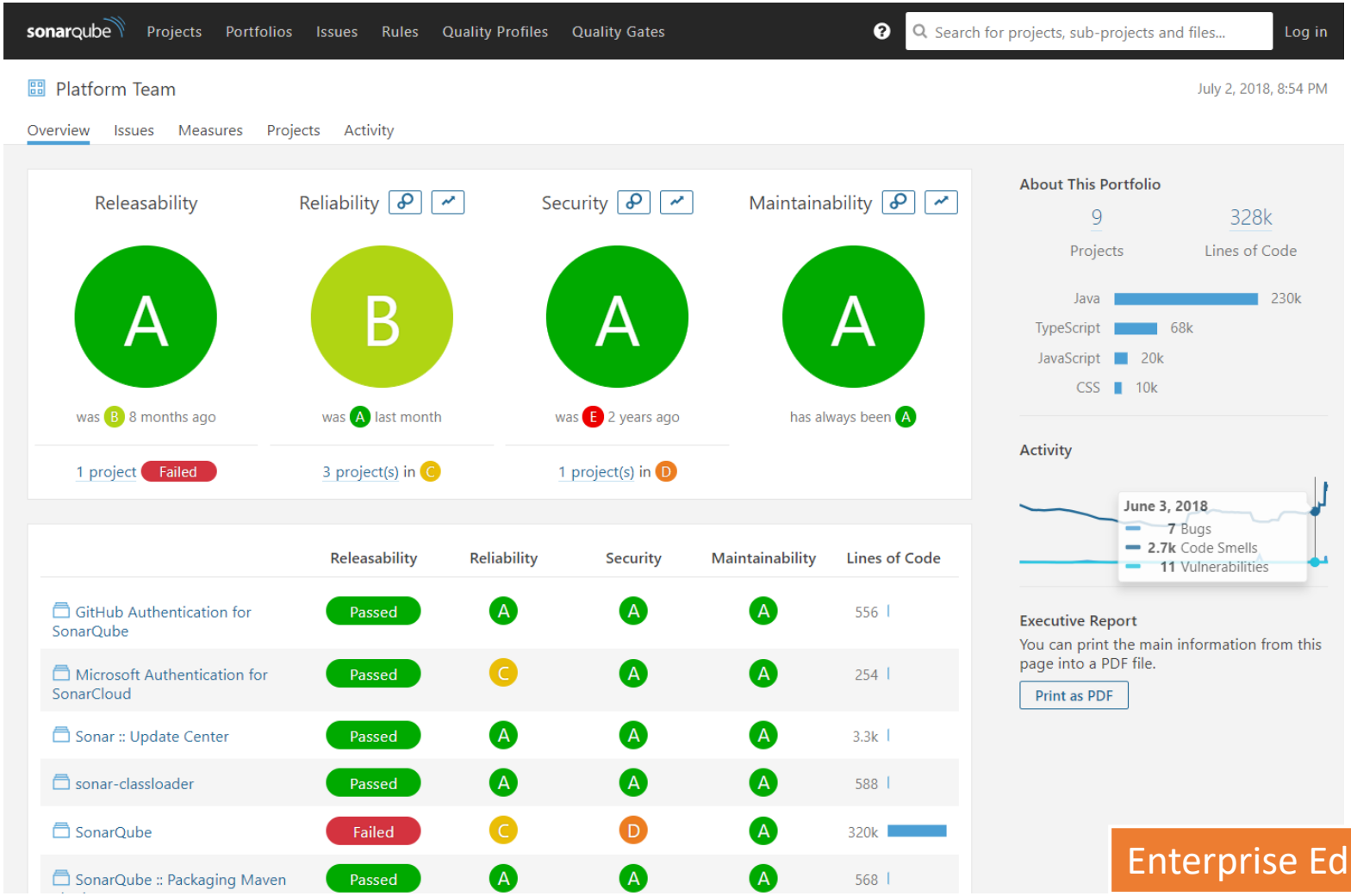


Enterprise Edition

29

SonarQube > Portfolios 2/2

SonarQube Portfolios를 통해 팀별 혹은 제품별로 원하는 형태의 포트폴리오를 구성할 수 있습니다. 품질 게이트를 실패한 프로젝트의 비율은 Releaseability로 표시됩니다. 이를 통해 릴리즈를 결정할 수 있습니다.



Enterprise Edition

SonarQube > Integration

SonarQube는 인기있는 Build Systems, CI Engine, IDE와 통합을 제공하며, SonarQube의 Web API와 Web Hook을 통해 원하는 시스템과의 통합을 수행할 수 있습니다.

Build Systems과 통합

- 표준 빌드 시스템과 강력하게 통합되어 Zero-Configuration을 제공합니다.
- Maven, MSBuild, Gradle, ANT와 같은 인기있는 빌드 시스템과 무설정 혹은 약간의 설정으로 프로젝트를 빠르게 스캔할 수 있습니다.

CI Engines과 통합

- 빌드 시스템과 쉬운 통합은 CI 엔진과도 쉽게 통합됨을 의미합니다.
- Bamboo, Travis CI, Jenkins, AppVeyor, Team Foundation Server, TeamCity와 같은 다양한 CI 엔진에서 원클릭 만으로 SonarQube를 연동할 수 있습니다.

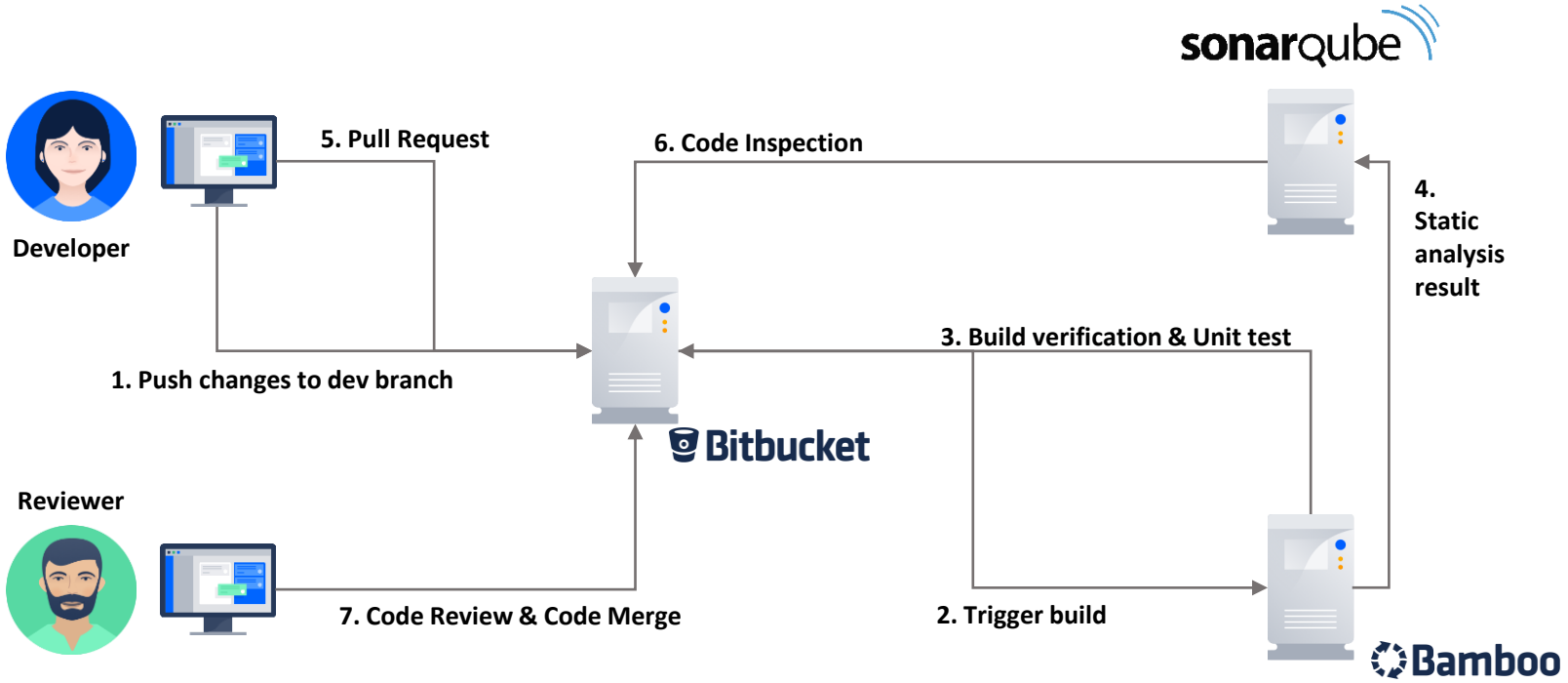
IDE와 통합

- 개발자는 자신이 좋아하는 IDE에서 코드 품질에 대한 피드백을 얻을 수 있습니다.
- SonarLint는 이클립스, Intelli J, 비주얼 스튜디오, 비주얼 스튜디오 코드, Atom과 같은 IDE에서 실시간으로 코드 품질에 대한 피드백을 제공합니다.

Web API와 Web Hook

- SonarQube의 웹 API를 사용하여 SonarQube를 자동적으로 프로비저닝하거나 BI에 원하는 데이터를 제공하거나 모니터링 할 수 있습니다.
- 표준 웹 훅(Webhook) 메커니즘을 사용하여 채팅방과 같은 외부 시스템에 알림을 수행할 수 있습니다.

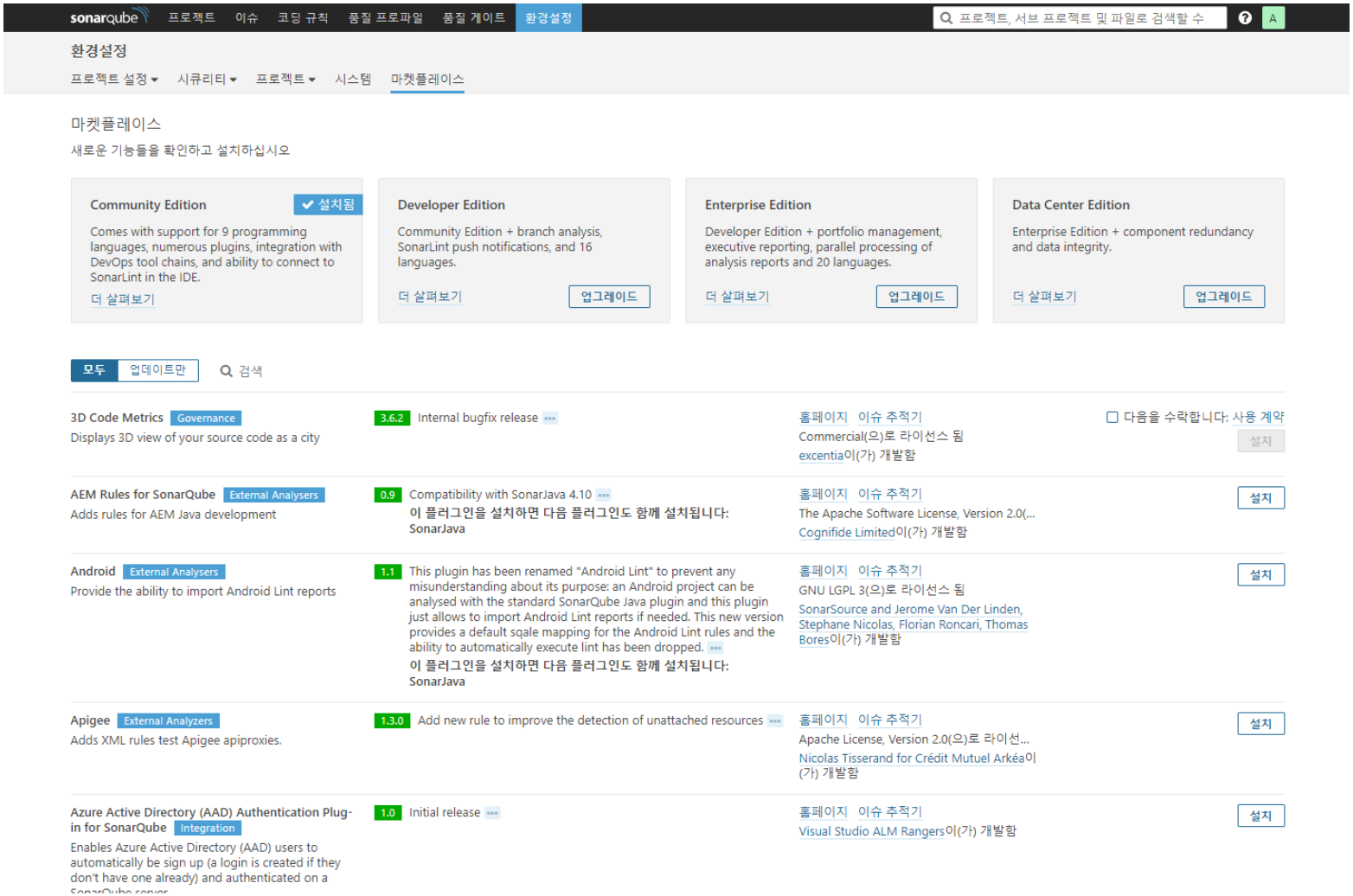
다음 그림은 Atlassian Bitbucket, Bamboo의 Continuous Integration과 통합된 SonarQube 구성 방안을 보여주고 있습니다.



32

SonarQube > Marketplace

SonarQube의 Marketplace를 통해 SonarQube 기능 향상을 지원하는 75개의 플러그인을 다운로드 받거나 설치할 수 있으며, 이러한 플러그인은 추후 추가될 예정입니다.



33

SonarQube Edition & Event

SonarQube에 관심있으신 모든 분에게 평가판 라이선스를 제공합니다.

Community Edition

85,000 회사에서 사용중

- ✓ SonarQube & 60+ 플러그인
- ✓ SonarLint
- ✓ 15 Languages

Free

Developer Edition

개발자를 위한 버전

- ✓ Community 포함
- ✓ 브랜치 분석
- ✓ SonarLint 알림
- ✓ 22 Languages
- ✓ 하나의 테스팅 및 스테이징 인스턴스 제공

20만원부터

Enterprise Edition

엔터프라이즈를 위해

- ✓ Developer 포함
- ✓ 포트폴리오 관리
- ✓ Executive 리포팅
- ✓ 27 Languages
- ✓ 전문가 기술 지원 포함 (30M부터)

2,100만원부터

Data Center Edition

고가용성을 위한

- ✓ Enterprise 포함
- ✓ 컴포넌트 리더던시
- ✓ 데이터 무결성
- ✓ 3개의 추가적인 라이센스

13,500만원부터

* 각 분석 된 프로젝트의 LOC(Line Of Code)를 합산하여 계산

34

Thank you!

Contact
curvc@curvc.com
02-6245-5478

35

SonarQube Price

sonarqube

전체 Code Line수로 SonarQube의 가격정책이 변경됩니다.

단위 (원)

Developer Edition		Enterprise Edition		Datacenter Edition	
100K	250,000	1M	**24,000,000**	20M	**150,000,000**
250K	1,350,000	5M	38,000,000	50M	225,000,000
500K	2,700,900	10M	56,250,000	75M	300,000,000
1M	4,500,000	20M	75,000,000	100M	375,000,000
2M	9,000,000	30M	112,500,000	250M	600,000,000
5M	26,250,000	50M	135,000,000	500M	900,000,000
10M	54,000,000	75M	187,500,000		
20M	75,000,000	100M	270,000,000		

환율에 따라 해당 가격은 차이가 날 수 있습니다. 현재 가격은 참고용 가격입니다.

36

curvc

SonarCFamily for C/C++

C++		C	
Rules	316	Rules	226
Bugs	69	Bugs	56
Code Smells	245	Code Smells	168
Vulnerabilities	2	Vulnerabilities	2
CWE	27	CWE	26
SANS Top 25	2	SANS Top 25	2
MISRA	119	MISRA	111
CERT	89	CERT	69
• Memory Leak • Dead Code • Logic Flow Error • Dereference of Null Pointers • Coding Conventions • Error Handling		• Memory Leak • Dead Code • Logic Flow Error • Dereference of Null Pointers • Coding Conventions • Error Handling	

37

SonarCFamily for C/C++

- Supported Compilers, Language Standards and Operating Systems
  - Any version of Clang, GCC and Microsoft C/C++ compilers
  - Any version of Intel compiler for Linux and OS X
  - ARM5 and ARM6 compilers
  - Compilers based wholly on GCC including for instance Linaro GCC and WindRiver GCC are also supported
  - C89, C99, C11, C++03, C++11 and C++14 standards
  - Microsoft Component Extensions for runtime platforms : C++/CLI and C++/CX
  - GNU extensions
  - Microsoft Windows, Linux and Mac OS X for runtime environment
- Metrics
  - Code coverage by tests: SonarCFamily for C/C++ support the import of MS visual studio and GCOV coverage reports along with the import of CPP unit report.
- Custom Rules
  - SonarCFamily for C/C++ doesn't yet provide the ability to write custom rules.
- CWE compatibility
  - SonarCFamily for C/C++ is officially registered as CWE compatible.

38