

1. SonarQube 소개

이 문서는 SonarQube를 사용하기 위해 알아야 할 내용을 간단히 소개 하기 위해 작성되었다.

- SonarQube는
- SonarQube 주요 특징
- SonarQube 데모 사이트
- SonarQube 소프트웨어 품질
 - 1.Code Smells(Maintainability)
 - 2.Bugs(Reliability)
 - 3.Vulnerabilities(Security)
 - 4.Duplications
 - 5.Unit Tests
 - 6.Complexity
 - 7.Size
- 주요 용어

SonarQube는

SonarQube는 지속적인 인스펙션 패러다임을 지원하는 최상의 제품입니다. 지속적인 코드 인스펙션은 소프트웨어 개발 라이프사이클의 완전한 부분으로 내부 소프트웨어 품질 향상 시킬 수 있는 코드 품질 관리 패러다임입니다.

지속적인 코드 품질 관리를 통해 개발 초기에 코드 문제를 해결하고 자동화된 코드 감사를 통해 최신 상태의 코드 문제를 확인할 수 있게 해줍니다.

SonarQube 주요 특징

- 지속적인 인스펙션
 - 지속적인 통합과 같이 빌드와 연동하여 지속적으로 코드에 대한 인스펙션을 수행합니다.
- 품질 중앙화
 - 개발 조직의 코드 품질을 중앙 저장소에서 가시화하고 단일 위치에서 관리합니다.
- DevOps와의 통합
 - 다양한 빌드 시스템, CI 엔진과 통합되어 DevOps 실적을 지원합니다.
- 품질 요구사항 설정
 - 품질 게이트를 통해 표준화된 코드 품질 요구사항을 설정합니다.
- 다중 언어 분석
 - 20개가 넘는 프로그램 언어에 대한 코드 분석을 지원합니다.
- 플러그인을 통한 확장
 - 다수의 플러그인을 통해 SonarQube의 기능을 확장할 수 있습니다.

SonarQube 데모 사이트

- 클라우드 사이트를 통한 데모
 - <https://sonarcloud.io/projects>
- 최신 버전의 SonarQube 체험
 - <https://next.sonarqube.com/sonarqube/projects>

SonarQube 소프트웨어 품질

SonarQube는 다음 7가지 품질 요소를 기준으로 소프트웨어 품질을 구분하고 관리하고 있다.

- Code Smells(Maintainability)
- Bugs(Reliability)
- Vulnerabilities(Security)
- Coverage
- Duplications
- Size
- Complexity

1.Code Smells(Maintainability)

심각한 이슈는 아니지만 베스트 프랙티스에서 사소한 이슈들로 모듈성(modularity), 이해가능성(understandability), 변경가능성(changeability), 테스트용의성(testability), 재사용성(reusability) 등이 포함된다.

- Code Smells
- Debt
- Debt Ratio

2. Bugs(Reliability)

일반적으로 잠재적인 버그 혹은 실행시간에 예상되는 동작을 하지 않는 코드를 나타낸다.

- Code Smells
- Debt
- Debt Ratio

3. Vulnerabilities(Security)

해커들에게 잠재적인 약점이 될 수 있는 보안상의 이슈를 말한다. SQL 인젝션, 크로스 사이트 스크립팅과 같은 보안 취약성을 찾아낸다.

- Code Smells
- Debt
- Debt Ratio

4. Duplications

코드 중복은 코드의 품질을 저하시키는 가장 큰 요인 중 하나이다.

- Density
- Duplicated Lines
- Duplicated Blocks
- Duplicated Files

5. Unit Tests

단위테스트 커버리지를 통해 단위 테스트의 수행 정도와 수행한 테스트의 성공/실패 정보를 제공한다.

Unit Test Coverage

- Line Coverage : 구문 커버리지
- Condition Coverage : 조건 커버리지

Unit Test Result

- Unit Tests : 단위 테스트 수
- Failures : 실패한 단위테스트 코드 수
- Errors : 에러가 발생한 단위 테스트 코드 수
- Skipped : 스킵한 단위 테스트 코드 수
- Success : 단위 테스트 성공률
- Duration : 단위 테스트 수행에 소요된 시간

6. Complexity

순환복잡도(Cyclomatic Complexity) 측정

- 코드의 논리적인 흐름상에 존재하는 독립적인 선형 경로의 개수를 의미

인지복잡도(Cognitive Complexity) 측정

7. Size

소스코드 사이즈와 관련된 다양한 지표를 제공합니다.

- 코드 라인수
- 전체 라인수
- 구문
- 함수
- 클래스
- 파일
- 디렉토리
- 주석 수
- 코멘트 비율

주요 용어

Continuous Inspection

- 지속적으로 소스 코드를 분석하고 보고 함으로써 품질 트렌드를 파악하고 과거 대비 개선 여부를 파악하고 발생한 이슈를 처리의 반복

Technical Debt

- 최적의 개발 상태를 유지하기 위해서 제거해야 하는 변경 비용의 격차를 의미

Issues

- 소프트웨어 품질을 개선할 수 있는 항목을 이슈라는 최소 단위로 관리

Quality Metric

- 품질에 영향을 미치는 특징을 가진 요소를 의미하며 소프트웨어 품질 특성들을 정량적으로 표시하기 위한 기준

Quality Measure

- 품질에 영향을 미치는 특징을 가진 요소들을 정량적으로 측정할 수치