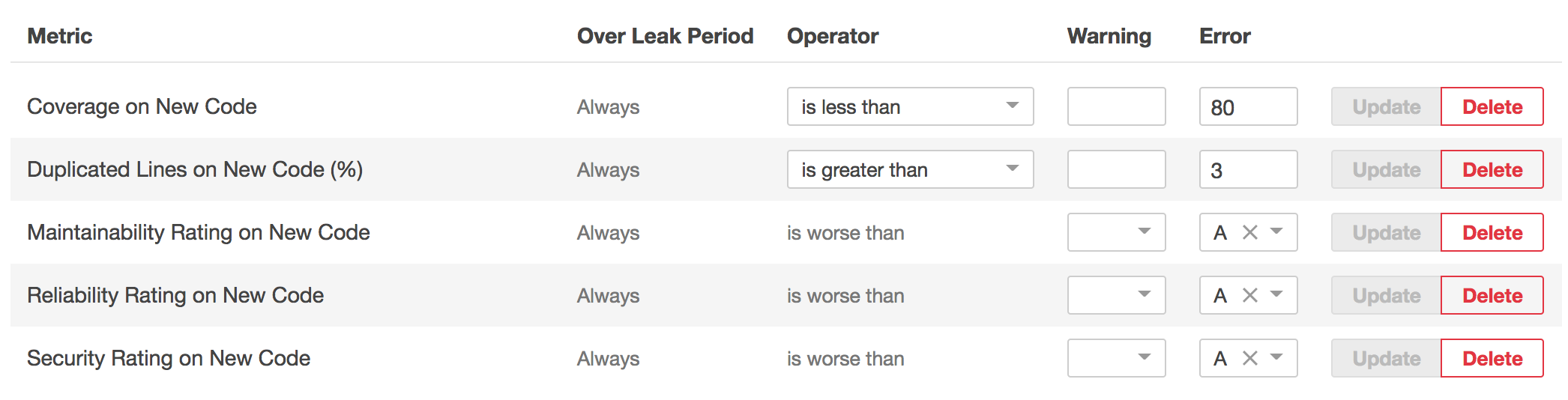
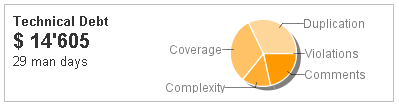
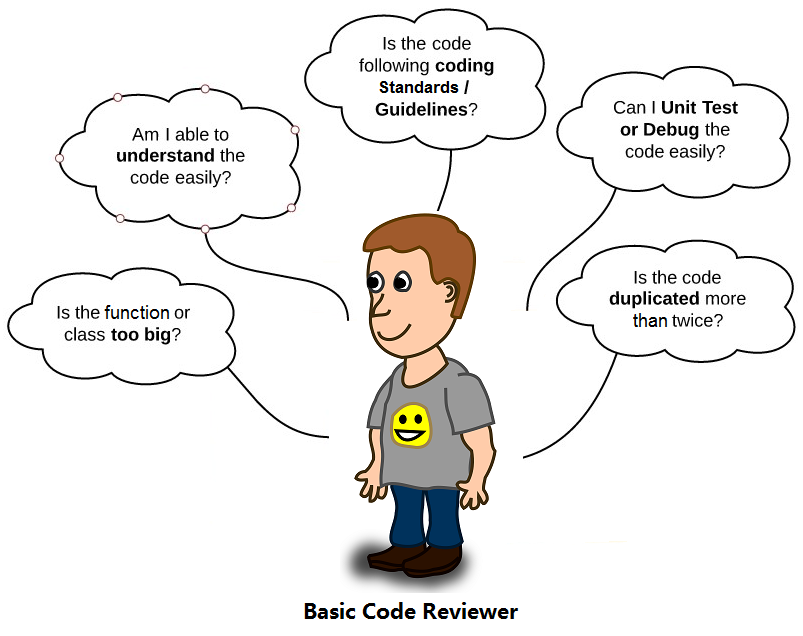
**SonarQube Metrics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Key** | **Reason** | **Description** |
| 1 | **Complexity**  **Cognitive Complexity** | complexity  cognitive\_complexity | JAVA: IF, FOR, WHILE, CASE, CATCH, THROW  .NET: If, Case, Continue, End, Resume, Error, GoTo | Whenever the control flow of a function splits, the complexity counter gets incremented by one. Each **function**has a minimum complexity of 1.  How hard it is to understand the code's control flow |
| 2 | **Duplications** | Duplicated\_blocks  duplicated\_files  duplicated\_lines  duplicated\_lines\_density   |  |  | | --- | --- | |  |  | | * There should be at least 10 successive and duplicated statements whatever the number of tokens and lines. | Number of duplicated blocks, files of lines.  Density of duplication = **Duplicated lines** / **Lines**\* 100 |
| 3 | **Issues** | Violations  open\_issues  reopened\_issues  confirmed\_issues |  | Number of issues.  whose status is Open  whose status is Reopened  whose status is Confirmed |
| 4 | **Severity** | Blocker (Operational/Security risk)  Critical  Major  (Productivity)  Minor  Info | * Calling garbage collector, not closing a socket, etc. * NullPointerException, badly caught exceptions, lack of unit tests, etc.   Too complex methods, package cycles, etc.  naming conventions  not yet well defined security risk |  |
| 5 | **Maintainability**  **Technical Debt** | Code\_Smells  new\_code\_smells  sqale\_index | * Number of code smells. * Number of new code smells.   Effort to fix all maintainability issues. | The measure is stored in minutes in the DB. An 8-hour day is assumed when values are shown in days. |
| 7 | **Quality Gates**  **QG Status**  **QG Details** | alert\_status  quality\_gate\_details | * ERROR, WARN, OK | State of the Quality Gate associated to your Prjct  For all the conditions of your Quality Gate, you know which condition is failing and which is not. |
| 8 | **Reliability**  **Bugs**  **New Bugs**  **Reliability Rating** | bugs  new\_bugs  reliability\_rating | Number of bugs.  Number of new bugs.  A = 0 Bug B = at least 1 Minor Bug C = at least 1 Major Bug D = at least 1 Critical Bug E = at least 1 Blocker Bug | Effort to fix all bug issues. The measure is stored in minutes in the DB. An 8-hour day is assumed when values are shown in days. |
| 9 | **Security**  **Vulnerabilities**  **New Vulnerabilities**  **Security Rating** | Vulnerabilities  new\_vulnerabilities  security\_rating | Number of vulnerabilities.  Number of new vulnerabilities.  A = 0 Vulnerability B = at least 1 Minor Vulnerability C = at least 1 Major Vulnerability D = at least 1 Critical Vulnerability E = at least 1 Blocker Vulnerability | Effort to fix all vulnerability issues. The measure is stored in minutes in the DB. An 8-hour day is assumed when values are shown in days. |
| 10 | **Size**  **Classes**  **Comment lines**  **Directories**  **Files**  **Lines**  **Lines of Code** | Classes  comment\_lines  directories  files  lines  ncloc | Number of classes  Number of lines containing.  Number of directories.  Number of files.  Number of physical lines |  |
| 11 | **Tests**  **Skipped unit tests**  **Unit tests**  **Unit tests errors**  **Unit test failures** | skipped\_tests  tests  test\_errors  test\_failures | Number of skipped unit tests.  Number of unit tests.  Number of unit tests that have failed.  Unit tests that have failed with an unexpected exception. |  |





**Basic Code Review Checklist:**



### Detailed Code Review Checklist: Code formatting, Architecture, Coding best practices, Non-functional requirements, Reusability, Reliability, Extensibility, Security, Performance, Scalability, Usability, and OOAD Principles

