European Maritime Safety Agency

Appendix 2
to Tender Specifications

EMSA Quality Gate for Java projects

EMSA

SonarQube is used for quality checking. EMSA uses a special Java Quality Gate based on "Sonar way" quality profile.

Fail to pass this Quality Gate will imply the rejections of the version being delivered.
All projects (new ones and existent ones) will be submitted to the Quality Gate. For existent projects an adoption plan may be established and agreed with EMSA. However, "**Blocking issues**" shall not be accepted.

The following rules summarize the initial Quality Gate:

- **Blocking issues**
  - Deliveries containing blocking issues shall not be accepted. The following issues are considered blocking:
    - Avoid Decimal Literals In Big Decimal Constructor
    - **Deprecated**: Avoid Print Stack Trace. Replaced by …
    - Throwable.printStackTrace(...) should never be called
    - Big Integer Instantiation
    - **Deprecated**: Broken Null Check. Replaced by …
    - Short-circuit logic should be used to prevent null pointer dereferences in conditionals
    - Class defines equal(Object); should it be equals(Object)
    - **Deprecated**: Class defines hashcode(); should it be hashCode(). Replaced by …
    - Methods should not be named "hashcode"
    - Class defines tostring(); should it be toString()?
    - Correctness - A known null value is checked to see if it is an instance of a type
    - Correctness - close() invoked on a value that is always null
    - Correctness - equals method always returns false
    - Correctness - equals method always returns true
    - **Deprecated**: Correctness - equals(...) used to compare incompatible arrays. Replaced by …
    - The Array.equals(Object obj) method should never be used
    - Correctness - Impossible cast
    - Correctness - Impossible downcast
    - Correctness - Impossible downcast of toArray() result
    - Correctness - Null value is guaranteed to be dereferenced
    - **Deprecated**: Equals Hash Code. Replaced by …
    - "equals(Object obj)" and "hashCode()" should be overridden in pairs
    - Integer Instantiation
    - Multithreaded correctness - Call to static Calendar
    - Multithreaded correctness - Call to static DateFormat
    - Performance - Maps and sets of URLs can be performance hogs
    - Performance - The equals and hashCode methods of URL are blocking
    - Preserve Stack Trace
    - Security - Hardcoded constant database password

        - IP addresses should not be hardcoded (new)
        - String Instantiation
        - String To String
        - **Deprecated**: System Println. Replaced by …
        - System.out and System.err should not be used as loggers
        - **Deprecated**: Bad practice - Method invokes System.exit(...). Replaced by …
        - System.exit(...) and Runtime.getRuntime().exit(...) should not be called
        - **Deprecated**: Unused Private Field. Replaced by …
        - Unused private fields should be removed
        - Useless Operation On Immutable
        - **NEW:** Security - Array is stored directly
        - **NEW:** Credentials should not be hard-coded
        - **NEW:** Values passed to OS commands should be sanitized
        - **NEW:** Values passed to SQL commands should be sanitized
        - **NEW:** Values passed to LDAP queries should be sanitized
        - **NEW:** "static final" arrays should be "private"

- **Critical issues**
    - Might be accepted but
        - have to be justified
        - A correction target date or version as to be defined

      Depending on the situation, justification can be:
        - Justified only for a period of time and correction target date/version defined or…
        - Justified and permanently accepted.
    - Critical issues cannot increase from version to version

- **Major issue**
    - Major issues cannot increase from version to version

- **Cyclomatic complexity**
    - Max is set to 25
    - Cyclomatic complexity index over 25 might be accepted but:
        - have to be justified
        - refactoring have to be planned to a next version or date

      Depending on the situation, the justification can be:
        - Justified only for a period of time and correction target date/version defined or…
        - Justified and permanently accepted.
    - Cyclomatic complexity index cannot increase from version to version:

- **Duplications**
    - Duplications shall be lower than 7.5%

- Duplications cannot increase from version to version
- Auto-generated classes/code can be ignored

- **Documentation and Comments**
  - Code comments should be greater than 20%
  - Public API Documentation should be greater than 50%
  - Both indicators cannot decrease from version to version
  - EMSA shall manually and randomly assess source code files to validate the quality and usefulness of the documentation and comments.

- **Unit Tests Coverage**
  - New starting projects shall have a minimum of 25% test coverage
  - For existent project, a minimum increase of 5% per major version is required.
  - Test coverage cannot decrease from version to version

Quality Gate shall be mandatory for all projects. As for any other project task, a Quality Gate will consume effort and time. Contractors are encouraged to adopt continuous and rigorous quality checking measures during the development process and submit each version to EMSA Quality Gate before delivery.