EBRAINS Software Quality Guideline

Alan B Stokes[^MAN] Daniel Keller[^EPFL]
Daviti Gogshelidze[^FZJINM1] Dennis Terhorst[^FZJINM6]
Eric Müller[^HEIDELBERG] George Andreou[^ATHENA]
James Gonzalo King[^EPFL] Orfeas Aidonopoulos[^ATHENA]
Sandra Diaz[^FZJSimlab] Thorsten Hater[^FZJSimlab]

Contents

Checklist

Quick Summary

This section can act as a quick reference, be used for ESQ-guideline compliance checks, or overview for developers which aspects of software quality may need consideration.

To check/validate compliance with this guideline, the following checklist should provide a quick and brief overview. Ideally the validation can be technically facilitated by frameworks like the Core-Infrastructure Badge [71].

The following items should provide a quick overview for developers and for validating guideline compliance of a tool.

The requirement levels of these points are marked by color:

- Passing EBRAINS Software Quality checks: all required items fulfilled.
- Silver EBRAINS Software Quality level: all required and suggested items fulfilled.
- Gold EBRAINS Software Quality level: all required, suggested and optional items fulfilled.

Metadata

Software	NEST Simulator			
Version	NEST Simulator			
Submitter	NEST Simulator	Date	NEST Simula	(yyyy-mm-dd)
Curator (TC)	NEST Simulator	Date	NEST Simula	(yyyy-mm-dd)

Dependency Management

- [N] Software package-, API-, data-type- and service dependencies must be explicitly specified in terms of version constraints and feature variants. [deps-well-defined]
- [N] Software package-, API-, data-type- and service dependency information must be included in every release. [deps-per-release]
- [N] Software package dependencies should be tracked and handled by a software tool. [deps-not-manual]

Software Project Management

- [N] Tools should use an open issue/bug tracking system accessible also by the users.
- [N] Sufficient developer coverage of all parts of the code base must be sustained. Having only a single person with the full knowledge of parts or complete software tools should be avoided.
- [N] All EBRAINS components must have on-boarding documentation for new developers, testers and administrators following the documentation guidelines.
- [N] Communication between and among users and developers should be facilitated.
- [N] All EBRAINS tools should be architecturally designed to comply with the EBRAINS APIs and to be easily integrated with other EBRAINS components (ref. EIA-WG).

- [N] All EBRAINS tools should have a development road-map.

Version Control

- [N] Software packages must specify a upstream repository location (which may be non-public).
- [N] Software packages must specify a development and code integration workflow.
- [N] Software packages must explicitly define a versioning scheme.
- [N] Stable releases should bear a defined version number (ref. Dependency Management).
- [N] It may be required that repositories that contain software releases should be mirrored to the trusted EBRAINS git service.

Testing

- [N] Tools should define and regularly run automated unit and integration tests.
- [N] Tools may use a combination of different CI providers (e.g. for different sets of tests).
- [N] Components must Implement and provide pass results for
- [N] Developers must ensure that both technical and scientifically relevant functionality is covered by the test suits.
- [N] For software offered as a service: security, access control, load and stress tests should be also provided.
- [N] For software which combines multiple EBRAINS components, tests on the interfaces between tools are mandatory.
- [N] For software deployed on HPC resources, tests should consider the latest deployed underlying software modules and dependencies at least on the primary and backup system. (t.b.d.)

Documentation

- [N] EBRAINS software must have user-level documentation.
- [N] EBRAINS software must have developer-level documentation.
- [N] EBRAINS software must have maintenance/operations documentation.
- [N] Follow EBRAINS Documentation Guidelines
- [N] All forms of documentation must be actively maintained.
- [N] User-level documentation must enable users to find, install and use the software/tool and understand it's operations and fundamental assumptions.
- [N] Maintenance/Operations Documentation must enable users without internal knowledge of the tool to deploy and provide the tool and to maintain basic development structures (building documentation, updating packages/dependencies, install and deploy containers, etc.)
- [N] Software should have deprecation notes and avoid direct removes
- [N] Software should have an internal architecture view of the software (see Project management)
- [N] Documented coding guidelines and stick to conventions (cf. Testing)

- N Documented the integration workflow (e.g. automatic tests, CR, approval processes)
- N Projects should define a code-of-conduct and requirements for contributing (e.g. CLA, cf. Licensing)
- N Project management conventions should be documented (development workflow, code review guidelines)

Code Quality

- N Software packages must explicitly specify a code style to adhere to.
- N Software packages should specify a set of linters that are regularly used. (t.b.d.)
- N Software packages must explicitly specify a documentation standard to adhere to. (t.b.d.)
- N The software package build process should include building the documentation.
- N Software packages should provide documentation about how to apply typical dynamic analysis tools.
- N The build process must build and execute all standalone tests. (t.b.d.)
- N Software packages should describe the code structure. (t.b.d.)
- N Software packages should define a workflow for handling technical debt and to ensure sustainability. (t.b.d.)

Deployment Plans and Continuous Deployment (CD)

- N Releases may be required to be made available through standard mechanisms or in standard locations (ESD-WG)
- N Releases may be required to follow procedures and use mechanisms defined by the EIS-WG (t.b.d.)
- N Releases must be accompanied by adequate (t.b.d) release notes for users and operators.
- N Software packages may be required to have installation tests (e.g. “make installcheck”) (t.b.d.)

Licensing

- N Chosen license should be an open-source license (compatibility with EBRAINS t.b.d.)
- N Requirements/conditions for contributions to the project must be clear (e.g. contributor agreement)
- N (EBRAINS) Software must have a defined license
- N Software package license should be identifiable in a standardized way (t.b.d.)
- N Software should consider General Data Protection Regulations [70] (GDPR)

UNKNOWN

- [N] Service users should be informed about service changes sufficient time before change happens on the production (Users should have time to make necessary changes)
- [N] Service users as well as end users (If it is possible) should be notified before planned upgrades and planned “going offline”