

# EBRAINS Software Quality Guidelines

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

2

# 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.

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

- P: to be filled by Package manager / developer
- R: to be filled by Release manager / technical coordination

Submitter(P)	<input type="text"/>	Date	<input type="text"/>	(yyyy-mm-dd)
Software(P)	<input type="text"/>			
Version(P)	<input type="text"/>			
Curator (R)	<input type="text"/>	Date	<input type="text"/>	(yyyy-mm-dd)
Result (R)	<input type="text"/>			

## Dependency Management

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

## Software Project Management

- ☐ [proj-public-issues] Tools should use an open issue/bug tracking system accessible also by the users. (bool)
- ☐ [proj-dev-coverage] 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. (bool)
- ☐ [proj-onboarding] All EBRAINS components must have on-boarding documentation for new developers, testers and administrators following the documentation guidelines. (bool)
- ☐ [proj-communication] Communication between and among users and developers should be facilitated. (bool)

- ☐ [proj-comply-eia] 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). (bool)
- ☐ [proj-roadmap] All EBRAINS tools should have a development road-map. (bool)
- ☐ [gdpr-compliant] Software must provide General Data Protection Regulations (GDPR) compliance. (bool)

#### Version Control

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

#### Testing

- ☐ [test-auto-run] Tools should define and regularly run automated unit and integration tests. (bool)
- ☐ [test-many-providers] Tools may use a combination of different CI providers (e.g. for different sets of tests). (bool)
- ☐ [test-pass] Components must Implement and provide pass results for a) integration tests, b) system tests and c) (optionally) acceptance tests. (bool)
- ☐ [test-tech-and-science] Developers must ensure that both technical and scientifically relevant functionality is covered by the test suits. (bool)
- ☐ [test-services] For software offered as a service: security, access control, load and stress tests should be also provided. (bool)
- ☐ [test-interfaces] Software which can be combined with other EBRAINS components, must provide tests for the interoperability or compliance to interface definitions. (bool)
- ☐ [test-hpc-stages] 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.) (bool)

#### Documentation

- ☐ [doc-users-level] EBRAINS software must have user-level documentation. (bool)
- ☐ [doc-developers-level] EBRAINS software must have developer-level documentation. (bool)
- ☐ [doc-operations-level] EBRAINS software must have maintenance/operations documentation. (bool)
- ☐ [doc-maintained] All forms of documentation must be actively maintained. (bool)

- ☐ [doc-user-enabling] User-level documentation must enable users to find, install and use the software/tool and understand it's operations and fundamental assumptions. (bool)
- ☐ [doc-maint-enabling] 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.) (bool)
- ☐ [doc-deprecations] Software should have deprecation notes and avoid direct changes. (bool)
- ☐ [doc-architecture] Software should have an internal architecture view of the software (see Project management) (bool)
- ☐ [doc-conventions] Components should provide documented coding guidelines and stick to conventions (cf. Testing) (bool)
- ☐ [doc-workflow] Components should provided documentation for the integration workflow (e.g. automatic tests, CR, approval processes) (bool)
- ☐ [doc-conduct] Projects should define a code-of-conduct and requirements for contributing (e.g. CLA, cf. Licensing) (bool)
- ☐ [doc-project-management] Project management conventions should be documented (development workflow, code review guidelines) (bool)

#### Code Quality

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

#### Deployment Plans and Continuous Deployment (CD)

- ☐ [cd-standards] Releases may be required to be made available through standard mechanisms or in standard locations (the `ebrains-spack-builds` repository) (bool)
- ☐ [cd-secured] Releases may be required to follow procedures and use mechanisms defined by the EIS-WG (t.b.d.) (bool)
- ☐ [cd-release-notes] Releases must be accompanied by adequate (t.b.d) release notes for users and operators. (bool)

- ☐ [cd-testable] Software packages may be required to have installation tests (e.g. “make installcheck”) (t.b.d.) (bool)

#### Licensing

- ☐ [license-open-source] Chosen license should be an open-source license (compatibility with EBRAINS t.b.d.) (bool)
- ☐ [license-contributing] Requirements/conditions for contributions to the project must be clear (e.g. contributor agreement) (bool)
- ☐ [license-defined] (EBRAINS) Software must have a defined license (bool)
- ☐ [license-identifiable] Software package license should be identifiable in a standardized way (bool)