Commission d'Evaluation

Inria Evaluation Committee
Criteria for Software Self-Assessment V3

January 2021

The EC considers software a major research outcome of the Institute. However, experience has shown that software descriptions in the team descriptions and the candidate application files are very hard to use, because team members and applicants are much less used to self-assessment of software than to self-assessment of theoretical contributions and publications.

The goal of this document is to provide Inria teams and candidates with a self-assessment mechanism for their software developments, to be used by the Inria Evaluation Committee (EC) for evaluation seminars and internal or external recruitments/promotions. It is **a request for evaluation** that the juries and evaluation panels will use to select the appropriate criteria for evaluating software applications developed by teams and candidates. What the EC wants is a fair self-assessment of the real state, evolution, and impact of the software, which in principle would prove exact in a subsequent in-depth evaluation. Examples of some Inria Software Artifacts.

- OCaml: `Family=research; Audience=community; evolution=lts; Duration>=20; contribution=devel,softcont; Url=https://caml.inria.fr/ocaml/`
- Flyspell.el: `Family=utility; Audience=universe; evolution=basic; Duration>=20; contribution=leader; Url=https://www.emacswiki.org/emacs/FlySpell`

The software self-assessment should be followed (Section *Free Description*) by a brief description of the software you have contributed to and the nature of your contributions.

1 Software Family (Family)

The software families are described in the EC document "Software Artifact Evaluation", referred to as SAE: `https://hal.inria.fr/hal-03110723/document` (in French) and `https://hal.inria.fr/hal-03110728/document` (in English).

1. `research`: Software as a Vector for Knowledge (see SAE, Section 3.1).
2. `vehicle`: Software as a Vehicle for Research (see SAE, Section 3.2).
3. `transfer`: Transfer software, (see SAE, Section 3.3).
4. `utility`: Utility, (see SAE, Section 3.4).

2 Audience (Audience)

1. `personal`: personal or internal team prototype (to experiment an idea);
2. `team`: to be used by people in the team or close to the team (including contractual partners);
3. `partners`: to be used by people inside and outside the team but without a clear and strong dissemination and support action plan;
4. `community`: large audience software, usable by people inside and outside the field with a clear and strong dissemination, validation, and support action plan;
5. `universe`: wide-audience software (aims to be usable by a wide public, to become the reference software in its area, etc.).

3 Evolution and Maintenance

- `nofuture`: no real future plans;
- `basic`: basic maintenance to keep the software alive;
- `lts`: long term support.

4 Duration of the Development (Duration)

Indicate here the number of years of your contribution to the software development.

5 Contribution Characterization

Characterize your contributions to the development. More than one contribution are possible, in particular if the nature of your contribution has evolved over the years.

- `leader`: you have been at the initiative of the development and you have been involved in the coding, debugging, and maintenance.
- `instigator`: you have been at the initiative of the development but you have not been involved in the coding.
- `devel`: you have been deeply involved in the coding.
- `softcont`: you have contributed to the documentation, maintenance, installation scripts, ...

Supervising activities must be reported in dedicated section (Form 1) in the application form.

6 Web Page (Url)

Indicate the software artifact URL.

Free Description

In this free section, present in **at most 10 lines**, all the pieces of information you will find useful to let the examiner better understand the software application you have developed and the nature of your contribution. This section can contain facts such as the programming language(s) used, the code size, the complexity and the nature of your contribution, etc. It can also describe the relationships between this software and your other research activities.