## PRESOFT projet: Research Software Management Plan template (PRESOFT project)

### 1. Metadata

**Software name** - *If you need to choose a name, avoid the name of a brand and other software names*

**Short description of the software** - *A short sentence describing your software*

**Software web page or website**

**Link to source code or package**

**Contact***(email adress)*

**Research unit in charge of the software**

**Main developers and their affiliations**

**Software version**

*Exemple de réponse*:

version 1

**Date of the software version**

**Licence**

**Scientific discipline** - *For example according to the ERC* <https://erc.europa.eu/sites/default/files/document/file/ERC_Panel_structure_2018.pdf> *or to the EGI scientific classification:* <https://wiki.egi.eu/wiki/Scientific_Disciplines>

**Main functionalities** - *In the form of keywords*

**Main technical characteristics** - *In the form of keywords*

**Other keywords**

### 2.1 Software context: History

**Preparatory material** - *Identify and date the preparatory material*

**Specifications (if any), conception model (UML or other), use cases...** - *References and dates for the specifications, conception model...*

**Previous software versions** - *Identify and date the previous versions*

**Components included in the software and external dependencies** - *Identify and describe the different components that are part of the software: name, version, date, authors, website, licence...*

**New components to be included in the new version of the software** - *Identify and describe the different components*

**Roadmap** (*Link***)**

**Are there other software developments with similar functionalities? Which are the differences**?

**Publications, data and other associated productions** - *For example: the team publications to explain the software design or to show the obtained scientific results by using the software*

**Up to this date (to be given), estimation of the software’s cost** - *Number of person/months for example*

### 2.2 Software context: Project(s) related to the software

**Project(s) related to the software**

### 2.3 Software context: Legal issues and distribution policy

**Intellectual property** - *Identify authors, rightholders.*

**Rightholders or copyright statement**

**Distribution policy** - *Constraints linked to the project (s), the partners and their organisms*

**Licence(s)** - *Beware of possible heritage and licence compatibility issues. Mention the licences of the documentation, of the web site...*

**If the code is to be open, when will it be open?** - *To be validated with the possible partners and according to the constraints linked to the funding*

**Management of the intellectual property of external contributions** - *Rights' transfer agreement to be planned*

**Non disclosure or privacy clauses and sensitive data processing (if needed)**

### 3.1 Software features: Scientific goals

**Objectives, expected results** - *Describe in a synthetic way the scientific goals and the expected results linked to the software*

### 3.2 Software features: Usage and distribution objectives

**Planned or considered lifespan**

**Planned usage** - *What for (publications, teaching, production level usage, industry level usage)?*

**Target public** - *For example: researchers, team, restricted distribution, collaboration, wide distribution...*

**Planned user support** - *Type of support, tools, resources, quality of service for the users. For example, user support, ticket system, a person in “best effort”...*

**Distribution goals** - *The software is “for internal use only”, the software will be published via an article, the software will be distributed widely...*

**Collaboration community wished** - *If yes, which one?*

* Don't know
* Yes
* No

*Exemple de réponse*:

For example in the target scientific community.

**Adequacy of the resources (development, maintenance...) to the distribution goals** - *Are the available resources suitable? (human, financial and material resources)*

**Risk analysis** - *A risk analysis may be useful before launching an expensive development or an unwise distribution*

**Software preservation** - *What is the objective for the preservation and what is the solution used? - Please distinguish short term backup and long term archiving*

### 3.3 Software features: Technical features

**Used technologies**

**Dependencies** - *OS, SDK, libraries, browser, external APIs...*

**Already existing components reuse** - *Technical constraints*

**Documentation** - *Give the documentation's url*

**Used norms and standards** - *Example: ISO norm of the development language*

### 4. Team organisation

**Governance** - *For example the organisation officially in charge of the software, a consortium...*

**Consortium agreement including governance, development and future of the software** - *In the case of a shared development between several organisms - If the software is developed in the framework of a project, the consortium agreement must take it into account.*

**Team** - *List the members of the team. Indicate for each person its status (employee /institution, internship, student, retired…) and their participation dates*

**Organisation around the software** - *Responsibilities of the different actors: development, training, support, distribution, translation… - Distinguish the different roles: leader, main developers, minor contributors, scientific contributors (no code writing), documentation writers…*

**Costs and funding distribution**

**Type of development** - *Collaborative or not (practical organisation of the collaboration). Note that the 5th section details the development organisation*

**Actions to be planned in case of a person's leave**

### 5. Development organisation

**Development team** - *On one or several sites, depending on one or several institutions...*

**Development plan** - *Roadmap including the new versions, functionalities planned and dates.*

**Development methods, used standards, tools and infrastructures (code repository)** - *Example: tools for version management and collaborative development*

**Actor's responsibilities in the development**

**Quality procedures** - *For example, actions taken to foster the software maintainability, best practices applied, verification tests...*

**Security (taken into account in the development)**

**Version delivery, bugs, tests and validation management** - *Are there testing or other validation procedures? With which follow-up? Are they to be given to final users? How do you manage bugs?*

**Documentation production management (internal and for users, installation and requirements, use examples)** - *Explain how the documentation is produced and updated for each version (responsibilities, organisation…)*

**Describe main planned evolutions** - *For example: integration in other projects, software translations...*

**If external participations are expected and possible, which are the rules (validation of the contributions, contribution integration in the major versions, participation integration)?** - *It is advisable to define accurately the rules before any external participation.*

### 6. Distribution organisation

**Reference repository** - *For example: the link to the software version on SourceSup, Zenodo or Gitlab IN2P3*

**Persistent identifier** - *Indicate for example the DOI of your software*

**Citation form** - *You can suggest to cite the publication that describes your software or the one that seems to be the most important.*

*Otherwise you may propose for example: “author(s), software name, short description, version, date, url”*

**Links to articles or other research outputs external to the team and that use the software** - *Important: to show that the software is used outside the development team or the original laboratories.*

**Referencing (announces, websites of the scientific community...)**

**Communications** - *Conferences, posters, flyers...*

**Publications in a software journal**

**User support (such as offered to the users)**

**Usage indicators** - *Number of downloads, number of exchanges with users...*

### 7. SMP management

**Person in charge of this SMP**

**Is the SMP required by a project funding, an agreement, contract or other?**

**Organisation to write and update the SMP and monitor actions and goals** - *Is there a collaborative place for this SMP? Is it a text document? What is the update frequency or it is updated continuously? What type of events triggers an update? Who are the actors?*

*The current document has been created with the DMP OPIDoR service. Please don't forget to keep the successive versions in your local workspace.*

**Distribution of this SMP**

* Confidential
* Don't know
* Public
* Restricted

**Links with the current project’s Data Management Plan (if any)** - *If yes, is there a reference model or important points to develop? Is this SMP a part of the project’s DMP? -  In certain calls, the DMP template includes a section for software, but a DMP is focused on data, not on software and it is not designed for software management.*