

# PLAN-E Workshop Software Sustainability Dublin, May 9 2016

---

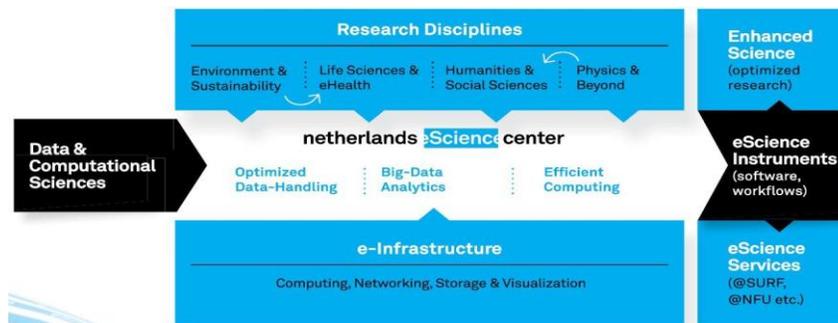
## Introductions

### Three invited contributions embedded the following discussion:

#### The eScience technology platform (Rob van Nieuwpoort)

About a coherent set of technologies to tackle the grand challenges in eScience. Includes guidelines for proper coding behavior for future sustainability. eSTeP, the Technology Platform, will be open to Dutch national and PLAN-E members for sharing certified codes.

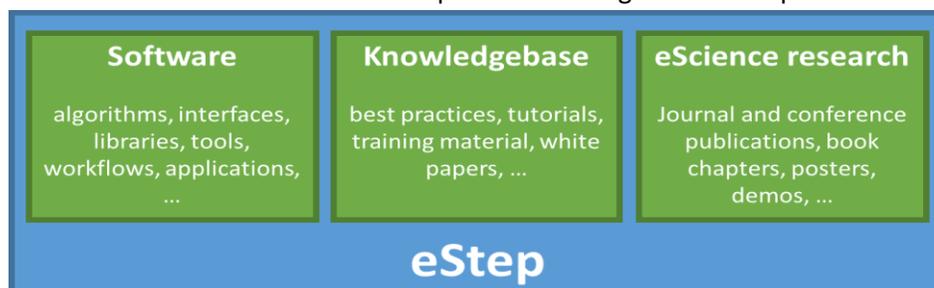
More from the presentation:



This figure sketches the role of the Netherlands eScience Center in between data, ict, infrastructure and science enhancement and accelerating discovery. The relations with the external worlds and the infrastructures are presented.

Cross cutting basic skills @NleSC encompass: Code quality and best practices; Integration of software; Scaling of software; Analytics&Statistics; Visualisation.

Among the portfolio of projects and skills, eSTeP, the eScience Technology Platform is a persistent activity in support of all others. It forms a coherent set of technologies to tackle the grand challenges in science and it is the research vehicle that keeps the knowledge at NLeSC up-to-date.



eSTeP is combined with other goals:

- Prevent fragmentation;
- Promote the exchange of re-use best practices;
- Represent NLeSC expertise and knowledge base;
- Improve the eScience state of the art with a fundamental eScience research line.

Main criteria for access of software to eSTeP are:

- State-of-the-art /Best-of-breed?
- Generic and overarching?
- Match with our expertise areas?
- (Includes externally developed software; it is an open platform!)

The sustainability approach:

- Must be taken into account from the start
- Prevent duplication, fragmentation
- Build something that is worth sustaining!
- Sufficiently generic
- Make things Modular
- High quality
- Enforce software engineering guidelines and best practices
- Educate partners with software carpentry and data carpentry
- Open source / open access, open standards, unless...
- Community coding
- Standardization for software and data formats

NLeSC enforces strict coding styles. See for example: Nicholas C. Zakas: Why Coding Style Matters <http://coding.smashingmagazine.com/2012/10/25/why-coding-style-matters>

NLeSC provides editor configuration (<http://editorconfig.org/>).

General guidelines are applied:

- For Web development: <https://github.com/bendc/frontend-guidelines>
- AngularJS: <https://github.com/johnpapa/angular-styleguide>
- Airbnb JavaScript Style Guide: <https://github.com/airbnb/javascript>
- Python PEP8: <https://www.python.org/dev/peps/pep-0008/>
- Java Code Conventions for the Java™ Programming Language (Oracle)
- Google Style Guides: <https://github.com/google/styleguide>
- Wikipedia: [https://en.wikipedia.org/wiki/Coding\\_conventions](https://en.wikipedia.org/wiki/Coding_conventions)

Quality Improvement tools:

- Article about good development practices: [The Joel Test: 12 Steps to Better Code](#).
- SonarQube: <http://www.sonarqube.org>
- Code climate: <https://codeclimate.com>
- Codacy: <https://www.codacy.com>
- Scrutinizer: <https://scrutinizer-ci.com>
- Landscape: <https://landscape.io>
- Coveralls: <https://coveralls.io>
- See also <https://github.com/>.

## Unit&Integration testing:

- Guide: Writing Testable Code
- 'Unit Testing Best Practices' and other presentations on <http://artofunittesting.com/>.
- Continuous integration testing with Travis-CI and Jenkins-CI
- We require at least 70% code coverage
- Java: junit
- Javascript
- Jasmine, a behavior-driven development framework for testing JavaScript code.
- Karma, Runs tests in web browser with code coverage.
- PhantomJS, headless web browser on CI-servers.
- Python
- Unittest, nose and pytest.
- R
- testthat
- Web development
- To interact with web-browsers use Selenium.
- Sauce Labs hosts a matrix of web-browsers and Operating Systems for testing.
- AngularJS applications can be tested with Protractor.

## Documentation

- Document at multiple levels
- Source code comments
- API documentation
- Installation and usage documentation
- Comments at each level should take into account the different target audiences
- Use Markdown, a readable lightweight markup language that can be converted to many formats.

## Version control:

- Git and GitHub
- A successful and simple Git branching model: GitHub Flow <https://guides.github.com/introduction/flow/>
- Commit messages are formatted and formulated in a readable way
- <http://tbagery.com/2008/04/19/a-note-about-git-commit-messages.html>

## Releases and packaging:

- Tag versions, use github releases
- Semantic versioning
- Keep changelogs
- Packaging is important
- Use packaging that is well known and appropriate for user community: pypi, npm, maven, docker
- Make your code and data citable: get a DOI (Zenodo).

Even more can be found at the PLAN-E web site (full presentation by Rob van Nieuwpoort) and the web site of NLeSC.

## e-Science and Software Sustainability in France (Genevieve ROMIER)

The French Academic Research landscape is scattered<sup>1</sup>: in terms of national budget there are 20 main research institutes, among which CNRS (19%), CEA (14%), INRA (5%), Inserm (5%) and Inria (<2%). There is presently no national software-specific policy in place and the academic software producers are free to organize their software production and diffusion. However, there are a few initiatives regarding

---

<sup>1</sup> The following is based on information provided by the French membership of PLAN-E, The Platform of National eScience Centers in Europe (Dublin, May 9-10 2016)

software sustainability with a national scope: SourceSup, CeCILL, Plume, and some professional networks and entities to link academic work to industry and SME's.

SourceSup<sup>2</sup>: national academic forge "La forge Enseignement supérieur et Recherche", provided by CRU and after that by Renater (NREN) since 2004. This project hosts software development projects for academic research and universities and their collaborations. Currently there are 489 visible hosted projects. SourceSup gathers a set of services all based on open source technologies (subversion, git, Jenkins, Sonar, Nexus...). Authentication is done with the "federation Éducation-Recherche" identity federation (266 identity providers and catch all for external collaborations). It has a complete set of functionalities: link with Renater mailing lists server (Sympa<sup>3</sup>), possibility of project website, building, deploying and automating tools (Jenkins), code quality (Sonar), documentation management (Nuxeo). There are other institutional or thematic forges.

CeCILL licenses<sup>4</sup>. The CeCILL family licenses are developed by CEA, CNRS and Inria since 2005. The arguments: "Today Free Software is important in the scientific community as well as in administrations and in the enterprise. Nevertheless, the use of licenses created in the US, such as the GNU General Public License raises some legal issues. These issues may lead to uncertainties that may prevent some companies and organisations to contribute Free Software. To provide a better legal safety while keeping the spirit of these licenses, three French public research organisations, CEA, CNRS and Inria launched a project to write Free Software licenses conforming to French law. " CeCILL-A is designed to be fully compatible with GNU GPL.

Plume<sup>5</sup>: to Promote economical, Useful and Maintained software For the Higher Education And THE Research communities. Plume is mainly supported by CNRS 2006-2013. The portal provides a software catalogue, mainly Free/Libre Open Source Software (FLOSS) that is used and/or produced in universities and national research organisations, laboratories or departments. It presently involves 2200 members, 950 contributors, 18 themes, 1270 index cards, several keywords levels including institutes and user's classification, and is based on open source technologies (Drupal...). Typical actions: topical workshops and training sessions oriented to software development (tools, dissemination, IPR and licenses...) or targeting community building around software interest, cross-disciplines experience sharing. Due to lack of resources the project was frozen in mid-2013, but the information and the platform are still online.

Plume & research software: Several research units (ICJ, LAAS, LIGM) required research software descriptions in Plume to disseminate their software production. Some results: 350 research software products produced in French laboratories descriptions (1/3 translated in English); actions on software dissemination (documentation, guidelines, about 10 workshops); work on legal issues (French IPR laws and licenses); PLUME-FEATHER: <https://www.projet-plume.org/eng>.

---

<sup>2</sup> <https://sourcesup.renater.fr/>

<sup>3</sup> Sympa is an open source mailing lists server created by CRU and provided as a service by CRU then Renater since 1997 for the research and universities members

<sup>4</sup> <http://www.cecill.info/index.en.html>

<sup>5</sup> <https://www.projet-plume.org>

## Professional networks:

DevLOG<sup>6</sup>: software developers, officially supported by CNRS since 2011. Mailing list: 1000 subscribers. Main actions: JDEV annual conference, topical workshops and training sessions (software development, dissemination, IPR and licenses), community building, cross-disciplines experience sharing

RBDD<sup>7</sup>: databases professional network officially supported by CNRS since 2012. Mailing list: 350 subscribers. Main actions: recommendations, workshops and training sessions (databases and related tools development, dissemination, IPR and licenses, cross-disciplines experience sharing).

Other professional networks are focused on regions, disciplines (scientific computation such as "Groupe Calcul"...) or institutes. They are federated at national level.

Future initiative: IdGC together with other CNRS laboratories and European partners have submitted a proposal about software, awaiting granting.

## Involving the Science Community in Data Stewardship and Software Sustainability - A Conceptual Approach (Patrick Aerts)

A key note given at the Open Science Presidency Conference last April in Amsterdam. It addresses a methodology for involving explicitly scientists in the process of creating protocols for Software Sustainability and Data Stewardship. It is based on a paper published by DANS.

If it comes to Data Stewardship and Software Sustainability there are different stakeholders involved, each with a different view on the matter. A graphical representation of the breakdown into stakeholder categories, their responsibilities and roles and the domain of their actions is displayed below:



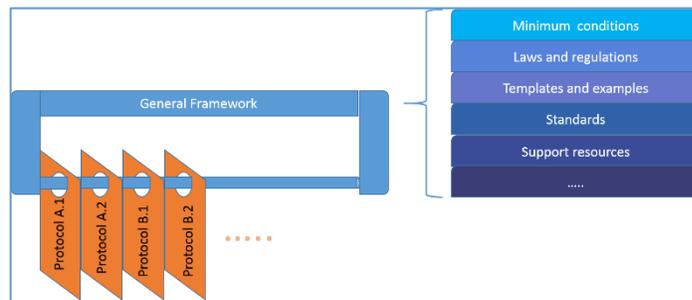
Figure 1 Overview categories of stakeholders and their roles

Details can be found in the presentation on the PLAN-E website and the document A Conceptual approach to Data Stewardship and Software Sustainability can be found at:...

The bottom line is that the matter should be approached along disciplinary lines, if so required per sub discipline and/or per data or software type. But the "user involvement" comes mainly from the action to set up so called protocols for data stewardship and for software sustainability per subdiscipline, by that subdiscipline, with the aid of experts. These protocols then are to be published as papers in the traditional scientific way for later reference by all of the (sub-) discipline. In due course then a series of protocols will come into existence, fulfilling a set of minimum criteria and general legislation by national and supra-national (for example) European governments.

<sup>6</sup> <http://devlog.cnrs.fr/>

<sup>7</sup> <http://rbdd.cnrs.fr/>



For the workshop the participants were split into two groups which both discussed both topics

- Software sustainability issues, technical (minimum conditions, software seal of approval)
- Plans for implementation of policies.

After that the output from both groups was plenary shared.

Software sustainability issues, technical (minimum conditions, software seal of approval)

Issue 1: Discussing the balance between best practice and efficiency

- ☞ It is necessary to find a proper balance between ultimate best practice and time-to-solution. This is sort of a competition: long term gain versus short term gain;
- ☞ Value of Software (in man-months) and Data stewardship cost must be taken into account;
- ☞ Version control and documentation must be required;
- ☞ Tools to create software properly and easy guidelines must be available
- ☞ Need funding engagement from funders for writing software “properly”

Issue 2: Discussing legal issues

- ☞ Legislation prevents using older software, in particular operating systems and the like;
- ☞ Licensing should be discussed when starting a software project;
- ☞ Licensing can be problematic in case of abandonware -> change legislation;
- ☞ Unesco Persist should be supported in their efforts towards global software companies (such as, but not limited to Apple, Google, IBM, Microsoft)
- ☞ Need legally watertight policies if to be enforced without legal challenge
- ☞ Uniformity needed for IP lateral transfer issues (the internet knows no borders). Especially important in strategic economically active areas (physical sciences, bio-sciences, etc.)
- ☞ Legal matters differ in impact largely going from the short term (years) to the long term (decennia)

- ☞ Have an open eye (and more) for cyber security related matters (NIS<sup>8</sup>, GDPR<sup>9</sup>, ENISA<sup>10</sup>)

### Issue 3: Discussing technology

- ☞ Commonality of knowledge of best practice is a good starting point
- ☞ Knowledge of formats
- ☞ Knowledge of Platforms (languages) and how to use them in language specific standards
- ☞ Documentation is essential. Part of the design phase
- ☞ Parallelism/Efficiency measures considered from design phase
- ☞ Licensing should also be part of the design phase considerations
- ☞ Support models, in particular for the after-project phase
- ☞ Availability of expertise/Personnel when considering different models
- ☞ Utilisation of contractors and outsourcing models

### Issue 4: Discussing Openness

- ☞ Open with qualifiers
  - Attribution
  - Accessible
  - Quality and correctness indicators
  - Testing and validation (unit + integration) before opening
  - Definition of sustainability (software - hardware relationship)
  - Decoupling to maximise Return on Investment
  - Metadata - minimal necessary, normally community specific
- ☞ FAIR must be followed through, for data and software;
- ☞ Should be open about the downsides of openness too;
- ☞ How to attribute when re-using? “Nobel Prize problem”;

---

<sup>8</sup> On 17 May 2016, the Council formally adopted new rules to step up the security of network and information systems across the EU. The network and information security (NIS) directive will increase cooperation between member states on the vital issue of cybersecurity. It lays down security obligations for operators of essential services (in critical sectors such as energy, transport, health and finance) and for digital service providers (online marketplaces, search engines and cloud services). Each EU country will also be required to designate one or more national authorities and to establish a strategy for dealing with cyber threats. From:

<http://www.consilium.europa.eu/en/press/press-releases/2016/05/17-wide-cybersecurity-rule-adopted/>

<sup>9</sup> New European cyber laws GDPR and NISD. In December 2015, two new pieces of EU legislation were agreed. General Data Protection Regulation (GDPR) represents a profound reform of data protection law in Europe, shifting the balance of power towards the citizen to whom the personal data belongs, away from organisations that collect, analyse and use such data. Network and Information Security Directive (NISD) can be regarded as a complementary law to GDPR, designed to create a focus on the protection of IT systems in European critical national infrastructure (CNI). Essentially it introduces new breach reporting obligations to whole new swathes of industry, including the energy, transport, banking and healthcare sectors.

Both GDPR and NISD are expected to come into force in spring 2016 but there will be a period of up to two years during which organisations will be allowed to prepare for the new regulations and for the directive to be transposed into country law. From: <http://www.cgi-group.co.uk/systems-integration-services/cyber-security/nisdandgdpr>

<sup>10</sup> The European Union Agency for Network and Information Security (ENISA) is a centre of expertise for cyber security in Europe. The Agency is located in Greece with its seat in Heraklion Crete and an operational office in Athens. (<https://www.enisa.europa.eu/about-enisa>)

- ☞ Resource limitations essentially delay openness
- ☞ Consider a blockchain<sup>11</sup> for software

#### Issue 5: Delimiting the scope of the software sustainability issue

- ☞ Consider limiting to software in direct support of a publication
- ☞ Consider limiting to “scientific software”
- ☞ Consider limiting to user-developed scientific software (so not including big packages, Matlab or Matlab-generated code)
- ☞ Criteria for sustaining (or levels of maintaining) software are urgently needed;
- ☞ “With money, everything can be sustained”

#### Plans for implementation of policies

#### Issue 6: Actions to stimulate proper conduct in coding practices;/How to get adoption in place?

- ☞ Attention to the matter during education (see eSkills discussion);
- ☞ “He who pays the piper calls the tune” applies to funding organisations
- ☞ Giving attribution and proper citation should be(come) the norm. (Naming and shaming of those who don’t...)
- ☞ Those who develop should be (scientifically) rewarded. But how to reward properly?
- ☞ Consider supporting the Software Seal of Approval:
  - In early stage of conceivment;
  - Initiative by DANS and NLeSC, (compare Data Seal of Approval)
  - Idea supported by SSI (UK) in this early stage
  - Idea supported by Knowledge Exchange Group (Jisc, SURF, DFG, DEFF, CSC)
  - Workshop by KE scheduled for October 2016.
- ☞ Have research councils and other policy parties be aware about the changing face of science and scientific methodologies (i.e. eScience)

#### Issue 7: Concrete actions

- ☞ Build on the early experiences by SSI (UK)
- ☞ Consider setting up Software Sustainability Initiatives in every European country
  - And then forming a European Software Sustainability Infrastructure from these national initiatives;
- ☞ Consider requesting the e-IRG to endorse such an SSI construct
- ☞ Consider discussing setting up a scientific European git<sup>12</sup> service
- ☞ Similarly: consider a registry of publicly available software

---

<sup>11</sup> The blockchain consists of *blocks* that hold [timestamped](#) batches of valid [transactions](#). Each block includes the [hash](#) of the prior block, linking the blocks together. The linked blocks form a *chain*, with each additional block reinforcing those before it, thus giving the database type its name. From: [https://en.wikipedia.org/wiki/Blockchain\\_\(database\)](https://en.wikipedia.org/wiki/Blockchain_(database))

<sup>12</sup> Git is a version control system that is widely used for software development and other version control tasks. It is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. From: [https://en.wikipedia.org/wiki/Git\\_\(software\)](https://en.wikipedia.org/wiki/Git_(software))