Open Source Adoption Factors

MEDETEL 2016 in Luxembourg

Etienne Saliez

- Assuming that the potential benefits of Open Source are already known, what should be improved ?

Open Source Adoption Key Issues

- Support services
- Application quality
- Technical Quality Control
- Responsibilities and insurances
- Reparative software maintenance
- Prospective software developments
- Openness, documentation quality

Support Services (1)

- Essential for the sustainability of any system in production.
- Coordination of contacts.
- Installation and initial training.
- Help desk at any time.
- First line of assistance in case of technical emergency. Decentralisation in the region of the users.

Support Services (2)

- Traditional business based on services quality and availability.
- Can be charged for the availability by means of a global yearly package.
- Require qualified people motivated to help users, rather than the most genial developers.

Application quality

- Although many applications are already available in MEDFLOSS.ORG, more and more will be necessary.
- Reuse of Open Source software modules is a priority.
- Try to find generic solutions when possible. Avoid too many local exceptions.
- Inventory of missing points and wishes with their relative priorities.
- Improve the integration between medical applications.

Technical Quality Control

- Peer review.
- Extended test procedures and build-in redundancy checks.
- Formal validation procedure of any new version by experts independent from the developers.
- Protections against intrusions. Reliability of content and confidentiality.
- Analysis of any incident.

Responsibilities

- Clear vision of who is responsible for what, users, software support, software development.
- Conformance to the current “rules of the art”.
- Conscious risk analysis. Keep risks extremely low as far as reasonably possible, taking account of patients care needs and available human resources.
- Since the risk can never be at absolute zero review insurance conditions.

Software reparative maintenance

- Any bug must be fixed as soon as possible.
- Need to maintain contact with a team of several developers having experience with the software.
- However normally not much work, but must be available at any time.

Software prospective maintenance

- Software become soon outdated after a few years.
  In general the user requirements increase with time for more extended software packages.
- Need to maintain a community motivated by development challenges.
  A community sufficiently large in order not to depend on any individual.

Modularity, Integration

- Seek modular solutions in MEDFLOSS.ORG and related repositories.
  Need for more interoperability.
- It should become possible to replace a component without affecting too much the whole system.
- A global approach starting from the patient point of view.

Development Economic Model

- Evaluation of the necessary resources and of the “return” of efforts, in qualitative and quantitative ways.
- Comparisons with alternative solutions, considering the global costs of at least 5 years in a large groups of users having similar needs.
- Thanks to Open Source volunteers, many developments have already been achieved, but developers need incomes in order to spend more time on critical projects.

Seek resources for continued developments

- Governments: development supported by taxpayer money should always be available in the public domain.
- Welfare foundations seeking to maximize the qualitative returns of their efforts.
- Donation from users needing a software extension, but not remaining owners.

Full Openness

- Documentation quality and transparency is a key factor of collaborative software.
  Independent informaticians in an other country must be able to understand how it works.
- Keep all the documentation available on a public repository, like “GitHub”.
- Avoid dependency on any single service provider. In case of need, other IT people must be able to take over.