

Intellectual Property Rights in Software Development: *best practices*

Salvatore Spinello

CERN, 21 October 2010

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



centre de recherche
BORDEAUX - SUD-OUEST

Introduction

The goal is to instil, from the beginning of software development, the good practices to have legal status as healthy as possible:

- 1. During the initial phase of development**
- 2. During the evolution of software**
- 3. When releasing or transferring the software**



1. During the initial phase of development

1.1 *Some principles to share with authors*

- IP generated by an employee through the course of his or her employment legally belongs to the employer
- The definition of ownership between authors can be different from ownership of copyright between organizations (particular agreements)
- Possible contractual complications if exogenous code is integrated in the software (e.g. rewrite the code if the contribution is small)
- It is recommended to be in a situation of centralized rights:
 - ensuring that the institutional contributors concede their right of exploitation
 - ensure that all individual contributors (students, trainee, ...) concede their rights (under full knowledge)
- It is requested to be extremely careful in the consortium agreements
- Consider authors (PhDs, engineers, etc.) financed or co-financed by third parties (Regional Government, Competitiveness Clusters, etc.)
- If doubt, please, meet the Technology Transfer Officer (TTO)



1.2 Coding: Implement best practices from the beginning

- Integration of exogenous code must be done knowingly and must be recorded and tracked
- Check that the external components include appropriate headers
- Never modify the headers of the components that are integrated
- Indicate in the software documentation all contributors
- Make modular architectures:
 - (if necessary) remove third-party code
 - facilitate legal analysis
 - optimize the transfer schemes
- Describe and regularly update the software architecture (help to trace the development)



1.3 Using the Forge

- Management must be very strict about archiving versions and related documentation (different releases)
- The organization of files must allow to easily distinguish endogenous from exogenous code
- Do not use unprofessional email address and avoid individual addresses (use of mailing-list is recommended)
- Well identify also the authors of documents (in order to identify the software authors who have not developed code)



1.4 Headers

- Diffuse largely to developers a note on the headers
- A good header allows a good track of software components: historical and legal
- Use tools to verify that all necessary information is present in the headers (e.g. Fossology)

Main Header: no authors' name (too many, different implication, etc.)

«Software X – Copyright © by ... and ... – Licence ... – 2004

This software is a derivative work based on software Y – Copyright © by ... – ... license»

File's Header: give a contact person (main contributor)

«Software X – Copyright © by ...

Contact : Xsoftware_list@...

Latest modification:

01/01/2010 – adding the function.... »

Function's Header: give the author's name

«Contact : author@...

Latest modification:

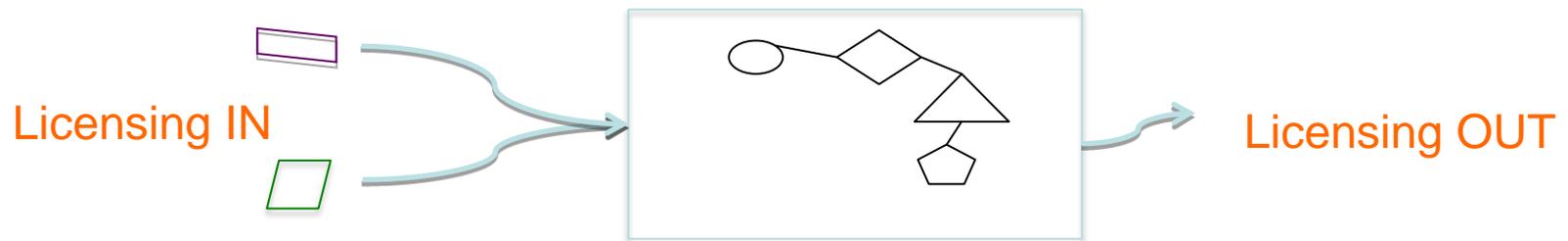
01/01/2010 - optimization of the algorithm.... »



2. During the evolution of software

2.1 Linking

- Know the existing contradictions when integrating exogenous components (licensing IN – OUT)
- How packages are used
- Whether or not source code has been modified
- The method of linking used in development
- Whether and how code is distributed



2.2 Change of License

- Only if there are good raisons
- It is necessary to be constant over time
- Make sure that we have the agreement of all copyright owners
- In all cases, this change is decided by the TTO



2.3 Governance

Open source projects are usually opened to the submission and integration of external contributions. According to the project and the type of governance it may be necessary to:

- Centralize IP rights on these external contributions, by obtaining an assignment of rights by the contributor
- Obtain a license on these external contributions, in order to secure (from a legal point of view) the exploitation of those contributions
- *Preventing Forking and Hijacking* (giving a trusted leader the exclusive right to decide which changes and extensions become part of the official code)



2.4 Submission to APP (French Software Protection Agency)

- Submit regularly the software to APP, in order to take into account the dilution of versions and the cumulated effort
- Protection, proof of precedence (legal and international)
- Submit a software to APP at each major step:
 - good enough maturity
 - major releases
 - background in a contract or collaboration
 - transfer
 - open source or binary code distribution
 - leaving of a key team member
- Agree on (co-) ownership



3. When releasing or transferring the software

3.1 Decision

- A distribution is not trivial
- Do not distribute the software (to keep it as internal) as long as the transfer scheme is not defined and agreed
- It is important to remember that all copyright owners have the exclusive right to choose the mode of distribution (it is necessary to have the agreement of all owners of rights related to the software before its distribution)
- The decision to perform distribution must be deeply thought and subject to the approval of the TTO



3.2 *Legal status of software*

- Establish the legal status of software before any distribution identifying:
 - authors / copyright owners
 - the nature of the software (first, derived, composed)
 - contracts affecting the software
 - other existing enforceable IPR against software
- It is important that the author becomes familiar with the issue of licenses, but it should not be autonomous to the analysis of compatibility. Indeed, the detailed analysis of compatibilities requires the competence of lawyers trained for this



3.3 Choice of license

- The TTO chooses the license, based on exchange with the authors
- It is forbidden to choose an ad hoc license, without TTO derogation
- It is important to be trained in the use of licenses (impact of using exogenous code)
- It is necessary to study all possible cases



3.4 Packaging and redistribution

- Well display the selected distribution license
- Follow the good practices of packaging (compressing format, installation routine, platforms, etc.)
- Installation goes through a phase of license acceptance (check the box "I accept")
- When using composite software, it may be appropriate to distribute everything (If necessary, analyse the licenses' compatibility)
- Pay attention to the icons, software's name (brand), images' sources, etc.



4. What to do in case of doubt?

- Contact the “Experimentation Department” for everything that concerns the development and tools
- Meet the TTO for everything related to transfer / license
- Meet the lawyer for everything related to contracts and legal matters