

# Public Administration and Free Software

## a hypothesis for a **New Business Model**

**Laura Garbati – CSI-Piemonte, Italy**

# Who we are



- A consortium of more than 100 public entities
- **Mission** : management of the information systems of the consortium members
- Use, develop and contribute to free software project



<http://www.doqui.it/>  
GNU GPLv2 license



## **NIVOLA**

- ✓ Under development
- ✓ It will be published under GPL3 license

## **Framework SIRA (GeoServer)**

✓  
<https://github.com/csi-sira>



<https://github.com/csi-sira>



<https://www.smartdatanet.it/yucca.html>  
EUPL 1.1 license



# Public Administration and OS

- Why a new business model ?
- The basic resistance to change
- What has been already done
  - By law
  - By praxis (AGID)
- CSI-Piemonte : a practical view
  - A step by step approach
  - Some practical cases



# Free software and Italian Law: a long long story

Since 2003 the Italian law directs the PA towards  
free software

What went wrong ?



get off the theory by confronting  
the resistance to change

# Resistance to change

my treasure !

compatibility ? But  
it's free ! (as a  
beer...)

will I lose  
control ?



who takes  
responsibility?

But it is a paradox because  
openness *free* the public !

# A wind of change ?



- PA needs not (only) a open licence but an **open mind** : a conversion through habits

Yes, yes but... how?

- Enforcement : not « It would be nice if » but... do it ! (new rt. 69 CAD)
- Show the path : new standardized tools (new AGID Guidelines)
- A modification from inside! (aka how-to for dummies!)

# Free software : just do it !

- New art. 69 Code of Digital Public Administration (D.lgs. 82/2005)

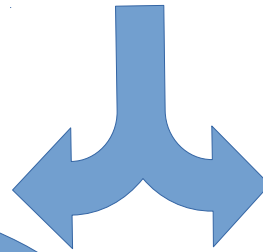
Before...

**The public administrations that are holders of solutions ... have the obligation to make available the relevant source code, complete with the documentation and released in public repository under an open license...**  
(modified by D.Lgs. 179/2016)

The public administrations that are holders of computer programs realized on specific indications of the public client, **must give them in source format**, complete with the available documentation, **in free use to other public administrations** that require them and who intend to adapt them to their needs, except for justified reasons

# AGID Guidelines on acquisition and reuse of software for public administrations

A guide for



evaluate software

reusing software  
in a free environment

- Open consultation since 5th of May
- Latest version 28th of July
- Soon to be published

<https://lg-acquisizione-e-riuso-software-per-la-pa.readthedocs.io/it/latest/>



# AGID Guidelines

before writing code...



- Guarantee ownership (of what is yours) : contractual clauses model
- Choose a licence : list of recommended choices throught licences (ref OSI licences)
- Choose a repository : a indexable“code hosting” platform; which allow open and collaborative software development, without authentication
- Adopting standards : SPDX and reuse.software/practices2.0

# Step by step: before writing code

- deliberate once and for all
  - to adopt and develop free software
  - a (flexible) licensing strategy *and* a repository
- create a FOSS Team (ref. Openchain - [openchainproject.org/](https://openchainproject.org/))
  - ...acting as a change agent (documentation, training..)
  - ...supporting project groups for the assessment of technical and legal impacts
- re-organize: all personnel must be oriented and involved
  - define a Responsibility Assignment Matrix (to develop and contribute)
  - endorse your expert !

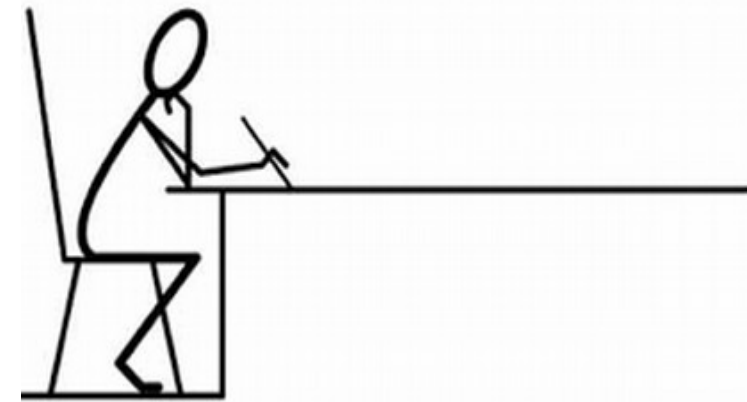


# AGID Guidelines/2

Openness by design



- Include rules in tender
- Develop on the repository
- Think also of the maintenance



the release in free software must be considered *from* the design phase

# Step by step : developing

- Technical compliance : standardized technical documentation
  - Source code style
  - secure coding rules
  - assessment test
  - versioning convention
- Legal compliance : a common (flexible) compatibility matrix
  - to develop from existing project
  - to develop usign third party items
  - To verify software already developed
- Keep a Bill fo Material
- Never use « anonymous » code



# Step by step : a compatibility matrix

un sw rilasciato con licenza	può essere integrato/linkato ad un applicativo inizialmente con licenza...								
	GPLv3			LGPL2 only			LGPL2 or later		
	integrazione	link dinamico	link statico	integrazione	link dinamico	link statico	integrazione	link dinamico	link statico
<b>EUPL 1.1 only</b>	x (ask permission)	verificare con il legale	x (ask permission)	x	verificare con il legale	verificare con il legale	x	verificare con il legale	verificare con il legale
<b>EUPL 1.1 or later</b>	✓ (a)	✓ (a)	✓ (a)	LGPL2 only (****)	✓	✓(****) (se derivata, combination under LGPL 2 only)	LGPL 2 o 3 (****)	verificare con il legale	LGPL2/3 (****) (if derived combined LGPL2 or later)
<b>EUPL 1.2</b>	Combinazione GPLv3	verificare con il legale	Combinazione GPLv3	combinazione LGPL 2 only	✓	✓ (se derivata, combination under LGPL 2 only)	LGPL2 o 3 (****)	verificare con il legale	✓ (se derivata, combined under LGPL2 or later)
<b>GPLv2 only</b>	x	verificare con il legale	x	GPLv2 only (b)	verificare con il legale	GPLv2 only (b)	Combination in under GPLv2 only (**)(b)	verificare con il legale	Combination in under GPLv2 only (**)(b)
<b>GPLv2 or later</b>	✓	✓	✓	Combination in under GPLv2 or later (b)	verificare con il legale	Combination in under GPLv2 or later (b)	Combination in under GPLv2 or later (b)	verificare con il legale	Combination in under GPLv2 or later (b)
<b>GPLv3</b>	✓	✓	✓	combination GPL3 (b)	verificare con il legale	combination GPL3 (b)	combination GPL3 (b)	verificare con il legale	combination GPL3 (b)
<b>LGPLv2 only</b>	GPL3 (b)	✓	verificare con il legale	✓	✓	✓	✓ (*)	✓	✓
<b>LGPLv2 or later</b>	GPL3 (b)	✓	GPL3 (b)	combination LGPL only (***)	✓	verificare con il legale	✓	✓	✓

# AGID Guidelines/2

Toward publication



- **What** : complete source code, database architecture, script, graphic asset, documentation (installation, compilation..)
- **How** : complete, not compressed, with maximized readability
- **Accompanied by** proper documentation: (licence, readme, rules for installation and compilation)

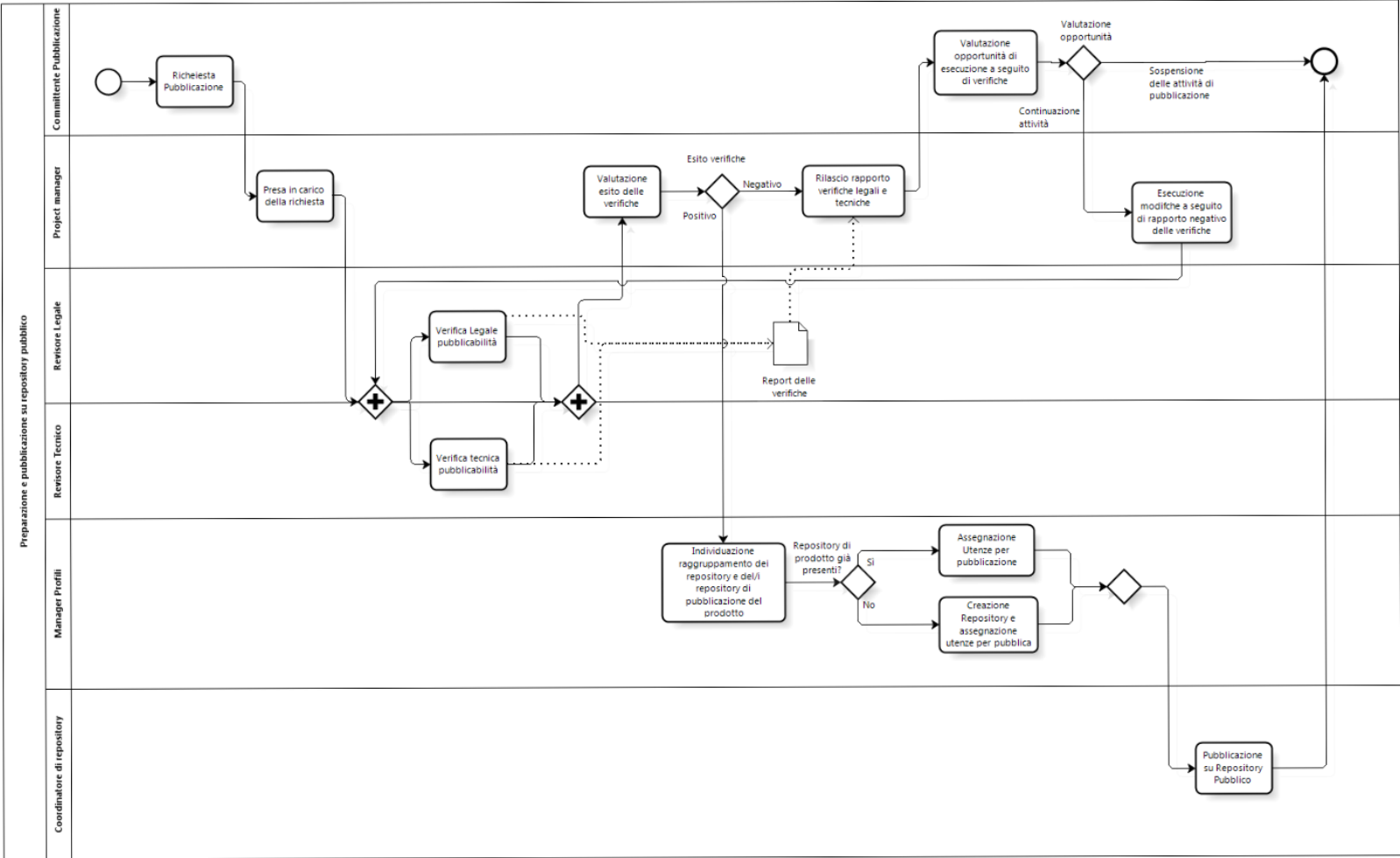
Appendix a : guide to publishing software s open source

# Step by step : before publication

- Standard publication process (from request to github)
- Standard Checklist
  - Do you know what you are publishing ? (BoM)
  - Have passed compatibility check ?
  - Documentation (list of reference) is complete ?
  - Licence and ownership are clear ?
  - The code is clean and the client is aligned ?



# Standard Publication Process





# AGID Guidelines/1

## Reuse and maintenance



- A national collector : Developers Italia  
(<https://developers.italia.it/>)
- For the owner : manage maintenance through community (open continuity)
- For the re-user : do not fork but interact to create a reuse 2.0



# Step by step : a new interaction between PA

- An idea for a two-level (interacting) **community**
  - **General rules and documentation** for every contributor
  - **A collaboration one-sided protocol for Public Administrations** who want to share results but also develop together using the repository and its tools and rules : from pull request to a change in government model

**Community makes the difference !**

# Step by step : models for community

- An authoritative model...
  - ...when it's worth it (and how keep control..)
  - ...when is better to join instead of being joined

based on  
**Collective  
Access**  
(GPL3 license)

CSI choose to contribute to  
the international  
community, which  
guarantees an high technical



a digital platform developed on  
behalf of Piedmont Region  
which aims to to manage  
Piedmontese cultural assets

# A practical exemplum: Nivola

- A new cloud platform *already* developed with open source tools and to be released with a free software license
- How ? Ask to the expert : who wrote the code ?
- Let' different expert interact (technical and legal):
  - List of libraries
  - List of licences
  - Compatibility check
  - Now preparing documentation!



# A practical exemplum: Nivola/2

- Step I : what's in the box ? (scancode)
- Step II : identify the relevant licences
- Step III : check the compatibility
- **Step IV : define the documentation (and the related document) and packages**
- Step V : publication !
- Step VI : hypotize a community



# OS & PA

So...

**...may the (free) force  
be with you !**

**...and thanks !!**

