Software Reuse in Local Public Bodies: Lessons Learned in Tuscany

Vincenzo Ambriola, Giovanni A. Cignoni

HAL Id: hal-01056574
https://inria.hal.science/hal-01056574

Submitted on 20 Aug 2014

Software Reuse in Local Public Bodies: Lessons Learned in Tuscany

Vincenzo Ambriola, Giovanni A. Cignoni

Dipartimento di Informatica, Università di Pisa
Largo B. Pontecorvo, 3 – I-56127 Pisa – Italy
{ambriola, cignoni}@di.unipi.it

Abstract. In the last years, in Italy, *software reuse* has become an e-government hot topic. In this context, reuse is intended as the large scale adoption of software applications developed by local and independent initiatives. Due to the large autonomy of Italian local public administrations, reuse is preferred to centralized development of applications. The paper presents the experience of a three years regional project to manage and enforce reuse in Tuscany. The main result of the project is a *model for reuse* that emphasizes freedom to develop and freedom to adopt. The implementation of the model is based on a *reusable application repository* that lists and certifies software products that are available for reuse.

Keywords: software reuse, regional policies, application repository.

1. Introduction

IT developers are well aware about *reuse* of software components and libraries. They usually share the idea that the practice of reuse is strongly related to the achievement of economical, organizational, and product quality goals. In other words, software reuse is a mature domain with well known and assessed benefits.

In Italy, in the last years, software reuse has become an *e-government* hot topic, adding a new flavour to the meaning of the term. In this context, reuse aims to the large scale adoption of software products which have been locally and independently developed by (or for) public administrations (PA's). These software products are autonomously developed by initiative either of a single PA or of a small group of PA's. In this context, reuse is an interesting issue for private companies too. Usually, local IT companies are involved in implementation, technical support, and services supplied to public bodies. Sometimes, IT companies drive the development of software products with direct investments.

This paper describes the *model for reuse* developed for the Tuscany Region and currently adopted and enforced by a regional law. Section 2 introduces the characteristics and the needs of the Italian local PA's. Section 3 describes the reuse model and its implementation as a *reusable application repository* where reusable products are listed and described along three different dimensions: *availability*, *requirements*, and *quality*. The next sections present the details of the model according to this three dimensional approach: legal availability (Section 4), functional and technological re-

quirements (Section 5), assured quality for reusability (Section 6). Section 7 presents the results of the experimentation of the model, since its introduction in late 2007 to now.

2. The Italian Context and the Tuscany Region Actions

In Italy, between 2000 and 2006 there has been a large financial investment aimed to introduce and exploit information technology in the PA's. In this period, usually called the "first phase of (Italian) e-government", more than 130 projects have been proposed by local PA's and financed by national programmes. In addition to the national initiative, a relevant number of other projects has been started and completed, some financed by regional programmes (36 in Tuscany), others as autonomous investments of local PA's.

A major characteristic of this phase was autonomy (a severe observer may say anarchy). Although national and regional programmes gave some directions, local PA's were completely autonomous in developing their technological solutions. This was a natural consequence of the large administrative autonomy that Italian local PA's have for political, historical, and cultural reasons. As it is easy to argue, in many cases autonomy is a source of difficulties and inefficiencies. However, autonomy is a fact that cannot be changed and that is not worth to discuss here. In the case of e-government, autonomy caused a scattered innovation, a large set of software applications, and, among these, a high level of replication.

For these reasons, after the first phase of e-government, reuse of developed applications has been identified as a way to:

- spread innovation to cover the whole Italian territory,
- enforce selection and integration between applications to decrease replication.

In this perspective, reuse is one of the principal guidelines of the Italian "second phase of e-government", currently on-going. As part of this strategy, reuse is ruled by the Italian law: every PA is required to make its owned software products available to any other PA's that request them.

Today there are many initiatives for promoting reuse in central and local Italian PA's. Details can be found at the National Centre for Information Technology in Public Administration, officially called *DigitPA* [1].

In Tuscany Region software reuse is part of the strategies for supporting innovation for local PA's. In 2005 the role of reuse has been formalized by a regional law that fosters the creation of a *Competence Centre* in charge of providing research and services. In May 2006, Tuscany Region has published a call for starting the *Regional Competence Centre for Reuse* (in Italian, *Centro Regionale di Competenza per il Riuso* or, in short, CRC.R [2]). In December 2006, this project has been assigned to the University of Pisa.

In November 2007 the first proposal of a model for reuse [3] was released and an experimental reusable application repository was set up. In December 2008 the experimentation phase ended and the model proposed by CRC.R was acknowledged and formally recognized in another regional law [4]. In July 2009 the model and its implementation were declared stable and Tuscany Region published a public call to

assign the services for the management of the repository to an industrial supplier. The selection ended in late 2009 and the supplier is currently starting its services. Today, the model for reuse presented in this paper is a running reality.

3. The CRC.R Model for Reuse

The notion of software reuse belongs to Software Engineering since it was founded at Garmisch-Partenkirchen 40 years ago [5]. Today, *reusability* is a software property, defined by the standard IEEE 610 [6] – which is stable since 1990. The reuse process is the subject of another standard, IEEE 1517 [7]. The *reusable assets model* proposed by OMG [8] is the reference framework for describing reusable software objects, both components and complete products. In particular, the OMG model deals with the need of *listing* and *describing* software components in order to retrieving them and to make convenient their selection, use, change, and evolution.

Listing and describing are two foundational aspects of the model for reuse proposed by CRC.R. The model is based on a *shared portfolio of software applications*. Reuse of applications and their natural selection are enforced by publishing their characteristics on the portfolio.

To be effectively reusable, applications have to show specific characteristics. Our model identifies these characteristics along three different dimensions:

- *availability* for reuse, that copes with legal issues and reuse agreements;
- *requirements*, both functional and technological;
- *quality*, intended as assurance of good reusability characteristics.

The proposed model can be viewed as a specialization of the well grounded general OMG model based on listing and describing reusable assets. The shared portfolio of software applications is implemented by a web tool: the *Reusable Application Repository* (RAR) that provides the listing facility.

Listing a product in a publicly browsable repository makes the potential reusers aware about its existence. The next step is to make the interested reusers able to evaluate the product. Description along the three dimensions fulfils this goal and specializes the OMG general model according to the needs of our particular context.

Availability may seem obvious: the presence of a product in the repository should be in itself a condition of its availability. In practice, however, there are several degrees of availability. For instance, availability varies with respect to freedom to modify and redistribute the product, or with respect to the kind of bureaucracy that is needed to actually get the product. In the worst case, reuse may involve the adoption of services that can be provided by a unique industrial supplier.

Product features are the characteristics usually published in a repository for reuse. In this context, r*equirements* have the classical meaning intended in software engineering: the requirements that the product fulfils (and not, as in standard commercial practice, the requirements that must be satisfied to install the product). Requirements must be declared in a way that eases comparison with other solutions, both on functional and technological perspectives. The former is related to the core needs of potential reusers. The latter is related to compatibility with already available platforms

in the installation environment, like hardware, networking, operating systems, database servers, web servers, and so on.

The decision to invest in a reusable product is difficult. A detailed and fair presentation of product requirements is needed to ease the evaluation made by the potential reuser itself, but it is not sufficient. Some advice is needed from a trusted third part. The third dimension of the model, *quality*, fulfils this goal providing incremental certifications of product quality with respect to reusability characteristics.

To make practically viable the evaluation of reusable assets along the three dimensions of the model we implemented three different kinds of tools:

- licenses, to set public references for reuse agreements;
- specification schemas, to define a common grid to declare requirements;
- certification process, to express quality for reuse with respect to given criteria.

Figure 1 shows a graphical representation of the model. In the next sections we discuss in details the three central concepts of the shared portfolio and their implementation in the RAR.
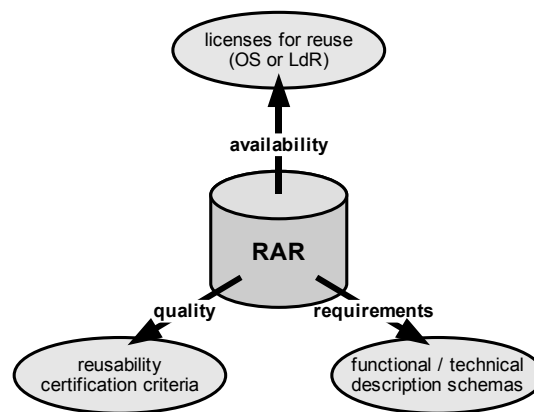

Fig. 1. The three dimensions of product description in the RAR

A last issue that the model must cope with is management of the reuse process among PA's in the Tuscany Region. In few words, reuse is promoted by building, managing, and rationalizing the portfolio of reusable applications of local PA's.

The regional level is a natural choice for implementing and managing the portfolio. From a quantitative perspective, it is large enough to make possible good rates of reuse with perceived benefits. From a functional point of view, software needs of the local PA's often depend on regional issues like laws, policies, and development programmes.

In a nutshell, governing reuse means a "well tuned development" of the portfolio that can be achieved by making systematic diffusion, selection, evolution, and integration of software products. A well tuned portfolio concurs to obtain the following benefits:

- scale economy in the development and management of software products;

- open market for services in support of products, competition among suppliers;
- better integration among services and applications;
- homogeneous and qualitatively selected services for PA's and citizens.

The portfolio must be functionally complete and technologically updated. It must be directly controlled by the local PA's that manage the development of products by means of reuse communities. IT companies participate to this development, either as providers of services or as co-investors.

The reuse community associated to a product is made of PA's and IT companies. While PA's manage the overall life of a product, IT companies are in charge of providing technological services and competences. Being the software products shared among customers, suppliers have to compete among them, and competition may only be based on economical efficiency and technological skills. In this perspective, our model aims at innovation both in the public and in the private sector.

4. Availability

As already stated, in general, there are several degrees of availability. The Italian context makes the issue even more complicated. According to the Italian law [9], a PA that owns a software application must provide it (with no charge) to any other PA that requests it for reuse. Thus availability seems granted. In practice, however, details of availability are ruled by licenses and agreements. And details often matter.

The most common solution implies that each reuse instance is formalized by a contract subscribed by the PA that owns the product and the PA that requests it for reuse. A standardization of this approach was proposed at national level, making available a common template. However, in practice the actual contracts modify, even substantially, the template clauses: the meaning of "reuse" changes drastically among products and even between distinct reuse instances of the same product. Moreover, this "availability on request" is a severe limitation to the practice of reuse: formalizing a contract each time produces a bureaucratic impedance to reuse.

A simpler and more practical solution is to release a product using licenses that explicitly guarantee its availability for reuse. The licences are few, written once and for all, they are public, and cannot be modified. Products can be securely distributed because they are protected by well known licenses. European Union Public License [10] or other Open Source (OS) licenses [11] are natural and for many reasons, auspicious choices to release products for reuse. CRC.R suggests them as a first choice.

However, considering what is required by the Italian law for reuse, it appears that OS licenses are more "open" than requested. This causes some fear and inertia by PA's. In particular, the chance that a product owned and paid by a public body may be used by anyone for profit (as many OS licenses grant) is interpreted by many legal officers as a bad use of public resources. In order to take into account these concerns, we defined an ad hoc *License for Reuse* (LR) [12].

The first goal of the LR is to make software products easy to reuse. The owner, by means of the license, makes available a product in all its components: source code, configuration data, executables, initial contents, documentation for developers, sys-

tem administrators, and users. The first advantage is simplification: reuse no more requires an explicit request, an agreement to be reached, and a contract to be signed.

The second goal of the LR is to grant the maximum level of openness with respect to all misuse concerns. The product is fully available for evaluation by potential reusers and for study by suppliers, thus guaranteeing a fair competition between IT companies interested to provide assistance and development services. LR allows all subjects (public and private) to modify a product and to realize derivative products.

The LR protects the product against actions which are perceived as inappropriate for products developed with public funds. To avoid the product from being improperly marked in terms of ownership, LR states that all contributions to realize and evolve the product must be recognized, with a clear evidence of the period of their execution and their specific nature. In order to avoid uses of a product that could be out of the scope defined by Italian law, LR does not allow private companies to use it for internal applicative purposes or for providing commercial services in ASP mode.

To maintain availability for reuse, LR states that a derived product can be distributed if and only if it is released with the same license. Moreover, in an attempt to preserve quality, LR states that derived products can be distributed only if they are not been *degraded* in reusability. In principle, it may be hard to prove degradation of software quality. In the case of reuse, however, many cases of degradation are easily identifiable: introduction of dependencies from proprietary components, limitation of functionality, limitation of supported platforms, documentation out of sync.

Last but not least, the LR is not a translation of another license stemmed from a different juridical context (like many OS licenses), since it has been written with an explicit reference to the Italian law. This aspect, that at a first glance can be considered marginal, is actually of great relevance for Italian PA's.

Although LR shares many characteristics with other OS licences, it is worth to say that it is not an OS license: licensed products cannot be used by private companies and the non-degradation clause may be interpreted as a limitation to the freedom to modify a product. However, we believe that, even as a "less open" version of an OS license, LR can be effective to spread fundamental OS principles: availability of code and documentation, freedom to change, co-operative development, competition based on technological competence, product life cycle management in charge of the whole stakeholders community (users and developers).

In conclusion, availability of products must be defined by shared and efficient means. OS licenses are a good choice. Where PA's have concerns, the LR fills the gap. In any case, OS licenses, LR, common templates for reuse agreements, custom agreements, are different levels of availability that the RAR made manifest. Besides being a practical solution, the LR also meets the goal to mark a reference for the highest level of availability for reuse before OS licenses.

5. Requirements

The RAR is the tool that makes visible the characteristics of products that belong to the application portfolio. By browsing the RAR, local PA's can verify the convenience of adopting an existing software product before deciding to invest in the development or the acquisition of another one with the same functionality.

The description of the product characteristics must satisfy two needs: searching the products that fit the needs of a PA and selecting the best one in terms of convenience of reuse. The RAR dimension about requirements copes with the need fulfilment issue. Product requirements includes:

- *functional description*; the product must be clearly identified in terms of offered functionalities;
- *documentation*; product documentation must be available including manuals, software architecture, protocols, programming interfaces, test results, and so on;
- *platform dependence*; all dependencies relative to hardware, system and base software, libraries, web services, and the like must be explicitly stated.

To maximize the chance to compare products, requirements are given according to predefined glossaries and ontologies, which cover several level of functionalities, types of documentation and platform components types and versions. Only at the very last level of details, and just for additional documentation purposes, it is possible to fill free text fields or to add links to external documents.

Glossaries and ontologies are managed by CRC.R: they can be dynamically updated to satisfy the needs of the products listed in the RAR. The concept of shared portfolio means that the procedure to describe products requirements is shared too. This solution leads to a common description framework that enhances a fair and objective comparison among products. Managing the requirement description schema is a continuous and demanding task since currently it counts about 300 base functionalities organized in 30 classes. This effort is essential if we want to maintain a fair and shared base for describing and comparing reusable products.

When a public body decide to enrol a product in the RAR we require that a *product manager* be in charge of its evolution. The product manager has access to the RAR and is – formally – committed to a defined role, which includes:

- to fill and keep updated in the RAR all the fields of the product description;
- to give full support to reusers, even potential ones, for the evaluation of the product, providing access to further documentation and demo's.

Since the role of the product manager is demanding, it can be played by several professionals sharing or dividing their competencies. In our model, the role has been defined in order to foster the reuse of a product. Actually, a product that is not supported by human resources adequate to fill its description and to keep it up-to-date, is not reusable. The behaviour of a product manager is a good measure of the support given to a product and, thus, it is a practical indicator of its reusability.

Forcing the commitment of the product manager is a way to implement the notion of *product community*. The PA that originally developed and offered a product for reuse is naturally inclined to lead the community and to provide real resources to promote the reuse of *its* product. We believe that, for PA's, this strategy is politically rewarding. From another perspective, IT companies can invest in the support of a product to gain visibility and show competencies that can be useful for the bidding procedures that assign services and development activities.

6. Quality

The CRC.R identified a process of governing reuse that makes systematic availability, diffusion, selection, evolution, and integration of software products belonging to the shared application portfolio. LR and OS licenses satisfy the availability goal, product requirements description satisfies the diffusion goal. To fully cope with the selection issue the RAR provides a flexible *certification schema*.

Convenience of reuse of a given product is strictly related to assessment and visibility of its quality. Pragmatically, a product is economically convenient if, in addition to satisfying the functional needs of a PA, it is:

- technologically at the state of the art,
- fully documented,
- independent from specific infrastructures (or providers),
- capable to attract a user community that can share experiences and sustain its development.

The RAR description schema already eases autonomous evaluation of these characteristics. In addition, the RAR offers a certification process that implements fair product quality evaluation against common and – again – shared *criteria*.

To guarantee fair, efficient, and timely evaluation of the criteria for all the products enrolled in the RAR, the certification process is completely automatic. Criteria are coded in a set of predicates defined on the fields of the product descriptions. Predicates are periodically evaluated as batch processes (in practice every night). Automation is possible because descriptions of the products are given with respect to predefined glossaries and ontologies. Of course, the validity of a certificate relies on the accuracy of the data in the product description. Here it is worth to remember that description data are formal declarations expressed by a product manager which is committed to a well defined role.

Certification criteria can be tuned according to different perceptions of quality. For software products in general and, in particular, in the perspective of reuse, quality assessment changes in time and may depend on many factors:

- internal needs of PA's;
- services that have to be implemented, by law or by choice;
- availability of new technologies;
- changes in the hardware and software platforms;
- policies for the development of the information society.

The predicate mechanism gives a high degree of freedom in the definition of the criteria. Multiple sets of criteria may be active at the same time. Each set corresponds to a *certificate*, i.e. a symbol that identifies a quality or a group of related qualities of product that are important to highlight for the sake of reuse. For instance, it is possible to define criteria that:

- match the presence of a minimal set of functionalities for products that declare to belong to a given category;

- check the availability of important piece of documentation like licenses and manuals for users and technicians;
- count the number of declared reusing PA's to evaluate the product against a threshold of proven reusability.

The certification mechanism was designed by keeping in mind the flexibility needed to implement what was described as a "well tuned development of the shared portfolio". The initial set of products candidate to be enrolled in the RAR were quite far from having reusability qualities. On the other hand, being severe was not a good choice to start the reuse process. The adopted solution, that strongly relies on the certification mechanism, is to initially build certificates that can highlight the few qualities that somewhere are present. The severity level of certificates will increase in the future and will be used jointly with funding programmes promoted by Tuscany Region. For instance, a product may access a programme if it already has a given certificate or if the funded project aims to improve the product to a higher certificate. In other words, certification is a tool to foster competition and selection among the products of the RAR.
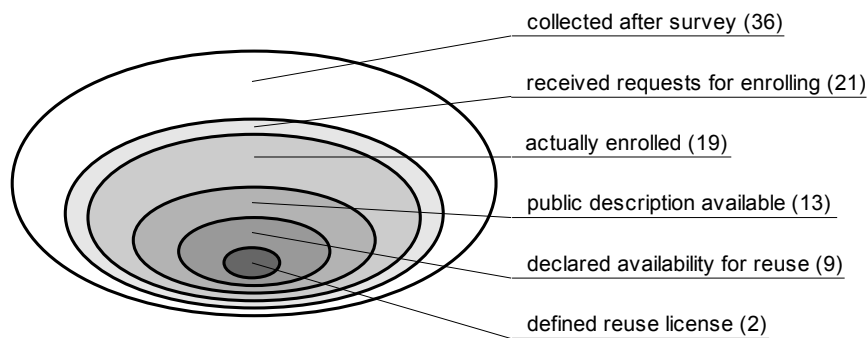
collected after survey (36)

received requests for enrolling (21)

actually enrolled (19)

public description available (13)

declared availability for reuse (9)

defined reuse license (2)

Fig. 2. The actual state of products today in the RAR

7. Results and Lessons Learned

In December 2006, when the CRC.R project started, a survey was made to collect a list of potentially reusable products. The survey started considering the set of projects funded by the Tuscany Region during the so called first phase of e-government, but it was – and still is – open to other contributes. In about three years of activity a total of 36 products were investigated. The results, in terms of products listed in the RAR today, are graphically summarized in figure 2.

To be enrolled in the RAR a product must submit a formal request. All 36 products which emerged from the survey were heartily invited to apply. However, only 21 actually showed up (and 2 of them did not finalize their application).

1st lesson: many public bodies are not ready to support reuse of their products.

Once enrolled, the product manager has to describe the product. The description may require some time and a bit of interaction with CRC.R to learn glossaries and ontologies and, maybe, trade modification and extensions of the shared sets. During the process, to avoid publication of partial data, the product is shown in the RAR in a "closed" state. Of the 19 enrolled products only 13 terminated the process and were opened to public.

2nd lesson: moving along the requirement dimension is difficult.

Looking at the 13 products with open descriptions, the current state of affairs is not encouraging yet. Functional descriptions are in general complete, but documentation and platform dependencies are often incomplete and sometimes missing. It is also a source of concerns the fact that, despite explicit evidence of incompleteness, the managers nevertheless decided to open the product to the public.

Moreover, we found some critical issues related to the declaration of availability for reuse. Of the 13 open products, 9 formally declared their availability, but under particular conditions which, unfortunately, were not specified. Only 2 products had a clear specification of terms and conditions. In many cases, problems about availability of a product were due to contracts with suppliers or project partners that were signed years ago, without taking into account future reuse.

3rd lesson: moving along the availability dimension is even more difficult.

The certification criteria actually in place are very forgiving: the mechanism is not yet used at its full power, and in practice the exploiting of the certification features of the RAR was postponed. From a practical point, simply by asking requirements and availability we triggered a severe selection.

Selection is a good thing, especially when it prevents future investments in badly supported products. However, many PA's were disappointed by this cruel reality and there are concerns about the opportunity to release official evaluations, especially when evaluations are mostly negative.

4th lesson: moving along the quality dimension is, at the moment, almost impossible.

Just by looking at these results one is inclined to conclude that reuse in Tuscany is an *impossible mission*. There are, however, a number of encouraging signals.

First of all, albeit the severe and demanding approach, in the first years of experimentation, our model attracted some new products. Considering the 13 opened products, 4 are new with respect to the initial set and are the outcome of independent projects. We believe that the candidate area for reuse is larger than expected and that the portion outside the regional programmes can express a significant number of products (about one third of the "good" ones).

It is proved that difficulties about availability of the product for reuse can be solved. One of the two products that clearly state terms and conditions for reuse has reached this result by exploiting the consultancy services of the CRC.R. The case was quite complex: the product was the result of several supplies for a PA, the ownership of the product moved from a private company to another; all parties have to agree

about a license. The mediation of the CRC.R was needed to set a number of meetings and to prepare few formal agreements. At the end, a little effort and the general willingness to make the product available for reuse achieved the improvement goal.

From a more technical point of view, functional and technical descriptions of the product were demonstrated feasible too. Actually, products reached full descriptions whenever product managers (and the PA's they represent) were truly committed to invest a reasonable effort to retrieve their documentation and to interact with the suppliers to gather the needed information.

8. Conclusions

In Italy there are examples of different approaches to the promotion and management of reuse. For instance, several other Regions decided to adopt centralized solutions by setting up public owned software houses in charge to develop "standard" application portfolios. Examples in this direction are, for instance, from Piedmont [13] and Latium [14].

However, in Italy local PA's have large autonomy, in part granted by law and in part due to a deep rooted cultural attitude. The strategy adopted by Tuscany Region conforms to this situation: in the model proposed by CRC.R the Region acts as a "soft" coordinator of the application portfolio of its local PA's. The model developed and implemented by CRC.R was successful. With respect to the goal of enforcing selection among products, the three dimensions approach highlighted defects in the products that severely reduced their number. In few, while remarkable, cases it was also demonstrated that targeted product improvement, for instance along just one dimension, is feasible at reasonable costs.

Being the content and the evolution of our shared portfolio driven by natural selection more than by centralized decisions, we can say that the model fully respect *freedom to develop*: new (and hopefully improved) products will arise when the RAR will show the need for them. Moreover, to grant wide freedom to develop and distribute derived products, preference is given both to our LR or to any OS licence.

To respect the *freedom to reuse* by local PA's, the description and the certification schema allows a fair comparison among products, leading to responsible and well-aware choices for reusers. The preferred licenses concur to the capability to evaluate, study, and test products without explicit requests and before signing any contracts.

The model proposed by CRC.R has some costs: on the part of PA's and IT companies, to promote and support reuse of their products; on the part of the management of the RAR, since it requires some effort and high skills.

Costs have their part on the severe selection the products were forced to. However, selection was expected. We consider selection necessary and, by far, preferable to a huge catalogue of ghost products with confused descriptions and unreliable support. Costs are also a clear indication that investments to build reusable products need to be sponsored by regional programmes which have to recognize and award the products able to sustain comparison and selection.

References

1. DigitPA, National Centre for Information Technology in Public Administration, http://www.cnipa.gov.it/
2. CRC.R, Regional Competence Centre for Reuse, http://www.crcr.toscana.it/
3. Ambriola V., Cignoni G.A.: A Regional Experiment to Govern Reuse in Local Public Bodies", in proceedings of the 2nd International Conference on Methodologies, Technologies and Tools enabling e-Government, MeTTeG08 (2008)
4. Tuscany Regional Law 5913/2008: "Istruzioni per le modalità del riuso di applicazioni e prodotti di amministrazione digitale nell'ambito della Rete Telematica Regionale Toscana e per la gestione del Catalogo Regionale per il Riuso (in brief, Instructions for reuse of software products and for management of the Regional Catalog of Reusable Products)" (2008)
5. Naur P., Randell B. (eds.): Software Engineering, in proceedings of NATO Science Committee Conference (1968)
6. IEEE, Std. 610.12: Standard Glossary of Software Engineering Terminology, Institute of Electrical and Electronics Engineers (1990)
7. IEEE, Std. 1517: Standard for Information Technology – Software Life Cycle Processes – Reuse Process, Institute of Electrical and Electronics Engineers (2004)
8. Reusable Asset Specification v. 2.2, Object Management Group (2004)
9. Italian Law 07/03/2005 n. 82: Codice dell'Amministrazione Digitale (Codex of digital administration) (2005)
10. European Commission: European Union Public License, IDABC, EUPL v. 1.1 (2009)
11. Open Source Initiative: List of Open source Licenses.", http://www.opensource.org/
12. Cignoni G.A, Ambriola V., Flick C.: Licenza di Riuso v. 1.00 (Reuse License), CRC.R, Dipartimento di Informatica, Università di Pisa (2008)
13. CSI Piemonte, http://www.csipiemonte.it/en
14. LAit, http://www.laitspa.it/