SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2007; **12**: 191–198
Published online 17 January 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/spip.315

SLA: A Legal Assurance Process Model for Software Engineering Management

Research Section

Ricardo J. Rejas-Muslera,[1]*† Juan J. Cuadrado-Gallego,[2] Miguel-Angel Sicilia[2] and Daniel Rodríguez[3]
[1] *Universidad Francisco de Vitoria, Madrid, Spain*
[2] *Universidad de Alcalá, Madrid, Spain*
[3] *University of Reading, Reading, UK*

The legal assurance activities and measures are a key element for the viability of information systems projects because nowadays there can arise legal risks in some cases, which can be a serious threat for project commercial and financial success. In spite of this, there does not exist in the main evaluation and improvement processes models a process of legal assurance that systematizes and orders the activities and measures precisely by to manage such legal risks. On the other hand, the professional practice does not generally incorporate standardized processes in order to discipline the legal assurance activities and measures. This circumstance can generate the appearance of deficits in the project's legal security. This work proposes to consider the legal assurance activities and measures as a process to implement more in the evaluation and improvement processes models, with the objective to provide a suitable instrument for the management of inherent legal risks to any information systems project. This concept of the legal assurance activities and measures as a process allows the exceeding of the present reactivity characteristic of the effective professional practice and elevates it to proactive management, suitable for avoiding the legal risks that can threaten the project. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS: assurance process model; engineering management; legal risk; software process assessment and improvement models; measurement of the legal protection; risk management

1. INTRODUCTION

The ever increasing importance of software systems in all economic and social sectors implies an increase in the importance of legal aspects associated with such systems. Legal aspects are not only related to the end product but they are also related to every activity in the software development lifecycle. Therefore, legal aspects should be considered as a new kind of project risk that without adequate management can increase the possibility of failure of the project.

Nevertheless, the most important software process assessment and improvement models, such as the Capability Maturity Model Integration (CMMI) CMMI or ISO 15 504 SPICE 15 504, do not properly include processes for legal audits. In the CMMI model, it is possible to find scattered mentions of contractual or legal aspects in the requirements section but the CMMI does not specify activities to be carried out for legal risks management in each

* Correspondence to: Ricardo J. Rejas-Muslera, Universidad Francisco de Vitoria, Madrid, Spain
†E-mail: r.rejas.prof@ufv.es

phase of the software development lifecycle. This implies that activities performed in this area depend on the perception of the risk by managers. Generally, such activities do not follow any temporal pattern to systematically perform them and the most common activity consists only of performing a *Due Diligence* or legal audit before marketing the product. This lack of standardized process means that legal risks are handled reactively instead of proactively, which implies that the problem has already happened and little or nothing can be done about it.

This article presents a Software Legal Audit Process (SLA), which defines legal audit activities that must be performed as part of software process assessment and improvement models ICSE'2000 with the objective of minimizing legal risk for software projects. The aim is to provide industry with a framework to manage legal risks inherent in all software projects efficiently. Such a framework allows us to move from a reactive risk strategy to a proactive one.

The organization of the article is as follows. Section 2 identifies the importance of a legal assurance process model and Section 3 describes the objective of such a process. Section 4 provides a framework for legal audits followed by a measurement model in Section 5. Finally, Section 6 concludes the article and future work is outlined.

2. LEGAL ASSURANCE PROCESS MODEL

To properly manage all legal implications of a software project throughout its development lifecycle, we need first to define a legal audit software process with all needed activities. On one hand, such a process must consider legal risks that may affect the project, and on the other, it must specify what actions can be adopted to avoid or minimize such risks and when.

We use the CMMI model as a starting point to describe the software legal assurance process. The CMMI model defines process as the *action of defining a process* CMMI, and specifies that the process description is

> A documented expression of a set of activities performed to achieve a given purpose that provides an operational definition of the major components of a process. The documentation specifies, in a complete, precise, and verifiable manner, the requirements, design, behavior, or other characteristics of a process. It also may include procedures for determining whether these provisions have been satisfied. Process descriptions may be found at the activity, project, or organizational level.

Given the previous definition, the legal assurance process consists of the description of its process:

1. Purpose definition. This process is defined in Section 3 and it has 2 main objectives:
   (a) Optimization of business opportunities.
   (b) Management of the legal risk that can jeopardize the project.
2. Document a set of activities and specify the requirements, design, and behavior and other characteristics of a process in comprehensive, precise and verifiable format. Section 4 describes what activities need to be performed and when they need to be applied.
3. It is also possible to include a quantitative process to verify the extent of the activities set in place. Section 5 presents such an estimation model.

3. DEFINING THE LEGAL ASSURANCE PROCESS OBJECTIVES

Taking into account that the final aim of any software process assessment and improvement model is to increase process performance and the quality of the products, the definition of a legal assurance process should allow us (i) to optimize business opportunities and (ii) reduce project risks.

3.1. Optimization of Business Opportunities

In current industrial environments, a major differential factor between companies is their ability to create and commercialize knowledge (Kamil 2003). This is a strategic ability especially in the software market owing to the intangible nature of software products and intellectual properties. As a result, a proper protection of these actives is a key element in the management of software organizations.

Taking into account a strategy to manage intellectual property will generate and optimize business opportunities in the following areas:

DOI: 10.1002/spip

- Product commercialization in internal and external markets. An nonexistent or inadequate protection can reduce or even eliminate the commercial life of a software product; a successful commercial product will generate clones that will be commercialized at a much lower cost, avoiding development costs.
- Project funding. Adequate intellectual property protection will provide a powerful financial instrument that can be used to:
  - guarantee credit applications;
  - attract venture capital; or even
  - apply for government benefits and grants for Research and Development (R&D).

3.2. Risk Management

In addition to the possibility of maximizing business opportunities, assurance processes aim to reduce risks or potential threats derived from the failure to comply with the law or inadequate adaptation to legal regulations. Such issues, in turn, can generate legal claims from third parties, economic sanctions from governments or local authorities, and even penal actions that will obviously affect the successful outcome of a project.

The problem of organizing legal assurance activities within the software development lifecycle is not a trivial process because each assurance activity must be applied at the right time, i.e. the efficiency of legal assurance activities depends on applying the right action at the right time. Therefore, it is not enough to perform legal audits or *due diligence* once the project has been completed. By performing ordinary legal audits before launching a product, it is possible to find potential threats that force modifications of certain aspects of the project; usually such unplanned modifications are very expensive or cannot be performed. Neither is sporadic actions by project managers to properly manage legal risks enough. The transformation of products into profitable assets minimizing legal risks demands a process that: (i) determines available legal assurance activities:

- determines available legal assurance activities
- realizes a descriptive analysis of previously defined activities
- incorporates such activities into the software development lifecycle of the software product

In addition to the identification and incorporation of legal assurance activities at the right time, the management of these activities will be greatly improved if supported by a process that includes estimation techniques (from a quantitative point of view) about the protection level in a specific project.

4. LEGAL ASSURANCE ACTIVITIES AND MEASURES: DESCRIPTIVE ANALYSIS AND CONTEXT WITHIN THE SOFTWARE DEVELOPMENT LIFECYCLE

With the objective of creating a process for the management of legal assurance activities in a software project, it is necessary to describe the legal assurance activities available but also to describe when to apply them within the software development lifecycle. Legal assurance activities depend on different factors that can be classified into the following:

- Intrinsic or inherent to the project. For example, accounting software will need different measures than a website used to commercialize products. First, we need to deal with intellectual property rights and data protection measures. Second, we also need to consider other aspects such as contract conditions and the publicity of products.
- Extrinsic or factors external to the project, for example, legal regulations or the market structure in which the product operates.

Therefore, the first step consists of determining what activities need to be applied for each project. Then, an analysis in terms of scope and context within the software development lifecycle is to be carried out.

In the following subsections, we present a series of legal activities following the software development lifecycle to deal with the protection of the software by means of copyrights that can be applied to any software project OMPI 1967 DOCE 1991. These activities form a generic process model and adapted to the European Union (EU) regulations that would need to be adapted to extrinsic and intrinsic factors of the project. In the case of a software project developed in the USA (http://www.uspto.gov/) or Japan, for example, it would be necessary to consider software patents as a protection system.

DOI: 10.1002/spip

4.1. Planning Activities

In relation to project stakeholders, we need to distinguish between internal and external personnel, i.e. personnel working directly under a contract with the developer corporation (internal) or personnel working for the corporation caring out the development but with a contract on another corporation.

4.1.1. External Personnel

The development of the software will be generally agreed upon under the terms of a contract, the parties being the parts free to define their respective rights and obligations in the contract. Therefore, the ownership will be agreed upon in the contract, independent of who is in charge of its development. With the objective of stating clearly the rights and obligations, the contract must:

- establish with clarity the contractual figure, i.e. type of contract to regulate the software development (pr_{11}).
- state clearly and concretely the entitlement of the product development rights (pr_{12}).
- establish measurements of preventive character that provide with efficacy for the contract: penal and arbitral clauses (pr_{13}).
- agree on the confidentiality in two ways: in relation to the developed product and in relation to the acquired knowledge in the process of development environment (pr_{14}).
- establish the entitlement of the rights as the mechanism in terms of evolution, i.e. who will be the ownership of the product after modifications, updates or improvements (pr_{15}).

4.1.2. Internal Personnel

It is needed, in the first instance, to state the ownership of the software product to be developed, i.e. its exploitation rights. According to European regulations about royalties (WIPO 2004), the ownership belongs to the company provided that it was developed by employees carrying out their duties or following instructions from the managers. However, if the software was developed by external personnel with other duties not specific to that project, 50% of the ownership belongs to the company and the other 50% is equally shared among all the personnel who developed the product.

Therefore, legal assurance activities that need to be adopted during the contracting stage must focus on clarifying the ownership of the software product. Assurance activities must be directed to:

- Form, by means of official contract models and clauses that state which employee duties must be included as part of the software development process, avoiding personnel signed for other duties (pr_{21}).
- State in writing concrete instructions directed to the developer in relation to the current project (pr_{22}).
- Protect adequately the development in terms of confidentiality agreements (pr_{23}).

4.2. Requirements

4.2.1. User Requirements Document

The user requirements document (URD) is a especially important document Shari Lawrence Pfleeger 2005 for legal security aspects of the project. In the URD, functional and nonfunctional requirements are stated. Therefore, it is possible to find most of the obligations of the organization carrying out the development, and reciprocally, the rights of the consumer in relation to functional and nonfunctional characteristics that the software must comply with. Legal assurance measures in this phase need to focus on auditing the final version of the URD. Assurance activities in the URD need to take into account the following considerations:

- The URD needs to state clearly the features that the software must comply with. This is an important point as it describes the contract object in detail (rr_{11}).
- The URD needs to establish that the software development organization has the personnel and knowledge to carry out the project (rr_{12}).
- The URD states an explicit agreement between the organization developing the software and the client, i.e. the document must be signed off (rr_{13}).

4.2.2. Traceability Document (rr_{21})

Once the URD has been completed, from a legal point of view, it is a desirable practice to create a traceability document between the URD and the requirements specification document (RSD) (Bray 2004) in such a way that every requirement found in the URD has a correlative requirement in the RSD. The objective of this document is that technical specifications correspond to agreed

DOI: 10.1002/spip

features between the organization carrying out the development and the client.

The creation of this document is not only essential for technical reasons to the organization carrying out the development, but it also serves as evidence of an appropriate development process. In the event of a legal claim from the customer, both the URD as well as the traceability document will allow the organization to prove the quality and adaptation of their development process.

4.2.3. Prototyping

Some development projects create a prototype during the requirement stage to find, refine or agree to the requirements. Prototypes can include some functionality that can confuse clients in relation to the achievement of milestones. Also clients could consider the prototype as an early or temporary version until the final product is completed (Kendall 2005). To avoid these type of claims, it is necessary to create a document with the prototype with the following considerations:

- The prototype is used only as a tool for gathering requirements (rr_{31}).
- The delivery of the prototype of does not modify any of the deadlines already agreed upon in the contract or URD (rr_{32}).
- The prototype lacks technical quality to be incorporated into production or even to use any functionality properly (rr_{33}).
- The client will exonerate the organization developing the product from any responsibility of using the prototype for any purpose other than requirement elicitation (rr_{34}).

4.3. Design and Development

During the design and development stage, it is necessary to consider three groups of activities:

- A first group of activities needs to be related to ensure the ownership of the product.
- A second group is to avoid the failure to comply with legal requirements.
- The final group of activities needs to verify that the design or the projected development does not infringe on algorithms protected by patents or intellectual properties (http://www.european-patent-office.org/index.en.php).

To deal with the first group, during the software development stage it is advisable to include elements to prevent illegal copies and state the ownership of the product. There are two main techniques:

- Stenography: fingerprinting or watermarking consists of introducing a small data file in the digital image or text (dr_{11}).
- Introduction of innocuous, unnecessary and implausible code (dr_{12}).

In the case of dealing with personal data, regulations regarding data protection must be taken into account (95/46/EC 1995). The design and development should include elements such as:

- procedures for identification and authentication of the users (dr_{21})
- updated information of who has access to users data files (dr_{22})
- back-up procedures (dr_{23})
- data transfer procedures and networks such as virtual private networks (dr_{24})

4.4. Deployment

During the software deployment phase, it is needed to consider three groups of activities:

- The first group is again related to the ownership of the product.
- The second group is related to avoiding confrontations with clients having clear definition of rights and obligations in the contract or licenses.
- The third group of measures must deal with the ownership of the product once updates and improvements have been performed.

In relation to the first group and before marketing the product, it is necessary to prove the ownership of the product. This can be achieved with the following measures:

- Register the product with Intellectual Property offices (http://www.wipo.int/portal/index.html.es) or the corresponding patent offices (http://www.european-patent-office.org/index.en.php), (http://www.uspto.gov/) (der_{11}).
- Register with a notary or escrow contract the project contents, graphical design, source code or any identifier of the software (der_{12}).
- Patent the product. It is possible to patent systems or applications such as cart lists, or electronics auctions (der_{13}).

DOI: 10.1002/spip

- Insert the copyright symbols, i.e. 'Copyright © 2006. Spain. All rights reserved' (*der*$_{14}$).

For the second group, marketing assurance measures depend on the segment to which the product will be offered. General measure could be divided into the following two types:

- Turnkey software development projects. Generally, the relationship between the parties is agreed upon in a contract and corresponding appendices at the beginning of the project. The following measures should be taken into account:
  - Documents of transferring and reception with the approval of the client stating a trial period (*der*$_{21}$).
  - Stating a product guarantee and maintenance contract skating who is responsible of maintenance costs and under what conditions. It is advisable to distinguish between different types of maintenance, i.e. corrective, adaptive, perfective and preventive maintenance (*der*$_{22}$).
- For projects that will customized for commercialization, we need to consider the following measures:
  - Include a software license with the general rights and obligations from both parties (*der*$_{21}$).
  - In the customization process, it is needed to adopt the measures from the previous point (*der*$_{22}$).

For the third group, once the deployment or commercialization has been carried out, during the maintenance phase, the product can be modified, updated or improved. It is again necessary to state the ownership of the product. To do so, the following measures can be adopted:

- Write a maintenance contract and state who is the owner of such modifications or improvements (*der*$_{31}$).
- Register with a notary the ownership of the modifications such as register with a notary such as contents of the project, graphical design, source code or any other element that can identify the modifications of the software (*der*$_{32}$).

As we have stated previously, this is a generic process model of legal assurance issues that covers only specific eventualities, and every software project must define its own process depending on extrinsic and intrinsic factors.

5. EVALUATION OF LEGAL ASPECTS IN IT PROJECTS: MEASUREMENT OF THE LEGAL PROTECTION

As well as establishing the legal risks for each project and specific software lifecycle phase, it could be very useful to calculate quantitatively the degree of legal protection of a software project. To do so, we have developed an assessment model. The variable Software Legal Assurance Percentage (SLAPe) measures quantitatively the percentage of legal protection of a software product. The SLAP measure is calculated as follows:

$$\text{SLAPe} = \frac{P + R + D + DE}{4}$$

where P evaluates the legal protection in the planning phase, R in the requirement analysis phase, D in development and DE in the deployment phase.

5.1. Evaluation of the Legal Protection in the Planification Phase

To obtain the parameter P the following equation is used:

$$P = \left(\sum_{i=1}^{n} \frac{p_i}{i} \right) \times 100$$

where $n = 1$ if there is only internal staff, $n = 2$ if there is also external staff and ($i = 1$ with internal staff; $i = 2$ with external staff)

$$n = \begin{cases} 1 \text{ if there is only internal staff} \\ 2 \text{ if there is also external staff} \end{cases},$$

$$i = \begin{cases} 1 \text{ with internal staff} \\ 2 \text{ with external staff} \end{cases}$$

and p_i is calculated as follows:

$$p_i = \sum_{j=1}^{m} \frac{pr_{ij}}{m}$$

where pr stands for planning risk, $m = 5$ in the case external staff or $m = 3$ in the case of internal staff.

DOI: 10.1002/spip

5.2. Evaluation of the Legal Protection in the Requirements Phase

The parameter R is obtained as follows:

$$R = \left(\sum_{i=1}^{n} \frac{r_i}{i} \right) \times 100$$

where $n = 2$ except when a prototype is being developed, in such a case $n = 3$. The variable r_i is calculated as:

$$r_i = \sum_{j=1}^{m} \frac{rr_{ij}}{m}$$

where rr stands for requirement risk and if $i = 1$, user requirement $m = 3$; if $i = 2$, traceability, $m = 1$; if $i = 3$, prototyping, $m = 4$.

5.3. Evaluation of the Legal Protection in the Development Phase

The parameter D is obtained as follows

$$D = \left(\sum_{i=1}^{n} \frac{d_i}{i} \right) \times 100$$

where $n = 2$ in all cases except when the software handles personal data, and in such a case $n = 3$, and d_i is calculated as:

$$d_i = \sum_{j=1}^{m} \frac{dr_{ij}}{m}$$

For $i = 1$, for stenographic techniques, $m = 2$; if $i = 2$, general data, $m = 4$.

5.4. Evaluation of the Legal Protection in the Deployment Phase

In a manner similar to DE is calculated as:

$$DE = \left(\sum_{i=1}^{n} \frac{de_i}{i} \right) \times 100$$

where

$$de_i = \sum_{j=1}^{m} \frac{der_{ij}}{m}$$

The variable der is the deployment risk, and for $i = 1$, register, $m = 4$; for $i = 2$, software type, $m = 2$, (this is an optional attribute); and finally for $i = 3$, maintenance, $m = 2$.

For each of the individual values of the variables, P, R, D, der, if the aspect is verified, its value will be equal to 1, otherwise 0.

6. CONCLUSIONS

For financial, commercial or product quality reasons, a suitable legal assurance process is a management aspect that cannot be ignored by the organizations developing information systems or subcontracting such services. The results presented in this work have the objective of improving the professional practice of legal risks management for the information systems industry. To do so, it is necessary to analyze systematically all legal assurance activities and measures that can jeopardize a project and align them with current software process assessment and improvement models. With this aim, a series of transversal activities to be included within the software lifecycle process have been described. To structure the legal assurance activities as a process model, CMMI has been considered as the starting point. Finally, we also presented a quantitative model to assess the legal protection.

ACKNOWLEDGEMENT

This work has been supported by the project MCYT TIN 2004-06689-C03 (IN2GESOFT)

REFERENCES

CMMI-SE/SW/IPPD/SS, V1.1 Capability Maturity Model Integration. CMMISM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing, Software Engineering Institute, University of Carnegie Mellon, 2002.

SPICE 15504. (ISO/IEC 15504) 2004. Information Technology-Software Process Assessment.

Software Process: A Roadmap. Alfonso Fuggetta, 22nd International Conference on Software Engineering (ICSE'2000). Future of Software.

Appendices, Glossary. CMMI, v1.1, Software Engineering Institute, University of Carnegie Mellon, 2002.

Kamil I. 2003. Intellectual Property – A Power Tool for Economic Growth. WIPO.

DOI: 10.1002/spip

Records of the Intellectual Property Conference of Stockholm. 1967. World Intellectual Property Organization (WIPO), Número de publicación OMPI: 311, June 11 to July 14, Stockholm.

Board 91/250/CEE, Official Diary of the European Communities (DOCE) The Council of European Communities – Series L 91/122/42 of May 17, 1991 – Sig. Room T.

http://www.uspto.gov/. 2006. United States Patent and Trademark office.

WIPO. 2004. *Intellectual Property Handbook: Policy, Law and Use*. WIPO, Upper Saddle River, NJ.

Shari Lawrence Pfleeger. 2005. *Software Engineering: Theory and Practice*, 3rd edn. Prentice Hall, Upper Saddle River, NJ.

Bray IK. 2004. *An Introduction to Requirements Engineering*. Prentice Hall, Upper Saddle River, NJ.

Systems Analysis and Design. Kenneth E. Kendall and Julie E. Kendall. Prentice Hall: 2005.

http://www.european-patent-office.org/index.en.php. 2006. European Patent Office.

Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. The Council of the European Communities, Official Journal of the European Communities of 23 November 1995. No L. 281, 31.

http://www.wipo.int/portal/index.html.es. 2006, WIPO.

DOI: 10.1002/spip