Electronic Communications of the EASST
Volume 33 (2010)

# Proceedings of the
# Fourth International Workshop on
# Foundations and Techniques for
# Open Source Software Certification
# (OpenCert 2010)

## Damages and Benefits of Certification:
## A perspective from an Independent Assessment Body

Mario Fusani, Eda Marchetti

12 pages

ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

# Damages and Benefits of Certification: A perspective from an Independent Assessment Body

**Mario Fusani**[1], **Eda Marchetti**[2]

[1]Systems and Software Evaluation Centre and [2]Software Engineering Laboratory
ISTI-CNR, Pisa, Italy

**Abstract:** The paper investigates on the nature of software certification and its reasons of being. The numerous factors that impact on the achievement of its purposes are discussed, and also compared in the cases of Proprietary Software and Open Source Software. Some relevant features of a certification process for Open Source Software are finally proposed.

**Keywords:** Certification, Standardisation, Open Source Software

## 1 Introduction

Traditionally, software certification has started as a need about proprietary software. The objective of such a need has been the achievement and transfer, among the numerous parties involved in software production and use, of the confidence that a software-related product or service actually possesses declared behavioural, in case structural, characteristics.

The reason for considering here the software along the proprietary/non-proprietary perspective is that proprietary software, or Closed Source Software (CSS), and its ”opposite” Open Source Software (OSS), have considerable mutual impact with certification.

We are not investigating here the nature of OSS, object of endless streams of literature. The purpose of this paper is instead to show a research line for determining what OSS certification can be, which are the factors that impact in its goals, and how could such factors be put under control.

To do so, the reasons for software certification and its very concept, that has been maturing in the CSS realm for many years, have to be revisited. This is because it seems that the certification concept and its implications, yet extensively spoken about, have not been discussed deeply enough in literature and even little understood, and it is this lack of understanding that, in our opinion, carries more weight to the ”damages” side of the balance damage/benefit for all certification users.

We believe that the various stakeholders around both CSS and OSS, especially OSS, can benefit from added-value of certification, but such benefits depends on a number of ways certification process is defined and conducted. Inappropriate goals and use can bring more disadvantages than advantages, especially to end-users. End-users are our favorite stakeholders when we speak of certification, because it’s us (us who can enjoy in our daily life, although indirectly, of services generated by CSS and, more and more, by OSS).

The objective of investigating the various aspects (both in terms of goals and means) of OSS certification cannot be adequately achieved without analysing first them in the case of CSS, where they have been established for long, and, even before that, in the more general case of products and processes. The risk of being biased by CSS-related issues is low if we keep in mind the final OSS objective.

Thus, the rest of this paper is organised as follows:

In Section 2, the basic concepts of certification are re-visited in the light of the 25-year experience of the Systems and Software Evaluation Centre working with CSS. This overview is believed to be useful because, as mentioned above, most often ”certification” is used as a term and done as a practice, but without a clear notion of what it is about.

In Section 3, factors (technological, managerial, knowledge-oriented) that impact into the achievement of the CSS certification goals are evidenced and discussed.

In Section 4, the goals of CSS and OSS certification are compared, and factors and impacts considered in Section 3 are checked for survival or adaptation for the OSS case. New factors, pertinent to the OSS certification goals, are introduced. Pros and cons of various factors impacting in the OSS certification are discussed.

In Section 5, concluding remarks are drawn and future developments lines of the current research are sketched.

The abbreviations used in this paper are recollected at the end, in Table 6.

# 2 Basic concepts and practice of certification

## 2.1 What is certification?

Many representatives of various professional backgrounds use the term ”certification”, possibly with different meanings. Official definitions, as we see below, contain expressions that must be interpreted in turn. For instance, in that from the ISO Guide [ISO96]:

*A procedure by which a third party gives written assurance that a product, process or service conforms to specified requirements*

some terms need to be clarified, to avoid too many different interpretations:

*Third party* seems to be a crucial role in the *procedure*, and will be discussed below as a member of the stakeholders list.

*Assurance* can be given as a result of an activity, the conformity assessment, defined in the same Guide but perfected by the Standard ISO/IEC 17000 [ISO04] as follows:

*An activity that provides demonstration that specified requirements relating to a product, process, system, person or body are fulfilled.*

There are more acceptable definitions, but the mentioned ones are sufficient to illustrate the concept. The terms that need to be interpreted are related to most of the factors determining how the certification goals are achieved and will be discussed in the next two sections. Notice that nothing like a guarantee is mentioned in the definitions. Typically in software technology, the word ”guarantee” has not been welcomed ever since this standard was defined (in facts, vendors have been preferring to issue a good deal of variants on the use of disclaimers).

We want to observe that in most technologies, including software, the mentioned *demonstration*, can be better and more realistically understood as ”confidence transfer” among certification stakeholders (for example, from a third party organism to an user) and this, as we mention in the following, can work as a value-added aspect of certified objects such as products and processes.

It seems that the confidence and confidence-transfer concepts, probably because of apparent self-evidence, has not been deeply discussed in literature: mostly it was insisted in describing tests and measures [Voa00] [Tri02] [MTT09], and little attention was paid to the ”confidence” aspect itself. This is probably one reason why certification was often mistaken by users as a guarantee, and by producers as particularly severe verifications and validations. If we consider the software, then the confidence is generally weakened, and it can be observed again that responsibility disclaimers associated to products are much more frequent than certificates. Even the term ”certification” was somewhat banished in the US for years, because of the unspoken threat that customers unions could claim refunds for unsatisfying certified services.

Certification cannot be well explained without reporting the most general scenario in which it usually happens. Figure 1 shows, in a rather simplified way, how a stakeholder such as a Certification Body (CB) operates according to two distinct categories of standards (one for the CB process, another for object references). The actual certification targets are properties of some types of entities, such as products, processes, people, environment. CB credibility, one of the bases of the mentioned confidence (see also next Section), is supported by CB accreditation according to a refereed process, via International Accreditation Bodies (AB) that, in turn, monitor each other in a periodical peer-to-peer process. Other stakeholders such as suppliers and users at various levels are not shown here, but their relationships in the scheme can be rather easily figured out. A more comprehensive discussion can be found in [FFL06].

### 2.2 Confidence spreading among stakeholders

If we accept the principle that confidence and confidence passing is the bottom-line of the value of certification and its reason of being, an investigation about its nature is necessary, before checking how it can vary in different contexts, mainly in the CSS and OSS cases.

One first research question is: Who should possess such confidence? If we scan a possible list of stakeholders, such as the rich one reported in [Tay09] for both Commercial Off-The-Shelf (COTS) software and OSS, we may think of various candidates including customers and possibly end users, but likely in different measures.

Other related research questions are: Where does confidence start? How could it be transferred? Can it, purposely or accidentally, change during transfer? And how? Working on such questions means finding out the elements or factors that determine not only the success of certification and then the possible added value it gives to products, but also the mechanisms by which confidence is passed and changed. To give a quantitative nature to such factors seems quite hard to achieve. An attempt to express their strength by an ordinal scale is made in the next Section.

Figure 2 shows a typical scenario where a supplier who is confident that the product to be supplied has certain properties, but wants the (generally non technical) customer share the same confidence. Here a CB acts as a catalyst in confidence passing from supplier to customer, who

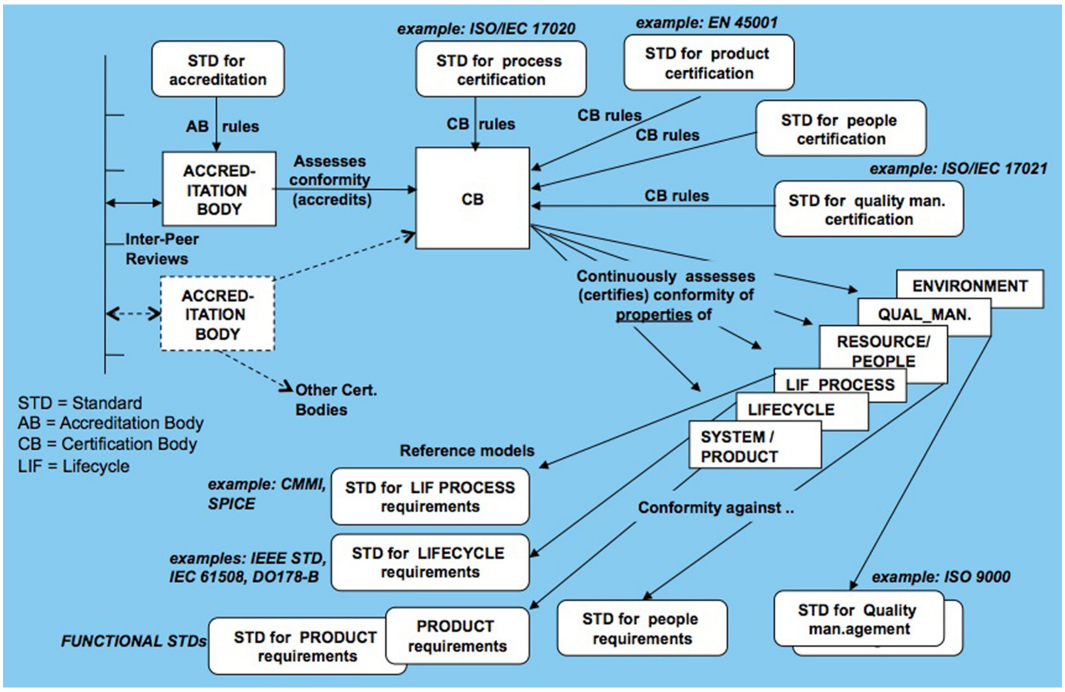

Figure 1: Simplified certification scheme

might be wanting or not to continue the transfer down to final users.

Two more remarks on confidence are useful before analysing what it can depend on.

First, there are different views of confidence, depending on the considered stakeholder. For example, a developer may want the confidence that the software is deadlock-free (this particular one can be achieved only by proof), whilst an end-user may be wanting that a software-related product behaves efficiently and safely, or, as a more subjective, imprecisely defined but more probable wish, that the product is ”fit for use”.

Next, we can be tempted of interpreting the *demonstration* mentioned in the definition given in Section 2.1 as ”proof” in mathematical sense. Just regarding the software, this is more realistic achievement now than used to be when the defining standards was last issued and reviewed, as formal methods and tools are becoming interesting, useful, and likely will be the most important technical ingredient of the certification process. However, the scope of formally provable statements, with all the necessary math to get to the final statement, is restricted only to portions of the chain ”user needs-requirements-architecture-code”. In many cases, more traditional, yet possibly automated means would be necessary, that are no proof, yet still there can be confidence. For example, a formal proof could be used to verify and then certify that a software has a property such as deadlock-freeness. If the property to be certified is maintainability, we would most likely have less rigorous but nevertheless workable alternatives, such as informal, procedure-guided analyses, simulations and tests. Confidence can be built, and should be, on proofs, but is generally no proof itself.

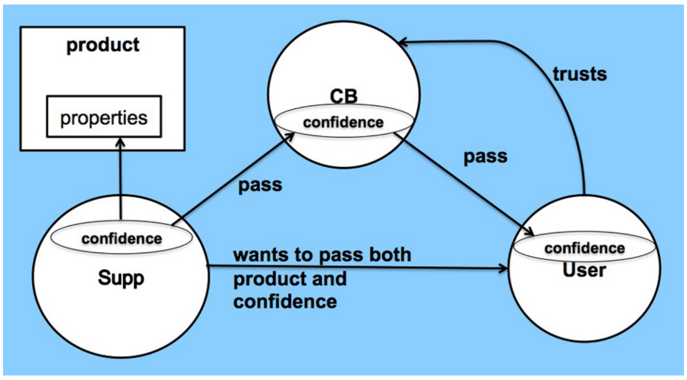

Figure 2: Confidence transfer

## 3 Certification goals: what their achievement depends on

We can now summarise an important certification goal in terms of confidence:

*Maximum, well-grounded confidence (that product properties declared by certifier are actual properties) can be transferred to stakeholders who 1) benefit of the products and associated services; 2) decide of products adoption.*

The same definition is simplified when the certifier is in one of the two stakeholders categories, but this is not the case in which confidence can get higher.

Now we can examine what elements, or factors, can facilitate or inhibit the achievement of this goal in various contexts. If certification is no proof (which is not), then the elements that give it some value (our factors) deserve to be investigated and discussed.

### 3.1 Impact factors in CSS environment

Whilst the certification goal is the same for both CSS and OSS, the factors on which it depends may vary to one another. We start examining the CSS-related factors, to check them also for the OSS case in next Section.

The following is a list of factors categories. It can possibly be an open-ended one, but it is sufficient for a comprehensive enough discussion of their impact on the certification goals.

- References used by CB
- Certification Objects
- Certification Process

References used by CB are those denoted as ”Reference models” in Figure 1.

Certification Objects are, in our case, properties of products to be certified.

Certification Process represents the activities to get to the certificate, executed mainly by the CB but also by other stakeholders.

Table 1: ”CB References” impact factors

| Factor | Supp | Cust | User | EUser |
|---|---|---|---|---|
| Requirements Specifications | L | M | H | H |
| Functional Standards | M | H | M | M |
| Life-cycle Standards | H | M | L | L |
| Product Quality Standards | M | H | H | H |

Table 2: ”Certification Objects” impact factors

| Factor | Supp | Cust | User | EUser |
|---|---|---|---|---|
| Functional | M | H | M | H |
| Performance | M | M | M | H |
| Architectural | H | M | L | L |
| Quality in use | M | H | H | H |
| Internal quality | H | H | M | L |

All these factors categories are expanded into lower-level factors in the first columns of Tables 1, 2 and 3, respectively.

In order to be helpful, the impact into the certification goal (that, we recall, is expressed by confidence acquired by stakeholders about the product) of each of the above factors categories, and especially of the lower-level factors, should be measurable. That is, it should be possible to find a mapping between each factor and a measure scale for its impact. Although this is a typical case of human judgement, for which measurability is questionable, we wish to express in some way the intensity of the impact. A scale could have a binary nature (there is impact or not), or could be expressed in an ordinal scale of judgement, for example {Low (L), Medium (M), High (H) impact}. Here we use the latter type of scale.

In Tables 1, 2, 3 only the most relevant types of stakeholders are taken into account, as targets of the various impact factors.

Limited space prevents discussion of the reasons for assigning all these measures, which are still an ongoing investigation. Some of these reasons are reported in the following discussion.

The information in these Tables should be interpreted in the following way:

”If Stakeholder S is informed (typically by a certificate) that:

- CB certified a product property against a Factor X of Table 1, then its confidence on the certification results is as entry (S, X) of Table 1;
- the certification object is Factor Y of Table 2, then its confidence on the certification results is as entry (S, Y) of Table 2;
- CB process has the same characteristic as Factor Z of Table 3, then its confidence on the certification results is as entry (S, Z) of Table 3.”

Table 3: ”Certification Process” impact factors

| Factor | Supp | Cust | User | EUser |
|---|---|---|---|---|
| CB is independent | M | M | H | H |
| CB is accredited on a refereed accreditation process | H | H | M | L |
| CB processes follow rigorous standards themselves | H | H | M | L |
| Tests are severe | H | M | L | L |
| Formal Methods used when possible | H | M | L | L |
| Reverse Engineering used when possible | H | M | L | L |
| Reference Standards easy to adopt | H | H | M | M |
| Reference Standards easy to check conformance | M | M | H | H |
| Traceable Product | M | L | L | L |
| Evidence of product lifecycle is available | H | H | L | L |

Confidence is build up as a list (not a sum because we have ordinals) of measures or scores L, M, H when multiple entries are enabled.

Notice that the factors regarding the Objects are directly related to the Reference factors.

### 3.2 Discussion on impacting factors in CSS environment

To start working on the research questions given above, a short discussion on the impacting factors follows.

Out of the yet limited set of stakeholders of the certification, the figures Cust (that may coincide with the User) and the EUser are more relevant for our declared purposes. The former is the entity thyat pays for the product, which is sold, possibly as a service, to the latter. Then usually EUser pays for all, and enjoys the good or service. Most often EUser is not able to judge about factors such as life-cycle or internal quality, and often does not even know that software is in the product / service, neither that a CB has certified it. When she/he knows that, say from an advertisement, that's where her/his only confidence comes from.

The factors shown in Table 3 are facts about CB and its actions. Just to comment on one of these, the factor ”Test are severe” does not necessary mean that CB executes tests. It may do it, as it can witness Supp's tests or thoroughly inspect Supp's testing documentation, make use of an independent, accredited Testing Laboratory, request re-execution of some tests or a combination of all that.

Those three lists of lower-level factors limit in some way the investigation area about the causes that determine the certification goals. Moreover, to direct the research along some line, we should further restrict to examine single cases, to try to generalise afterwords. Let us examine here just two cases in which certification can give misleading confidence to stakeholders.

Case 1:

What if, for the same product, the quality of the reference standards (see Table 3, entries 7 an 8) was not so good ? This fact could be missed by both Supp and Cust, that hardly look at such quality aspects: the main Supp's and Cust's concern about a Standard is compliance for liability reasons, and not Standard quality. Here a risk arises for EUser if the certificate was only issued for protect Supp against liability and for commercial reasons (certificates can be excellent advertisement), in case the Standard, yet technically correct, does not possess the qualities that facilitate its adoption and the certifiability of products against its (of the Standard) requirements. The probability of such a case is not a remote one [BCF+10], as EUsers representatives are rarely included in standard-making groups, a fact our Centre knows well, having been involved in both ISO and CENELEC standards working groups for years.

Case 2:

This is a typical problem our Centre has been found since long as an independent, if not yet accredited, product and process assessment body (Factors 1 and 3 of Table 3 hold, Factor 2 does not yet) . The target software product, a fiscal software for electronic cash registers (ECR) (fiscal software is now a reality in over 30 Countries), must be approved by the Government on the basis of evidence that some defined properties are certified by the Centre. Then, copies of the software (by the tens of thousands) are allowed to be installed in the ECRs. Any variant must be submitted again to certification before being installed in the field. The problem is how to monitor, in an inexpensive way, that no variant exists in the field, different from that that passed through the certification process. The problem, here reported in a rather simplified way, is hard to solve: check in Table 3, entry 9, that Cust and User have little notion about product traceability (even Supp may or may not have that notion). There are solutions, but not inexpensive ones, since they imply the definition and se-tup of a (virtual) network with identified and securely communicating nodes, that was not possible at the beginning of the regulations about fiscal software for ECRs. So, a variant, unchecked ECR software could be officially certified, and confidence of EUser would be quite high, but definitely ungrounded.

## 4 Impact factors in OSS environment

Tables 4 and 5, corresponding to the first and third higher-level factors, revisited, are shown below.

In this environment, the picture of the stakeholders and of their relationships is different. It seems important to evidence the CB because of its tighter connections with the other stakeholders. CB may well happen to work with User and then with the developers too [Voa00], because they are often also users, even if they take the suppliers' place. Developers could have been also shown in CSS environment, and were not just to keep the Tables comparatively simpler.

So, the boundaries among stakeholders roles become much more indistinct than in CSS, and what is shown from column 2 on in Tables 4 and 5 is nothing but emerging aspects.

The scenario gets more complicated as the Tester figure also emerges. In fact, OSS properties mostly get verified by testing in operational environment, especially for product selection and adoption, and also during actual service. Testing before delivery is only a fraction of the testing process.

Also, some of the factors mentioned in Section 3.1 are far less realistic (a consideration that is not worrying too much) and new factors become evident. Without the pretend to be exhaustive, we point out some most relevant differences.

We think important the possibility of having open source specifications in the references for the CB. In fact, in OSS the specifications are not any more controlled by a single organization but

Table 4: ”CB References” impact factors for OSS

| Factor | Dev | CB | Cust | Tester | User | EUser |
|---|---|---|---|---|---|---|
| Product Reqs Specifications | M | H | H | H | H | M |
| Open Source Specifications | M | H | H | H | M | H |
| Functional Standards | M | H | M | M | M | M |
| Life-cycle Standards | L | M | L | L | L | L |
| Product Quality Standards | M | H | H | H | M | H |

Table 5: ”Certification Process” impact factors for OSS

| Factor | Dev | CB | Cust | User | Tester | EUser |
|---|---|---|---|---|---|---|
| CB is independent | M | M | M | M | L | H |
| CB is accredited | M | H | H | L | L | M |
| standard conformant CB processes | M | H | M | L | L | L |
| Tests are severe | M | H | M | M | H | M |
| Formal Methods used when possible | H | H | M | M | M | M |
| Reverse Engineering used when possible | M | M | L | M | M | L |
| Reference Standards easy to adopt | H | H | H | H | H | M |
| Reference Standards easy to check conformance | M | M | M | M | M | M |
| Traceable Product | H | H | M | M | H | M |
| Evidence of product lifecycle is available | M | M | M | L | M | L |
| Use of collaborative tools | H | H | M | M | H | M |
| Automated process | H | H | H | H | H | M |
| Independent development | H | H | H | H | H | M |
| History of evolution | H | H | H | M | H | M |

they continuously evolve according to exigencies of individuals or companies, influencing therefore the certification process. Thus a certification process for OSS should reflect this adaptation, providing certificates with validity and a scope changing over the time.

The adoption of OSS also introduces the possibility of having independent development and the history of evolution.

The possibility of independent development from developers and companies having different skills could influence the certification process in different ways: from one side, high variability and evolution (versioning) of the same open source product could evidence unstable and not yet mature releases; from the other, the abilities and reliability of a specific (group of) developers could be part of the certification process itself and influence the final decision.

Also the history of the software evolution as well as any other information derivable by the current available OSI certification [Ope08] could be useful for a certification process. The certificates could be associated only to a certain evolution of the OSS and then related to a specific time and version.

Regarding life-cycle evidence, less impact here does not mean that we should have less interest in OSS life-cycle. On the contrary, if life means being alive, then OSS life-cycles can be much more alive than CSS. Such evolutionary dynamism does not fit well with most established standard and independent certification as it was shown in Sections 2 and 3. And all those verifications of requirements and design documents and their management are no more an issue.

Testing and configuration management processes are still there [OMK08], but their actors are geographically distributed. Moreover, novel, peculiar OSS processes are in place, related with web-based co-operative environments.

OSS life-cycle processes are only in part conformant to ”software lifecycle processes” [ISO08], but could be anatomically dissected and described accordingly: this may be a good promising derived research but is out of the scope of this paper. To this purpose we can refer to a recent paper [Bou10] that compares the development process of CSS with that of OSS.

Actors performing these processes are non-necessarily co-ordinating to each other, but the process can have much more impact if some collaborative tools are used among such stakeholders. In fact, we propose automation in certification process, to outline the characteristic of de - personalisation of decisions together with that of (at least virtually) centralised repositories for OSS product releases and certification records. In our opinion, CB independence and CB accreditation should hold. Only, the CB cannot assess the OSS life-cycle processes as it does in CSS, but basically analyses process results, including product versions, that were deposited in a repository by the OSS developers. Stakeholders such as Customers would not pick the products from developers to submit them to CB, but would get the certified software directly from a repository.

Re-considering the previous Case 1, we can notice the following for OSS:

End Users could be even more protected by defective standards in OSS, as potentially all the developers and users communities can have access to the ways a Reference Standard has been interpreted to certify a product. As all the stakeholders roles are intercommunicating, they can monitor all the certification process steps the CB carries on whose execution track can be recorded in the same repository that also collects the product versions.

And, re-considering previous Case 2:

Even it is hard to think that a public administration can be convinced at adopting secure OSS, this is not impossible. The mentioned problem could be resolved because all the variants would be taken from a repository in which only certified versions are available, even in the case the control has passed from Suppliers to the repository administrators or to privileged users, such as the CB.

### 4.1 Value of a virtual repository as a support of OSS certification

There would be rules for interacting with the repository. This would not alter the ”unruled” nature of typical OSS stakeholders, because repository interactions can happen only on a voluntary basis. Only, to get a (possibly dynamic) certificate, with all the advantages, made possible as long as the described factors hold, some firm modalities to act must be in place, to ensure the CB can operate.

This way, subscribing an OSS project to a virtual certification repository would be appealing indeed.

We insist on the unchangeability of some factors regarding the CB: i) independence, ii) having a process that follows publicly available standards for assessment and certification (adapted

Table 6: Abbreviations

| Abbreviation | Text |
|---|---|
| AB | Accreditation Body |
| CB | Certification Body |
| COTS | Commercial Off-The-Shelf |
| CSS | Closed Source Software |
| Cus | Customer |
| Dev | Developer |
| ECR | Electronic Cash Register |
| EUser | End User |
| OSS | Open Source Software |
| Supp | Supplier |

to OSS environment), iii) accreditation according to a well-known and refereed accreditation scheme, which implies periodical monitoring of the CB by a recognised accreditation body.

## 5 Conclusions

In this paper we presented an overview of the most important aspects of the CSS certification process and how they can be put in relation with the emerging OSS. As discussed in the paper, many of the impact factors of the CSS have to be readapted and rediscussed to face the innovations and the exigencies of the OSS context. In this paper we proposed a first tentative of revising the important factors of the CSS adding wherever is possible new and specific ones in relation with the peculiarities of the OSS process. What emerges from a first analysis is that the results of certification should not be unique anymore, but would show several degrees or scores depending on the characteristics of the OSS version considered.

It is therefore thinkable that the metrics for assigning a certificate could include either the evaluation of specific open source specifications or (and) evidences about the released version or (and) the reliability score of developers. The certification could also be related to the history of the OSS and would have a period of validity (start and end date), which is an evolution of the traditional certification policies.

To facilitate, and perhaps to make it possible, an OSS certification process, we think it is be important to have a repository in which, in addition to the conventional information, further and more specific data for certification analyses are included. This should involve information about the verification and validation activities performed as well as scores about the reliability attitude of the developers.

In the authors' opinion, one contribution of this paper is the summary of important aspects of the certification process for OSS. These aspects could not be completed here, and further analyses and improvements are necessary. Our intention was to make a first step on the construction of a OSS certification process trying to exploiting as much as possible of the evidences and the experiences of the CSS environment.

# Bibliography

[BCF$^+$10] I. Biscoglio, A. Coco, M. Fusani, S. Gnesi, G. Trentanni. An Approach to Ambiguity Analysis in Safety-related Standards. In *Proc. of QUATIC 2010 (7th International Conference on the Quality of Information and Communications Technology)*. September 29-October 2, 2010.

[Bou10] A. Boulanger. Open-source versus proprietary software: Is one more reliable and secure than the other? *IBM Systems Journal* 44(2):239–248, 2010.

[FFL06] F. Fabbrini, M. Fusani, G. Lami. Basic Concepts of Software Certification. In *Proc. of 1st International Workshop on Software Certification (CERTSOFT'06)*. Pp. 4–16. McMaster University, 2006.

[ISO96] ISO/IEC. ISO/IEC Guide 2:1996, Standardization and related activities  General vocabulary. 1996.

[ISO04] ISO/IEC. ISO/IEC 17000: 2004, ISO/IEC 17000:2004, Conformity assessment - Vocabulary and general principles. 2004.

[ISO08] ISO/IEC. ISO/IEC 12207:2008 - Information Technology: Software life cycle processes. 2008.

[MTT09] S. Morasca, D. Taibi, D. Tosi. Towards certifying the testing process of Open-Source Software: New challenges or old methodologies?. In *Proc. of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*. Pp. 25–30. IEEE Computer Society, 2009.

[OMK08] T. Otte, R. Moreton, H. D. Knoell. Applied Quality Assurance Methods under the Open Source Development Model. In *Proc. of the 32nd Annual IEEE International Computer Software and Applications Conference*. Pp. 1247–1252. COMPSAC, 2008.

[Ope08] OpenSource.org. OSI Certified Open Source Software. 2008.

[Tay09] R. Taylor. Understanding how OSS Development Models can influence assessment methods. In *Proc. of the Third International Workshop on Foundations and Techniques for Open Source Software Certification*. 28 March 2009.

[Tri02] L. Tripp. Software Certification Debate: Benefits of Certification. *IEEE Computer*, pp. 31–33, June 2002.

[Voa00] J. Voas. Developing a Usage-Based Software Certification Process. *IEEE Computer* 33:32–37, 2000.