Certification of Open Source Software – A Scoping Review

Eirini Kalliamvakou, Jens Weber, Alessia Knauss

Certification of Open Source Software – A Scoping Review

Eirini Kalliamvakou[1], Jens Weber[1], Alessia Knauss[2]
1 University of Victoria, Department of Computer Science, 3800 Finnerty Rd, Victoria, BC V8P5C2, Canada
{ikaliam, jens}@uvic.ca,
2 Chalmers University of Technology, Department of Computer Science and Engineering, Hörselgången 5, 41296 Gothenburg, Sweden
alessia.knauss@chalmers.se

Abstract. Open source software (OSS) systems are being used for increasingly critical functions in modern societies, e.g., in health care, finance, government, defense, and other safety and security sensitive sectors. There is an increasing interest in software certification as a means to assure quality and dependability of such systems. However, the development processes and organizational structures of OSS projects can be substantially different from traditional closed-source projects. The distributed, "bazaar-style" approach to software development in OSS systems is often perceived incompatible with certification. This paper presents the results of a scoping review on certification in OSS systems in order to identify and categorize key issues and provide a comprehensive overview of the current evidence on this topic.

1 Introduction

As the use of software expands to an increasing number of domains and products, it is also entering areas where, either by law or best practice, software must conform to certain rules and standards. Examples of such domains are healthcare, defense, and the increasingly important domain of transportation (e.g., self-driving vehicles). Due to the criticality of the systems and the high risk associated with software malfunctions in such domains, high standards in terms of the software's security, reliability, and safety have to be maintained [26]. Additionally, domains such as government and public administration, although not necessarily operating critical systems, have in many countries requirements in place that ask for some guarantee of the software's reliability and proper licensing before OSS can be used. The European Union, for example, introduced the OSEPA project (Open Source software usage for European Public Administration, http://osepa.eu) in order to systematically discuss issues related to OSS adoption in public administration. Their list of deliverables includes case studies and surveys regarding the successful adoption of OSS in public administration and the critical factors behind it. Technical characteristics that OSS must fulfill, software quality, and licensing are high on the list of priorities.

A common misconception regarding software certification is that it ensures the absence of defects [18]. This is not part of a certification's goal and it is also far from its reach. At most, software certification can attest, through a series of evaluation activities, that certain properties exist in the software, and that it conforms to specified standards. These properties can be associated with the end product or the process that lead to it, or both. The properties are assessed and verified against standards that must be met, and an awarded certificate is equivalent to assuring that the software has these properties. Today, software is mostly certified based on the process used in its development, which seems insufficient to verify software used as part of critical systems and devices [26]. The product needs to possess certain properties too, and therefore should be certified based on evidence that comes from the software code.

Open Source Software (OSS) is a special category of software in terms of distribution and licensing. The Open Source Initiative (OSI, http://opensource.org) provides the open source definition, which essentially lists the characteristics that software must possess to be considered open source. OSS started based on the idea that software should be openly accessible to all, and by people who practiced this principle of creating software that they released through the Internet visibly and openly. As a side effect, communities of like-minded developers started forming around OSS projects, a fact that has received much academic attention [3, 13, 27]. Today, OSS development is considered mainly community-driven and sustained through the volunteers that create and maintain the software. With the increasing adoption of OSS and the piling evidence of the high quality of OSS [1, 22, 28], the ability to adopt OSS instead of proprietary software in critical domains often hinges on issues of software certification.

The objective of this paper is to study existing literature systematically and present a comprehensive overview on experiences and issues related to the certification of OSS systems. We present the results of a scoping review of the literature on this topic. Our findings include 17 relevant papers on the topic of certification of OSS systems.

This paper is organized as follows: First we present our method, questions, and protocol for searching the literature regarding software certification in the OSS domain. We then present and discuss our findings. Finally, we will draw conclusions and acknowledge potential limitations of this study.

2 Research Method

We use a systematic scoping review method to gather published evidence on certification issues in OSS systems. According to Rumrill et al. scoping reviews "focus on examining the range and nature of a particular research area" [21]. Scoping reviews are often used as a pre-cursor to generate more specific research questions, which could be addressed by more in-depth Systematic Literature

Reviews on a particular issue or hypothesis [17]. Both methods have in common that they follow a well-defined methodology that minimizes bias and allows repeatability. This includes the definition of a research protocol that specifies the research questions, search strategy, inclusion/exclusion criteria, and the information to be extracted from the retrieved literature. The search and review methods are documented to allow the reader assessing rigor and repeatability of the study [16].

2.1 Objective and Research Questions

The objective behind conducting this scoping review is to gain an overview of the research and discussion in the field of software certification related to the cases of open source software projects. The research questions in scoping reviews are generally more abstract, aiming to give an overview of existing literature. Hence, we focus on identifying the amount of research activity, who is leading the research, the topics covered and the approaches or solutions used in OSS certification.
We formulate our research questions as follows:

RQ1: How much research activity is there in the area of OSS certification?
RQ2: Who is leading the research on certification of OSS systems?
RQ3: What issues/topics of certification for OSS have been identified and studied?
RQ4: What approaches or solutions have been proposed to address these issues?

To answer these questions and conduct the scoping review we devised a search protocol defined as follows.

2.2 Sources and Keywords

We performed our search by using digital collections of publishers and organizations relating to software engineering and computer science. We used the following databases to acquire the primary studies:

- IEEE Xplore
- ACM Digital Library
- Wiley InterScience (Computer Science section)
- Science Direct (Computer Science section)
- SpringerLink

These digital libraries are widely used and well established in the software engineering and computer science domains. A consultation with a subject librarian for computer science at the University of Victoria confirmed that this was a sufficiently comprehensive list of digital libraries to be used for our study. Meta

search engines such as Google Scholar and CiteseerX were used as validity checks in an attempt to retrieve any relevant publications not covered by the above collections.

The following query was used in our study, targeting both title and abstract of the publication: "open source OR FLOSS OR (Libre AND software OR project) OR (Free AND software OR project)) AND (certification OR certify)".

2.3 Search and Selection

Our search includes all publications that have been added to the publishers' digital collection up to April 2013. The following inclusion and exclusion criteria were applied to filter the query results:

In order to be considered for *inclusion*, papers were required to fulfill all the following requirements:

- Abstract and/or title contain the keywords as defined in our search string.
- Papers are published in journals, conference proceedings, or are book chapters. We also included papers that are part of grey literature (technical reports, white papers etc), although our search did not yield any. We did not include magazine papers unless they had academic references and were peer-reviewed.
- Software certification is the main theme of the paper and refers to OSS projects. Being the main theme is evidenced by the certification mentioned in more than one third of the pages of the publication.
- Publications are in English.
- The full paper content is available in the collection (not just its abstract).

In turn, papers were *excluded* if they failed to fulfill at least one of the above criteria.

Papers that met all above inclusion criteria were reviewed in full text to make a final decision on their relevance to this scoping review. We included papers that discuss the process and activities of obtaining some form of third party certification for any type of OSS. We also included papers that discuss or propose the use of tools, methods, approaches, and frameworks in OSS projects that seek certification. Another type of paper we were interested in would discuss possible changes and extensions in the way assessment is carried out for software certification in the case of OSS. Papers that offer positions or debate on how certification affects OSS and vice versa are also accepted, even when the proposed solutions are not fully validated since certification issues in OSS are a relatively recent issue and the status of the research is still formed.

We did not include papers that use the term certification in any other context than software receiving an assurance from a certification authority or body, based on conformance to specified standards. For example, we do not include papers that discuss certification in the context of network security and related tokens or certificates.

When reviewing the full text, we attempted to extract the issue discussed and the solutions proposed. If the issues and the solutions genuinely link to software

certification and not just the general software engineering domain, we include the paper. This criterion is reinforced by the assessment of how central the software certification theme is in the paper.

3 Results

In this section we present the findings of our study for each research question defined above.:

RQ1: How much research activity is there in the area of OSS certification?

Querying the online libraries with the search string defined in our protocol resulted in 114 papers after removing duplicate entries. Based on our inclusion and exclusion criteria, 11 of the identified papers were classified as relevant giving an inclusion rate of 9.6%. After performing the validity checks using meta search engines defined in our protocol, we identified an additional source that was not indexed in the selected digital libraries. This was an open access journal, the Electronic Communications of the European Association of Software Science and Technology (ECEASST), which hosts for publication some of the papers included in the proceedings of the International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert). We used the journal's search function to repeat our search protocol. The search through this additional source yielded 11 papers, 6 of which met our inclusion criteria. As a result, the final number of papers yielded by queries was 125, with 17 selected primary studies (inclusion rate 13.6%).

There were a few cases of relevant papers that refer to studies being part of the same research. Authors of the 17 primary studies have multiple publications on the same research and refer to them in some of the 17 papers. For now, we have included them in our list of relevant primary studies. However, in our refinement steps, we will carry out cross-referencing of the included papers, and we will categorize these sub-studies as secondary papers.

The results of our study show that the earliest (one) publication was in 2008. 4 out of the 17 papers were published in 2009, 6 papers in 2010, 5 papers in 2011, and 1 paper in 2012. Of the 17 included papers, 11 appeared in conference or workshop proceedings (64.7%) and 6 were journal articles (35.3%).

RQ2: Who is leading the research on certification in OSS systems?

Although with not a substantially higher number of publications, USA is leading the research efforts regarding certification in OSS. USA has 5 out of the 17 relevant publications. Interestingly, 4 out of the 5 publications involve at least one common author. Italy and the UK are the next in line with 4 and 2 publications respectively,

out of the list of included papers. The distribution of publications among the leading countries is shown in Table 2.

Table 1. Leading countries in OSS certification research

Country	# of Papers	Paper ID
U.S.A.	5	P6, P8, P9, P10, P11
Italy	4	P1, P3, P13, P16
U.K.	2	P5, P7

Ten of the primary studies originated in a particular safety critical domains, while seven are not domain specific. Most domain-specific papers target healthcare (5 out of 17 papers), followed by transportation (4 out of 17 papers).

RQ3: What issues of certification for OSS have been studied?

OSS certification compared to closed source. Several researchers have studied the differences of certifying OSS compared to the certification of closed source software. Fusani and Marchetti [10] point out that the stakeholder groups involved in the two kinds of software are different and gain confidence in the software through different sources of information and in different ways. The authors discuss the different factors that impact the closed source and open source environments and how they should be taken into account in the certification process. They observe that the evolution of OSS is much more dynamic and not as linear as in closed systems. Fabbrini et al. make similar observations but also present two concrete process scenarios for OSS certification, the first being developer-initiated and the second being client- (i.e., adopter-) initiated [8].

Kakarontzas et al. point out that OSS is more amenable to product-focused certification than to process-focused certification, because the development processes of OSS are often less tightly controlled than in closed source systems [11]. Moreover, the availability of the source code provides a better basis for product-focused evaluation. Feuser and Peleska [9] make similar argument, suggesting that the openness of OSS provides opportunities for open proofs that certification objectives are met. Moreover, extensive peer reviewing by a large community may improve quality assurance. Organizing this "crowd-sourced" review and certification process is a challenge. Khoroshilov suggests integrating it in the educational process of software engineering trainees [14]. Cerone and Settas propose the concept of a community-driven process of generating anti-patterns to be used for quality assurance [5].

Morasca et al. discuss the specific differences between the testing processes of OSS in contrast to closed source systems. For example, OSS testing processes can be guided by metrics such as code coverage and the results of static analyses, e.g., potential defect density, data and control flow analysis, program slicing etc. [19].

Certification economics. Comar et al. raise several common challenges with the certification process of safety critical software [6]. Certification commonly happens in discrete and costly steps and, once certified, the system is commonly closed to changes and adaptation to avoid the need for recertification. This effect is referred to as the "Big Freeze". To overcome this problem, they propose an approach to continuous certification, called the Open-DO process. Properly implemented the Open-Do process ensures that a system is certifiable at any time. Open-Do is supported by a suit of open source tools.

Cotroneo et al. consider economic aspects of certification from a different perspective, namely from an adopter's point of view [7]. The challenge here is to select and certify the best open source product from a potentially larger set of available systems. They propose definition and use of a pre-certification kit (PK) to filter out suitable candidates. A PK is generated in a two-step process, by firstly specifying a reference model that captures the requirements on the type of system to be certified (e.g., an operating system, a health record, etc.), and secondly selecting software metrics to be extracted from candidate systems, indicative of whether or not the specified requirements are being met.

Development process. Bertrand and Fuhrman discuss the suitability of OpenUP, an OSS development process adapted from the Unified Process, as a foundation for developing certifiable software [4]. They specifically attempt to align OpenUP with DO-178B, a certification standard used in avionics, and point out arising challenges, e.g., with respect to the use of different terminologies. Kakarontzas et al. introduce the OPEN-SME process, which emphasizes activities that prepare OSS for software reuse and focuses on generating trustworthy, product-focused evidence on software quality attributes [11].

OSS for evolving complex standards. Several authors have highlighted the role of OSS as an enabler for the development of standards in inherently complex application domains. Sethi et al. discuss interoperability and standardization issues in community tele-medicine [23]. They argue that a commonly accessible OSS interoperability framework may be more effective in enabling industry to produce certifiably interoperable technologies than a set of abstract standard specifications. Van der Leest uses an OSS prototype to study design alternatives and implementation trade-offs of the ARINC 653 standard used in the avionics domain [25].

OSS security. The certification of software security is an issue discussed by a number of authors. Smith et al. criticize the "security by checklist" approach commonly used in certifying closed source systems [24]. They expose security vulnerabilities in open source Electronic Health Record (EHR) software that would have been undetected by current certification programs. Similar studies have been published by Austin et al. [2], Helms and Williams [12], and King et al. [15].

RQ4: What approaches or solutions have been proposed to address these issues?

The previous section discussed major common themes of issues discussed in the selected primary studies. Table 3 gives a comprehensive overview of these issues and summarizes the proposed approaches or solutions for each paper.

Table 3. Summary of Issues and Solutions per paper covered in Primary Studies

Paper	Issues	Solutions
P1- [19]	Selecting **testing processes for OSS** vs. closed source software	**Evaluate maturity** stage of project testing process, improvements based on OSS characteristics
P2 – [6]	**"*Big freeze*" effect** after certification, to avoid need for re-certification	**Continuous integration**, software certifiable at all times, use of open tools and standards for certification-relevant material
P3 – [7]	Producing evidence for **certification is costly**, specific focus on operating system software	Use a **pre-certification kit** to evaluate properties the OS, including metrics and acceptable values
P4 – [4]	**Government-set standards** for civil avionics software (**DO-178B**)	**OpenUP process framework** as a checklist to prepare for certification, project can customize processes on top
P5 – [1]	Standard compliance checking, specific focus on CASE tools	**Compliance test generation** from standards specification
P6 – [25]	**Studying design alternatives for ARINC 653 certification is difficult** because platforms are closed & proprietary	**OSS prototype implementation** allows examination of benefits and weaknesses of design and architectural alternatives in using virtualization to achieve ARINC 653
P7 – [23]	Difficulty of **developing interoperability standards in complex domains**, e.g., telecare	**Open source framework** for designing communication standards
P8 – [24] P9 – [2]	"**Security by checklist**" fails to detect implementation level vulnerabilities	Enhancing existing test scripts to **include implementation level vulnerabilities**. Source code required
P10 – [12]	**Certification of access control** in secure software, focus on medical information systems	**Systematic method for product-focused assessment** based on access control criteria compiled from different standards
P11 – [15]	**Certification of secure audit mechanisms**, focus on medical information systems	**Systematic method for product-focused assessment** based on auditing criteria compiled from different standards
P12 – [11]	**OSS component reuse** without assurances	**Component-based certification of OSS** to increase confidence. **OpenSME process**
P13 – [8]	**Inputs for certifying for OSS** vs closed source	**Stakeholder-driven scenarios** for the certification process
P14 – [14]	**Who undertakes certification-related activities** in OSS communities?	**Making certification activities part of educational programs** in higher education
P15 – [9]	**Trustworthiness or security** of combined OSS and closed source components	**Open model approach** to guarantee secure code, security analysis in cases of closed source components, partitioning and hardware virtualization
P16 – [10]	**Impacting factors for confidence** in OSS vs. closed source systems	Evidence collected in **virtual certification repository** to be used by the Certification Body for assessment
P17 – [5]	**What evidence to produce** for product-focused certification of OSS?	**Anti-patterns** collected, formalized, and ontologically related by OSS community.

4 Discussion

Our scoping review indicates a comparably low level of research activity on the specific issues related to the certification of OSS systems and components. The lack of research activity in this area could be due to the fact that OSS have only recently been considered for critical system domains and high assurance applications, i.e., during the last decade. Another possible explanation may be that researchers may view certification issues in closed source systems as substantially similar to certification issues in OSS. Our scoping review has indicated, however, that this assumption may not hold in general and that there are indeed different issues to consider when certifying OSS systems.

All reviewed studies agree that OSS certification efforts should focus on product-based assurances, as process-based assurances may be impractical in loosely controlled, "bazaar-style" development communities. This shift away from process-based certification may not be detrimental, as process-based assurances have limited power in predicting product-quality [18].

Interestingly, the open, community-style nature of typical OSS projects may, in fact, make product-focused assurances more economically feasible, by applying a crowd-sourcing paradigm to assurance and certification. This hypothesis can be seen as an implication of "Linus' Law", as stated by Raymond [20]: "Given enough eyeballs, all bugs are shallow". While significant controversy still prevails about the general validity of this law, Khoroshilov points out ways to organize the community to provide certification services [14]. More empirical and theoretical research is needed on its applicability in context of OSS certification. Important research questions include "How do certification concerns shape and impact OSS communities?" and "How to organize open source communities for effective and economic certification? "

The availability of source code for OSS components provides opportunities for scrutiny by third party certification bodies. However, the complexity, size and evolving nature of many OSS projects severely limit the practicality of such efforts, unless the software is developed "with certification in mind". Cotroneo's pre-certification kit [7], Comar et al.'s Open-DO continuous certification process [6], Fusani and Marchetti's virtual certification repository [10], Kakarontzas et al.'s OPEN-SME reuse process [11] are examples for approaches to develop "for certification". Some of these proposals can be considered complimentary, others are alternatives. Little empirical evidence is available to-date about their effectiveness in practice. As an increasing number of OSS systems are subject to certification and may consider these proposals, the community will need more empirical evidence on their effectiveness.

5 Limitations

A possible bias in results of this scoping review is in the selection of relevant papers. We mitigated this threat to validity by using clearly predefined inclusion/exclusion criteria and having a second reviewer checking the selection performed by the first author on a random sample of the query results.

Another threat to validity is in the selection of data sources. The choice to perform electronic search could potentially exclude publications that might have been relevant to include, and therefore pose a risk of external validity. However, we believe that this risk is minimal as OSS is a relatively new research topic and the fact that it belongs to the research domain of software engineering safeguards that there is not much that is not indexed electronically.

However, there are certain peculiarities when it comes to electronic search that could alter the results, although not substantially. For example, during the course of the study, SpringerLink was migrating from an older to a newer website. Since the older website offered more comprehensive and robust search options, and was still available, we decided to continue using it as our source. However, before the study was finished the publisher disabled access to the older website, and we could not use two of our keyword combinations. We feel confident that the search would not produce additional results, because we were at a point in our study where the last keyword combinations did not produce any additional results in any of the other digital libraries either. Nevertheless, we are acknowledging this as a potential limitation to our study.

A final concern is whether there is any potential threat to reliability. We expect that replications of our study would offer results similar to ours. This, of course, depends also on the comprehensiveness of the research questions used, especially the synthesis and discussion. The approach followed by other researchers in discussing and interpreting the results may bring different insights, but we believe that the underlying findings and trends identified would remain the same.

6 Conclusion

In this paper we have presented a scoping review on certification of OSS systems. By following a systematic approach in searching existing literature, we identified 125 papers. After screening for inclusion and exclusion criteria, our analysis yielded 17 primary studies (inclusion rate 13.6%). We gave a summary for each of the primary studies, and summarized the issues of certification for OSS that have been studied in the primary studies. The issues included OSS certification compared to closed source certification, challenges due to certification economics, development processes, OSS in development of inherently complex application domains, as well as certification of OSS security.

References

1. Bunyakiati, P., & Finkelstein, A. (2009). The compliance testing of software tools with respect to the uml standards specification - the argouml case study. In Workshop on Automation of Software Test, 2009, pp. 138–143.
2. Austin, A., Smith, B., & Williams, L. (2010). Towards Improved Security Criteria for Certification of Electronic Health Record Systems. In Workshop on Software Engineering in Health Care (pp. 68–73). New York, NY, USA: ACM.
3. Bergquist, M., & Ljungberg, J. (2001). The power of gifts: organizing social relationships in open source communities. Inf. Systems Journal, 11(4), 305–320.
4. Bertrand, C., & Fuhrman, C. P. (2008). Towards Defining Software Development Processes in DO-178B With Openup. In Canadian Conference on Electrical and Computer Engineering, 2008 (pp. 851–854).
5. Cerone, A., & Settas, D. (2011). Using antipatterns to improve the quality of FLOSS development. Electronic Communications of the EASST, 48, 16 pages.
6. Comar, C., Gasperoni, F., & Ruiz, J. F. (2009). Open-Do: An Open-Source Initiative for the Development of Safety-Critical Software. In 4th IET Intl. Conference on Systems Safety 2009. (pp. 1–5).
7. Cotroneo, D., Di Leo, D., Silva, N., & Barbosa, R. (2011). The PreCertification Kit for Operating Systems in Safety Domains. In Workshop on Software Certification (WoSoCER) (pp. 19–24).
8. Fabbrini, F., Fusani, M. & Marchetti, E. (2011). Process Scenarios in Open Source Software Certification. Electronic Communicat. of EASST, 48, 15 pages.
9. Feuser, J., & Peleska, J. (2010). Security in Open Model Software with Hardware Virtualization: The Railway Control System Perspective. Electronic Communications of the EASST, 33, 14 pages.
10. Fusani, M., & Marchetti, E. (2010). Damages and Benefits of Certification: A perspective from an Independent Assessment Body. Electronic Communications of the EASST, 33, 3 pages.
11. Kakarontzas, G., Katsaros, P., & Stamelos, I. (2010). Component Certification as a Prerequisite for Widespread OSS Reuse. Electronic Communications of the EASST, 33, 20 pages.
12. Helms, E., & Williams, L. (2011). Evaluating Access Control of Open Source Electronic Health Record Systems. In Proc. of the 3rd Workshop on Software Engineering in Health Care (pp. 63–70). New York, NY, USA: ACM.
13. Hippel, E. von, & Krogh, G. von. (2003). Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. Organization Science, 14(2), 209–223.
14. Khoroshilov, A. (2009). Open Source Certification and Educational Process. Electronic Communications of the EASST, 20, 8 pages.
15. King, J. T., Smith, B., & Williams, L. (2012). Modifying Without a Trace: General Audit Guidelines are Inadequate for Open-Source Electronic Health Record Audit Mechanisms. In Int. Health Inform. Symp. (pp. 305–314). ACM.

16. Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosen berg, J. (2002). Preliminary guidelines for empirical research in software engineering. IEEE Trans. on Softw. Eng., 28(8), 721–734.
17. Kitchenham, B., et al. (2009). Systematic literature reviews in software engineering - A systematic literature review. Inf. Softw. Techn., 51(1), 7–15.
18. Maibaum, T., & Wassyng, A. (2008). A Product-Focused Approach to Software Certification. Computer, 41(2), 91–93.
19. Morasca, S., Taibi, D., & Tosi, D. (2009). Towards Certifying the Testing Process of Open-Source Software: New Challenges or Old Methodologies? In Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (pp. 25–30). IEEE.
20. Raymond, E. S. (1999). Cathedral and the bazaar. La Vergne, TN: [SnowBall Publishing].
21. Rumrill, P. D., Fitzgerald, S. M., & Merchant, W. R. (2010). Using scoping literature reviews as a means of understanding and interpreting existing literature. Work (Reading, Mass.), 35(3), 399–404.
22. Samoladas, I., Gousios, G., Spinellis, D., & Stamelos, I. (2008). The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation. In Open Source Devel., Communities and Quality (Vol. 275, pp. 237–248). Springer US.
23. Sethi, R., Azzi, D., & Khusainov, R. (2011). Interoperability and Standardisation in Community Telecare: A Review. In IET Seminar on Assisted Living 2011 (pp. 1–6).
24. Smith, B., et al. (2010). Challenges for Protecting the Privacy of Health Information: Required Certification Can Leave Common Vulnerabilities Undetected. In Security & Privacy in Medical & Homecare systems (pp. 1–12).
25. Van der Leest, S. H. (2010). ARINC 653 Hypervisor. In IEEE/AIAA 29th Digital Avionics Systems Conference (DASC) (pp. 5.E.2–1–5.E.2–20).
26. Wassyng, A., Maibaum, T., & Lawford, M. (2010). On Software Certification: We Need Product-Focused Approaches. In C. Choppy & O. Sokolsky (Eds.), Foundations of Computer Software. Future Trends and Techniques for Development (Vol. 6028, pp. 250–274). Springer
27. West, J., & O'Mahony, S. (2008). The Role of Participation Architecture in Growing Sponsored Open Source Communities. Ind. & Innov., 15(2), 145–168.
28. Zhao, L., & Elbaum, S. (2003). Quality assurance under the open source development model. Journal of Systems and Software, 66(1), 65 – 75.