

Aligning Software Certification Skill Based on Industrial Issues in Software Engineering Curriculum

Jamaiah H. Yahaya, Aziz Deraman, Fauziah Baharom, and Abdul Razak Hamdan

Abstract—Currently, software engineering relies and focuses on issues relatively with well-established software development approaches and software process improvement. There is lack of skill that guides students with the knowledge of developing quality software products that meet certain standard in software industry. As a result, software products were being produced and delivered with bugs and complaints were reported that software quality degraded gradually. Previous study indicated that code analysis and testing software alone could not guarantee the quality of the product. Apart from the increase of software complexity, not performing the best software engineering practices was another major cause of software failure. In order to produce good quality software products, the practices need to be highlighted from technical aspects and non-technical aspects such as people, environment and project constraints. To meet the challenges in quality software product, current education in software engineering must aligned with the software quality and certification models. This paper outlines the rationale and method for designing a new quality and certification skill in software engineering curriculum together with the challenges in meeting the needs from industries through alignment of the new curriculum. It presents the new topics of user-centred software certification processes, development of user and management skills in certification and continuous improvement in certification which might be included and needed in a software engineering curriculum.

Index Terms—software quality, software certification, software engineering curriculum, user-centered software certification

I. INTRODUCTION

In many countries, software certification is considered as a new concept in software engineering topics and among software industries.

Manuscript received March 6, 2012; revised April 14, 2012. This work was supported in part by the Malaysia Ministry of Science, Technology and Innovation under Science Fund Grant (01-01-02-SF0353)

Jamaiah H. Yahaya is with the School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia (phone: 60389216181; fax: 60389256732; e-mail: jhy@ftsm.ukm.my).

Aziz Deraman is with the School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia (e-mail: ad@ftsm.ukm.my).

Fauziah Baharom is with the School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok 06010, Kedah, Malaysia (e-mail: fauziah@uum.edu.my).

Abdul Razak Hamdan is with the School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor Malaysia (e-mail: arh@ftsm.ukm.my).

Even though it is a new concept it is becoming popular nowadays where several researches have been carried out in this area and been applied in real environment. In general we can see that industries have realised the significant and importance of software certification in gaining competitive in global businesses today [5].

The current syllabus in software engineering focuses on issues relatively with well-established software development approaches and software process improvement. There is lack of skill and knowledge for the students in producing good quality software products. Thus, software was delivered with bugs that need to be corrected and fixed after being installed at user's site. This also describes the dissatisfaction of users toward software products [10][11][12]. Complaints on the quality issues of software products are being reported from time to time.

II. IMPORTANT OF SOFTWARE QUALITY AND CERTIFICATION ISSUES

In general, software engineering focuses mainly on approaches of software development. The term certification can be defined as the process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. A software certification is defined by Jeffry Voas as a fact sheet that spells out known software output behaviours (and it could also spell out known internal behaviours). It also spells out what conditions those behaviours can manifest themselves [13]. Stanfford and Wallnau [14] define certification as a process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. Certification is a means for improving the discipline by promoting the practical implementation of standards, the awareness of a body of knowledge, the recognition of a code of ethics, and the need for professional development [15].

If we look at different scenario such as medical and drug, there is an approved source to endorse drugs and medicine available in the market. Even though a consumer or patient may not be able to assess accurately whether a particular drug is safe, but they can be reasonably confident that drugs obtained from approved sources have an endorsement of the U.S Food and Drug Administration (FDA) which confers important safely information.

With the development of software certification users might be able to choose the correct software that meets their requirements even though the users do not understand the processes and program underlying the complete software product. A few countries have started to develop the software product certification program which involves third

party certification body. Korea is one of the countries that emphasis on certifying software to ensure the quality of the software in Korea industry. The quality certification program is called Good Software [9]. Malaysia is in the phase of developing the Malaysian Certification Program with Department of Standard Malaysia and Malaysian Software Testing Board together with Technical Working Group which members are from selected professionals and academicians from local agencies and industries.

III. THE INDUSTRIAL EXPERIENCES

From previous discussion and current skills in software engineering topics, we realise that there is lacking of mechanism and knowledge among practitioners and students regarding continuous ensuring quality of software product before delivery as well as during the operational. Our research team has developed two models of certification which by mean of SCM-prod and SPAC. SCM-Prod is a software certification model by product quality approach and SPAC is a model based on development process approach. Software certification can be viewed in three perspectives, product, process and personnel, but the combination of these approaches will produce a balance in software certification model [3].

A. SCM-PROD Model

The first model of certification is SCM-Prod model which a certification model based on end product quality approach (see Fig. 1).

The software certification model based on end product quality approach developed in this research consists of pragmatic quality factor (PQF), assessment team, weighted scoring method (WSM), decision process, repository and certification representation method. PQF is the quality assessment guidelines that consist of software quality attributes, metrics and measures. Undertaking quality attributes defined in ISO9126 model as the based line of the assessment metrics, we define two sets of attributes, which by means of the behavioural and the impact attributes. The behavioural attributes consist of high level software quality characteristics, which include usability, functionality, maintainability, portability, integrity, efficiency and reliability. In addition, previous study showed that quality attributes can be classified into different levels and weight [6]. The impact attributes indicate the conformance in user requirements, expectation and perception. These two groups of attributes are important to balance the assessment between the technical aspects of quality and human factors.

Other interesting aspects of this model are the assessment team that involve in the assessment exercise and the certification representation method. This model focuses on user-centred certification where users are able to do the assessment and certification by themselves (self-certifying) and secondly the quality measurements used in this model focus more for user involvement and satisfaction. The requirements today's demanded the quality model to be simple, specific and practical to be measured by layman, users, customers, developers or stakeholders. This relates back with the general definition of quality, quality is defined as "fitness for use" and "conformance to requirements". The term "fitness of use" usually means characteristics such as functionality, usability, maintainability, and reusability and "conformance to requirements" means that software has

value to the users [16][21]. The previous developed and implemented certification models did not accommodate this requirement.

The certification representation method explains how the certification can be implemented and consists of algorithms and methods for certification. In this method there are two main certification approaches. First approach is to assess and certify based on individual attributes defined in PQF while the second approach is to assess and certify software as one product. The detail of this model can also be referred in [7][16].

B. SPAC Model

The second certification model developed by our research group is called SPAC Model – *Software Process Assessment and Certification Model*. The primary goal of this model is to ensure that the software development process are carried out effectively and efficiently to meet the expected quality criteria, delivered on time and within budget. This model is formulated based on the existing models, which are Capability Maturity Model (CMM) [17], ISO 9000, ISO/IEC 15504 [18] and Bootstrap [19]. It focuses mainly on five key factors that influenced the quality of software.

The factors are the quality of process performed the quality of people involved, the use of development technology, the stability of working environment and project conditions. SPAC consists of seven components, which can elaborated as the candidate software, the process quality factor, the certification and quality index, assessment team and repository. This model is demonstrated as in Fig. 2.

The following discusses the components of SPAC:-

1. The first component of SPAC is the Process Quality Factor (PSQF). It defines what to be measured in this model. PSQF identifies factors that affect the quality of software process in practice. The five factors are: process, people, environment, development technology and project constraint.
 - Process: The factor of process includes three basis activities, which are development, management and support activities.
 - People: This factor measures in term of skill, experience, knowledge, team commitment, user involvement and management responsibility.
 - Environment: This factor measures the comfort ability and safety aspects in the work place.
 - Development technology: This factor measures in term of standard and procedure, tools, methods and techniques and process origin.
 - Project constraint: This aspect of quality measures the time delivery and budget.
2. The second component is the candidate software to be assessed. This candidate is a completed product that is ready to be delivered to users or customers. Information on the development process is collected via multiple techniques: reviewing all artefacts produced during development process, interviewing key personnel and also observing the working environment.
3. The assessment team is the third component of this model. In this model assessment is carried out by a collaborative approach which consists of a group of people. Developers, independent assessor and project manager should be a part of the assessment team and the

team's leader must be an expert in software engineering and software quality.

4. The fourth component is the assessment and certification process. This component contains three main phases of implementation: preparation, execution and post assessment phase. These phases are then decomposed into 16 activities that provide guidance to facilitate the whole process of certification.
5. The fifth and sixth components are the quality and certification level.
6. The seventh component is the repository, which stores all information and results from assessment and certification exercises. This data is useful for future analysis and improvement.

Further discussion on this model can be found in different documents published by our research group [20].

C. Other Models

Many approaches to software certification mostly rely on formal verification, expert reviews, developer assessment and software metrics to determine the product quality as described in Welzel and Hausen [22], Voas [23], Lee, Ghandi & Wagle [24] and Heck, Klabbers & Eekelen [25]. Another approach is by integrating ISO9126 model as the certification quality benchmark such as Good Software [9], Requirement-driven Workbench [24] and SCM-Prod [7]. These models are appropriate for general software assessment with static attributes such as portability, usability, reliability, maintainability, functionality and efficiency. Different approach for certification is using function point [26]. Function point is a standard metric for the relative size and complexity of a software system, originally developed by the IBM in the late 1970s. Most of the studies mentioned above focused on certify software artifacts from developers, suppliers and auditors perspectives and do not emphasize much on user's perspective and involvement.

IV. ALIGNING WITH INDUSTRY

Fig. 3 presents an overview of the processes in software certification environment. This figure shows that software certification can be implemented in two ways:-

1. Certification of new products. This is the certification of a new software product that just completed in development and implementation and readied to be sold in the market.
2. Certification of product in-used. This is the certification of software product that is already in-used in certain environment.

As illustrated in Fig. 3, the basic processes of software certification for new product and product in-used have supporting processes that are concerned with certification management, certification model and certification system:

1. Certification management is concerned with managing a company's certification exercises of software products available in the company. It may involve accessing newly developed software or software already operating in the environment. The management process can be done by the third party agency or the independent SQA team in the company. Certification exercises conducted by the certification management may be stored in a certification repository that includes both the software

and information about their certification level and quality status.

2. Certification model is concerned with the standard, procedures and mechanisms for certifying software product. It may involve applying software product certification model (example is SCM-prod) or software development process certification (example is SPAC).
3. Certification system is the support tool and guidelines for assessment and certification. The certification system and the assessment can be conducted by the external independent certifier or self-certifier.

V. DISCUSSION

This paper has presented and discussed the models for certifying software products. The two certification models namely SCM-prod and SPAC model have been tested and applied in real environments collaboratively with industry in Malaysia. The models have been developed in a goal-directed way in order to meet the needs of the different interest groups associated with software quality. The other related papers from our research group discuss and explain our experiences in applying software certification in real industries environment in more detail [4][8]. Thus, with the models discussed in this paper, certification exercise can be done in two approaches or perspectives which are via end product and development process.

Our experience with users from various organisations and sectors indicates that software certification approach which is a higher level of quality assessment is beneficial to ensure quality of software. The certification approach can be applied at any time during the operational of the software, thus the continuous quality monitoring will be guaranteed [1]. In addition, results from certification will provide a valuable recognition on the quality of the software organization which can support the buoyancy and trustworthiness of the organization. Fig. 3 demonstrates the processes involved in software certification that relates to the industry. Currently, the syllabus of software engineering does not integrate with these requirements. Therefore, it is important to deliver this skill and knowledge to the students to ensure that our students have enough knowledge in producing and managing good quality software in the real industry.

Previous study indicated that code analysis and testing software alone would not guarantee the quality of the product [5]. Apart from the increase of software complexity, not performing the best software engineering practices was another major cause of software failure [2]. In order to produce good quality products, the practices need to be highlighted from technical aspects and non-technical aspects such as people, environment and project constraints. To meet the challenges in quality software product, current education in software engineering must align with the software quality and certification models.

Further analysis and study need to be carried out to investigate the correlation between the quality of the process and quality of the product. Furthermore, in our framework we focus on continuous improvement in two facets. Firstly is the continuous improvement of the software product itself. This model is enable easy assessment and certification exercises and offers better guidance and procedures. Having the model and intelligent toolset to support the certification

process allow users to evaluate and assess the software continuously, thus facilitate the continuous improvement of the software. Secondly, it provides continuous improvement of the quality model applied in the certification process.

VI. CONCLUSION

The two software certification models discussed in this paper have been applied in the real case studies collaborating with industry in Malaysia. The application has demonstrated the practicality and feasibility of the proposed processes and mechanisms. Later, the evaluation of the model by the software product's owner verified the integrity of the models. It is worth pointing out that the models are definitive, as the results from the application and evaluation process have confirmed and verified the models. This paper outlines the rationale and method for designing the new quality and certification skill in software engineering curriculum together with the challenges in meeting the needs from industries through alignment of the new curriculum. It presents the new topics of user-centred software certification processes, development of user and management skills in certification and continuous improvement in certification which might be included and needed in a software engineering curriculum.

REFERENCES

- [1] A. Deraman, J.H. Yahaya, F. Baharom, A.F.A. Fadzliah and A.R. Hamdan, "Continuous quality improvement in software certification environment," In *Proc. The International Conference on Electrical Engineering and Informatics (ICEEI 2007)*, Bandung, pp.11-17, 2007.
- [2] F. Baharom, A. Deraman and A.R. Hamdan, "A Survey on the current practices of software development process in Malaysia," *Journal of Information and Communication Technology (JICT)*, vol. 4, pp. 57-76, 2005.
- [3] J. Voas, "The Software Quality Triangle: Crosstalk," *The Journal of Defense Software Engineering*, vol. 11, no. 11, pp.12-14, 1998.
- [4] F. Baharom, J.H. Yahaya, A. Deraman and A.R. Hamdan, "SPQF: Software process quality factor for software process assessment and certification," in *Proc. The International Conference on Electrical Engineering and Informatics*, Bandung, Indonesia, 17-19 July 2011.
- [5] J.H. Yahaya, A. Deraman and A.R. Hamdan, "Software quality and certification: Perception and practices in Malaysia," *Journal of ICT (JICT)*, vol. 5, pp. 63-82, Dec 2006.
- [6] J.H. Yahaya, A. Deraman and A.R. Hamdan, "Software product certification model: Classification of quality attributes," in *Proc. The First Regional Conference of Computational Science and Technology (RCCST 07)*, Kota Kinabalu, pp. 436-440, 2007.
- [7] J.H. Yahaya, A. Deraman and A.R. Hamdan, "Software certification model based on product quality approach," *Journal of Sustainability Science and Management*, vol. 3, no. 2, pp. 14-29, Dec. 2008.
- [8] J.H. Yahaya, A. Deraman and A.R. Hamdan, "Continuously ensuring quality through software product certification: A case study," in *Proc. The International Conference on Information Society (i-Society 2010)*, London, UK, pp.193-198, 2010.
- [9] IT Times. (2011) 30 Korean Software Worthy of Global Recognition in 2012. [Online]. Available: <http://www.koreaitimes.com/story/15607/30-korean-software-worthy-global-recognition-2012>
- [10] Eagles (1995) Evaluation of natural language processing systems, Final Report (Sept). [Online]. Available: <http://www.issco.unige.ch/ewg95/>
- [11] P.J. Denning, "What is Software Quality? ", *A Commentary from Communications of ACM*, January, 1992
- [12] J.A. Whittaker and J. Voas, "50 years of software: Key principles for quality," *IEEE IT Pro*, pp. 28-35, Nov/Dec, 2002.
- [13] J. Voas, "User participation-based software certification," in *Proc. of Eurovav 1999*, pp. 267-276, 1999.
- [14] J. Stanford and K. Wallnau, "Is Third Party Certification Necessary?" in *Proc. 4th ICSE Workshop on Component-Based Software Engineering: Component Certification and System Prediction*, 2001.
- [15] L.L. Tripp, "Software Certification Debate: Benefits of Certification," *IEEE Software*, pp. 31- 33, June, 2002.
- [16] A. Deraman, J.H. Yahaya, F. Baharom and A.R. Hamdan, "User-Centred Software Product Certification: Theory and Practices," *International Journal of Digital Society (IJDS)*, vol. 1, no. 4, pp. 281-288, Dec. 2010.
- [17] M.C. Paulk, B. Curtis, M.B. Chrissis and C.V. Weber, *The Capability Maturity Model for Software*. IEEE Computer Society Press, pp. 427-438, 1997.
- [18] G. O'Regan, *A Practical Approach to Software Quality*. Springer, 2002.
- [19] P. Kuvaja, "BOOTSTRAP 3.0 - A SPICE Conformant Software Process Assessment Methodology," *Software Quality Journal*, vol. 8, pp. 7-19, 1999.
- [20] F. Baharom, A. Deraman, A.R. Hamdan. and J.H. Yahaya, "Software Certification towards Assuring and Improving the Quality of Software Development Process," In *Proc. The Malaysian Software Engineering Conference (mySec'08)*, Kuala Terengganu, 16-17 Dec, 2008.
- [21] A. Deraman, "Software Certification: The Way Forward (Keynote)," In *Proc. The 5th Malaysian Software Engineering Conference (MySec2011)*, Johor Bharu, 13-14 Dec 2011.
- [22] D. Welzel and H. Hausen, " Practical concurrent software evaluation for certification," *Journal System Software*, vol. 38, pp. 71-83, 1997.
- [23] J. Voas, "Certifying software for high assurance environments," *IEEE Software*, pp. 22-25, July/Aug 1999.
- [24] S.W. Lee, R.A. Ghandi and S. Wagle, (2007). "Towards a requirements-driven workbench for supporting software certification and accreditation," in *Proc. Software engineering for secure systems 2007 SESS'07:ICSE Workshops 2007. Third International Workshop. IEEE*, [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4273334>
- [25] P. Heck, M. Klabbbers and M. Eekelen, " A Software Product Certification Model," *Software Quality Journal* vol. 18, pp. 37-55, 2010.
- [26] IFPUG (2008). International Function Points Users Group, [Online]. Available: <http://www.ifpug.org/certification/software.htm>

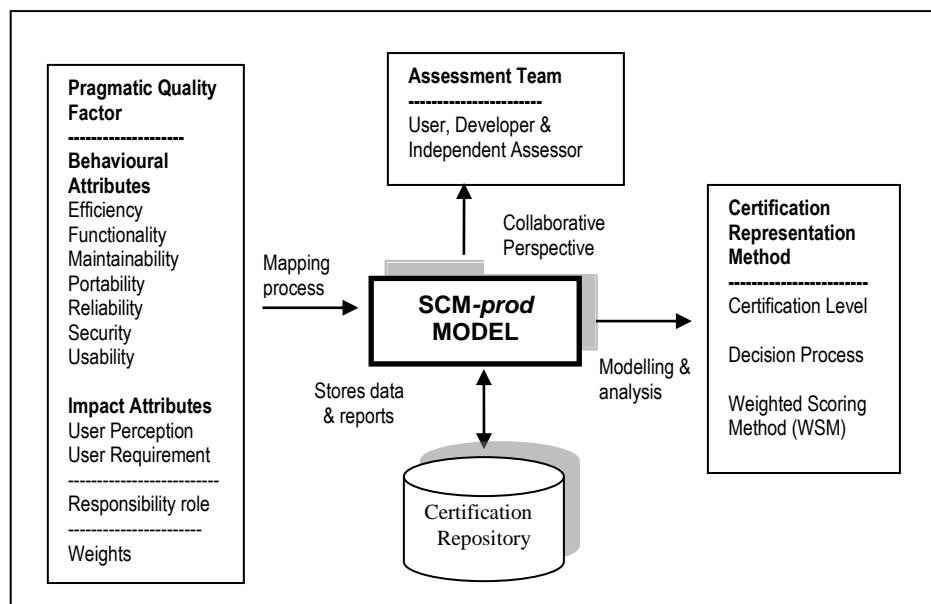


Fig. 1. SCM-Prod - User-centred Approach of Software Product Certification Model [7]

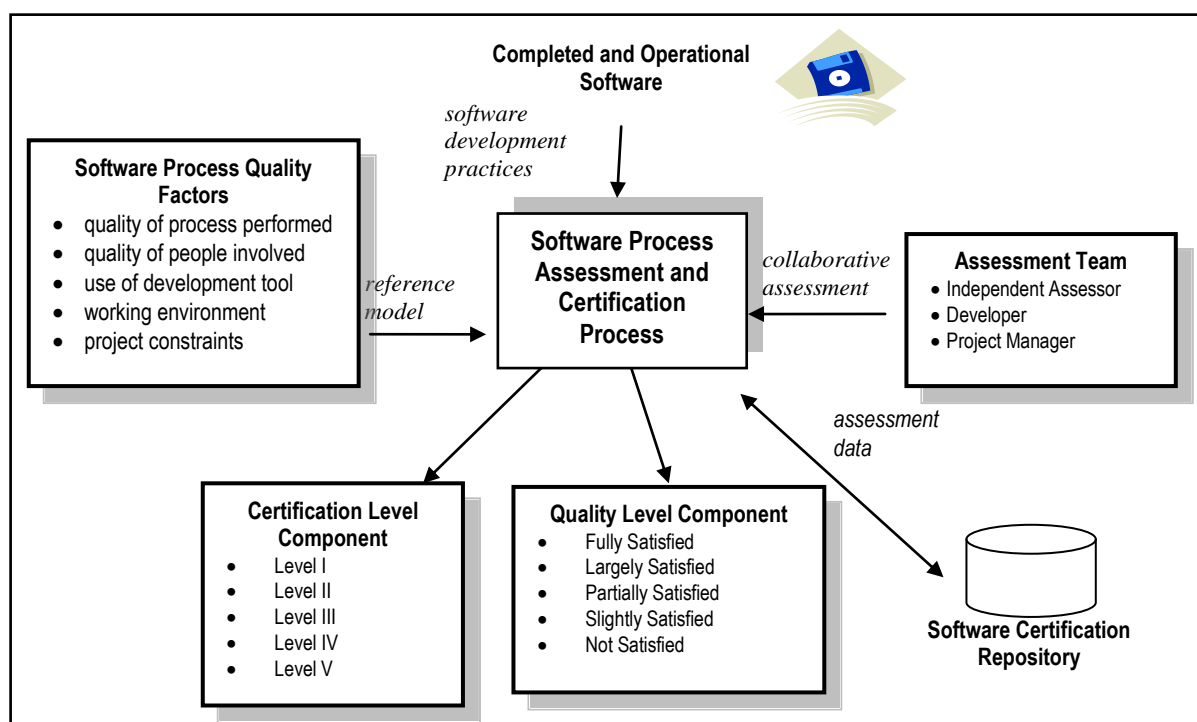


Fig. 2. Software Process Assessment and Certification (SPAC) Model [20]

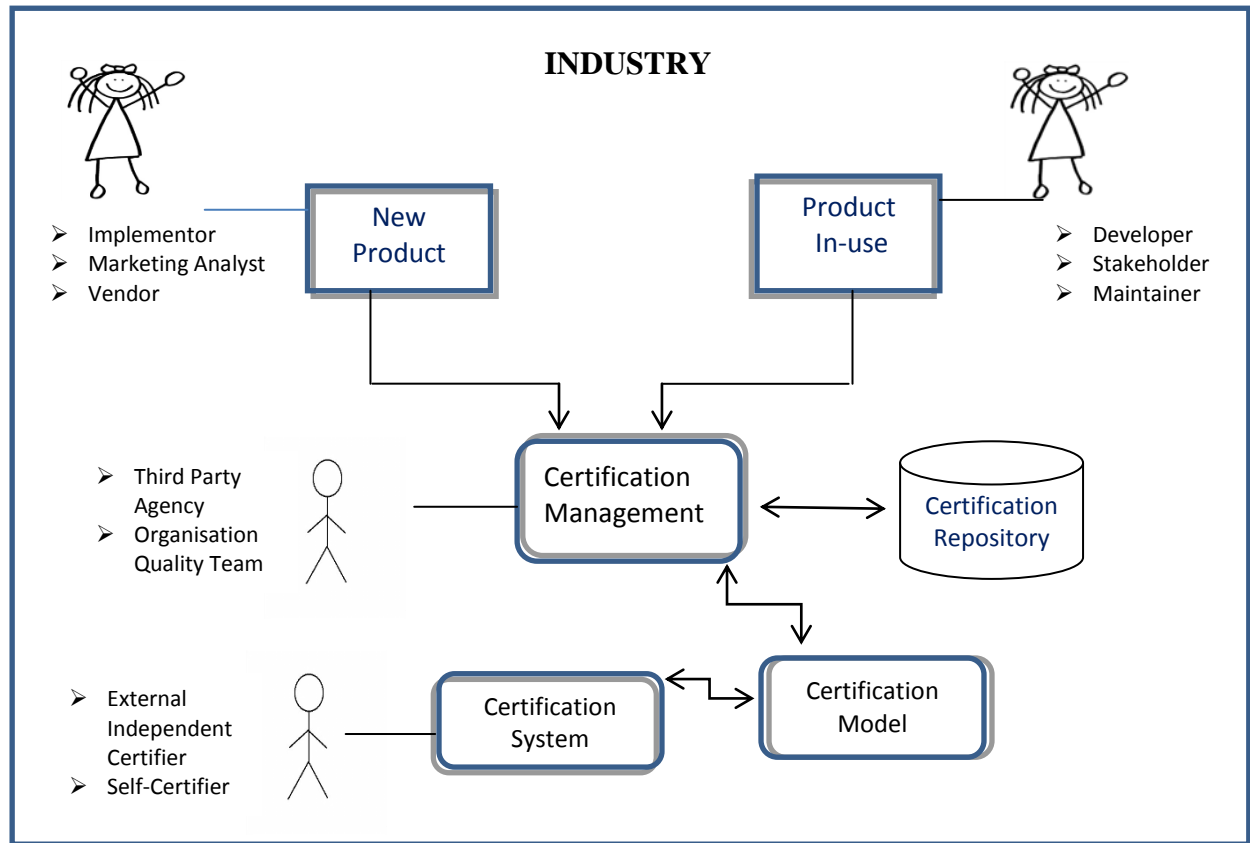


Fig.3. Software Certification Process in the Industry