



¹N.K. Uteliyeva , ²E.E. Ismail , ³D.F. Akhmetov

¹PhD student, Al-Farabi Kazakh National University,

¹Senior lecturer, Almaty University of Power Engineering and Telecommunications,
Almaty, Kazakhstan, E-mail: nurshatu7@mail.ru

²Candidate of Technical Sciences, Professor,
Almaty University of Power Engineering and Telecommunications,
Almaty, Kazakhstan, E-mail: ismaile@rambler.ru

³Doctor of Engineering, PhD, Rakuno Gakuen University, Hokkaido, Japan,
E-mail: sapalmaty@gmail.com

COMPARATIVE ANALYSIS OF MODELS OF QUALITY OF SOFTWARE TOOLS

Abstract. Although a large number of software quality models (SQMs) have been created at present, there is no universal model that can be applied to various software tools with the best results. To analyze and evaluate the quality of the assessment of a specific SQ, it is necessary to choose an adequate quality model that takes into account their features and requirements for them, and also most accurately describes the quality indicators at various stages of the life cycle. Each specific SQM is usually characterized by its own set of characteristics and quality attributes, has its own advantages and disadvantages. Based on these characteristics, models can be oriented to various fields of application.

In order to understand the appropriateness of using one or another SQM, one or another characteristics and quality attributes of SQ suitable for specific SQ projects, it is necessary to conduct a comprehensive analysis of the features of existing SQM.

The aim of this work is to conduct a comprehensive comparative analysis of the characteristics and features of modern SQMs to assess their capabilities and applicability, the ability to adapt to the features and requirements of the applied problem.

SQM analysis was carried out in the following aspects: structure, number of levels and characteristics and their semantic content, as well as in terms of identifying opportunities and applicability. As a result of a comparative analysis of the main characteristics and sub characteristics of basic SQMs, the most applicable (basic) set of characteristics and sub characteristics of software quality are identified. For application to specific types of software, this basic set can be adapted to the appropriate application conditions by selecting the relevant characteristics and sub characteristics, as well as possibly adding the necessary quality attributes from other models.

Key words: software, quality, safety, the characteristics of the quality, quality model, comparative analysis.

¹Утелиева Н.К., PhD докторант, әл-Фараби атындағы Қазақ ұлттық университеті,
аға оқытушы, Алматы энергетика және байланыс университеті,

Алматы қ., Қазақстан, E-mail: nurshatu7@mail.ru

²Исмаил Е.Е., т.ғ.к., профессор, Алматы энергетика және байланыс университеті,
Алматы қ., Қазақстан, E-mail: ismaile@rambler.ru

³Ахметов Д.Ф., Инженерия докторы, PhD, Rakuno Gakuen университеті,
Хоккайдо қ., Жапония, E-mail: sapalmaty@gmail.com

Бағдарламалық қамтамасыз ету сапасының модельдерін салыстырмалы талдау

Аңдатпа. Қазіргі уақытта бағдарламалық жасақтама сапасының көптеген үлгілері (БЖС) жасалғанына қарамастан, ең жақсы нәтижеге ие әртүрлі бағдарламалық құралдарға қолданылатын эмбебап модель жоқ. Белгілі бір БЖ-ді бағалаудың сапасын талдау және бағалау үшін олардың ерекшеліктері мен оларға қойылатын талаптарды ескеретін, сонымен қатар өмірлік циклдің әртүрлі кезеңдеріндегі сапа көрсеткіштерін неғұрлым дәл сипаттайтын сәйкес сапа моделін таңдау қажет.

Әрбір нақты БЖСМ, әдетте, өзіндік сипаттамалары мен сапалық атрибуттарымен сипатталады, өзінің артықшылықтары мен кемшіліктері бар. Осы сипаттамаларға сүйене отырып, модельдер қолданудың әртүрлі салаларына бағытталуы мүмкін.

Белгілі бір БЖ жобаларына сәйкес келетін бір немесе басқа БЖСМ, БЖ сипаттамалары мен сапалық атрибуттарын қолданудың орындылығын түсіну үшін қолданыстағы БЖСМ ерекшеліктеріне жан-жақты талдау жүргізу қажет.

Бұл жұмыстың мақсаты - заманауи БЖСМ-ның сипаттамалары мен ерекшеліктеріне олардың мүмкіндіктері мен қолданылуын, қолданылатын мәселенің ерекшеліктері мен талаптарына бейімделу қабілеттерін бағалау үшін жан-жақты салыстырмалы талдау жүргізу. БЖСМ талдауы келесі аспектілерде жүзеге асырылды: құрылымы, деңгейлері мен сипаттамалары (сипаттамалары бойынша) және олардың семантикалық мазмұны, сонымен қатар мүмкіндіктерді анықтау және қолдану мүмкіндігі.

Негізгі сипаттамаларды және негізгі БЖСМ сипаттамаларын салыстырмалы талдау нәтижесінде бағдарламалық қамтамасыздандырудың сапалық сипаттамаларына ең көп қолданылатын (негізгі) сипаттамалар анықталды. Бағдарламалық жасақтаманың белгілі бір түрлеріне қолдану үшін бұл негізгі жиынтықты нақты сипаттамаларды және сипаттамаларды таңдау арқылы қолданбалы бағдарламаның тиісті жағдайларына бейімдеуге болады, сонымен қатар басқа модельдерден қажетті сапалық атрибуттарды қосуға болады.

Түйін сөздер: бағдарламалық құрал, сапа, сапа сипаттамалары, сапа моделі, салыстырмалық талдау.

Сравнительный анализ моделей качества программных средств

¹Утелиева Н.К., PhD докторант, Казахский национальный университет имени аль-Фараби, старший преподаватель, Алматинский университет энергетики и связи, г. Алматы, Казахстан, E-mail: nurshatu7@mail.ru

²Исмаил Е.Е., к.т.н., профессор, Алматинский университет энергетики и связи, г. Алматы, Казахстан, E-mail: ismaile@rambler.ru

³Ахметов Д.Ф., доктор инженерии, PhD, Университет Rakuno Gakuen, г. Хоккакайдо, Япония, E-mail: sapalmaty@gmail.com

Аннотация. Хотя в настоящее время создано большое количество моделей качества программного обеспечения (МКПО), однако не существует универсальной модели, которая была бы применима для различных программных средств с наилучшими результатами. Для анализа и оценки качества оценки конкретного ПС необходимо выбрать адекватную модель качества, которая учитывала бы их особенности и требования к ним, а также наиболее точно описывала бы показатели качества на различных стадиях жизненного цикла.

Каждая конкретная МКПО обычно характеризуется своим набором характеристик и атрибутов качества, имеет свои достоинства и недостатки. Исходя из этих характеристик, модели могут быть ориентированы на различные сферы применения.

Для того чтобы понять целесообразность использования той или иной МКПО, тех или иных характеристик и атрибутов качества ПС, пригодных для конкретных проектов ПС, необходимо провести комплексный анализ особенностей существующих МКПО.

Целью настоящей работы является проведение комплексного сравнительного анализа характеристик и особенностей современных МКПО для оценки их возможностей и применимости, возможности адаптации к особенностям и требованиям прикладной задачи. Анализ МКПО проведен в следующих аспектах: структура, количество уровней и характеристик (под характеристик) и их смысловое содержание, а также в плане выявления возможностей и применимости. В результате сравнительного анализа основных характеристик и под характеристик базовых МКПО выявлены наиболее применимый (базовый) набор характеристик и под характеристик качества программных средств. Для применения к конкретным типам программных средств этот базовый набор может быть адаптирован к соответствующим условиям применения путем выбора актуальных характеристик и под характеристик, а также возможного добавления необходимых атрибутов качества из других моделей.

Ключевые слова: программное средство, качество, характеристики качества, модель качества, сравнительный анализ.

1 Introduction

The concept of software quality (SQ) is multidimensional and can be expressed adequately only by some structured system of characteristics or attributes, called a quality model. For almost half a century of the development of software engineering, dozens of different software quality models (SQM) have been proposed [1-10]. Characteristics, sub-characteristics and quality attributes of SQ, set in these models, became the basis for the formal description of quality characteristics and their quality assessment in specific projects.

Users and developers have needs for quality models, allowing a formation of the quality indicators structure depending on the specifics and the scope of a particular software product. The SQM may reflect, to varying degrees, the expectations of various categories of participants in a program project: managers, developers, support personnel, users, and must contain an ordered set of characteristics that ensure the consistency of their interests. In order to understand the feasibility of using a particular quality model, certain characteristics and quality characteristics of the SQ, it is necessary to conduct a comprehensive comparative analysis of the characteristics and features of the existing SQM. In a number of papers [11-15], similar studies were performed. However, the purposes of the SQM analysis in the mentioned papers were different, it was carried out mainly at the level of characteristics, and the sub-characteristics were not considered completely, or not studied at all. In some works, the modern standard ISO 25010, which describes the new SQM, adopted in 2010, has not been considered.

The purpose of this work is to conduct a comprehensive comparative analysis of the characteristics and features of modern SQM to assess their capabilities and applicability. The analysis is carried out in the following aspects: evolution, structure, characteristics (sub-characteristics) and their semantic content, as well as general development trends.

The SQM has a hierarchical structure. Its elements are the sets of characteristics (sub-characteristics, attributes) and the subordination relationships between them.

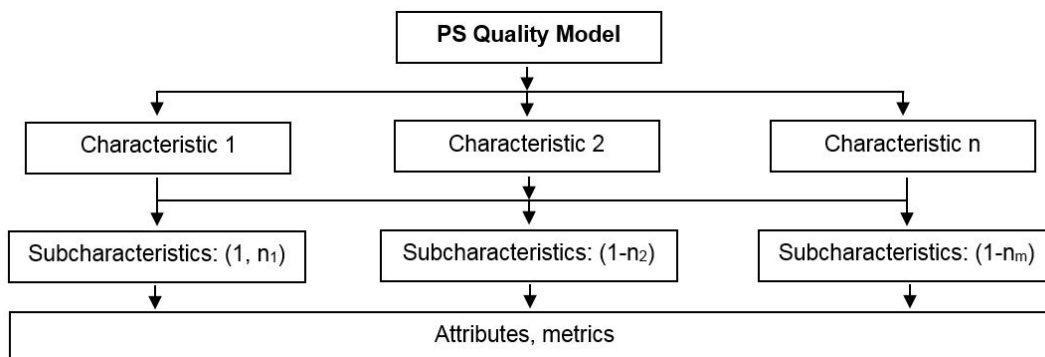


Figure 1 Structure of the PS quality model

2 Literature review

The most of present SQM are hierarchical models consisting of high-level indicators (characteristics) that are detailed by lower-level indicators (sub-characteristics, attributes, metrics)

until decomposition results in atomic and measurable quality elements. (Fig. 1). There are subordination relations between all elements of the SQM.

The structure of the SQM should reflect and bind the interests of the user, i.e. outgoing properties of the system, with internal properties that are understandable to developers.

Table 1 presents the results of the evolutionary analysis of the most well-known SQM with an indication of its main parameters, such as the number of hierarchy levels, the number of characteristics / sub-characteristics by levels / sublevels.

Table 1. – Key parameters of the most well-known SQM

No	Name of the SQM	Year of publication	Number of model levels	Number of characteristics / sub-characteristics	Developer	A source
1	McCall	1977	2	11/35	J. McCall and General Electrics	[1]
2	Boehm	1978	3	3/8/18	B.W. Boehm	[2]
3	FURPS/ FURPS+	1987/2000	2	5/25	R.B. Grady and Hewlett Packard	[3]
4	Ghezzi	1991	1	8	Carlo Ghezzi	[4]
5	ISO 9126	1991	2	6/21 + 4 (for quality in use)	ISO	[5]
6	IEEE	1993	2	6/19	IEEE (Institute of Electrical and Electronics Engineers)	[6]
7	Dromey	1995	2	4/13	G.R. Dromey	[7]
8	SATC NASA	1996	2	6/21	SATC (Software Assurance Technology Center NASA)	[8]
9	ISO/IEC 9126-1	2001	2	6/27 + 4 (for quality in use)	ISO/IEC	[9]
10	ISO/IEC 25010	2010	2	8/31 + 5/11(for quality in use)	ISO/IEC	[10]

An analysis of SQM-related studies [11-15] shows that the most significant SQM are: McCall, Boehm, FURPS, Dromy, IEEE, SATC, ISO / IEC 9126, ISO / IEC 25010.

Practically, all considered SQM are based on the formation of a hierarchical structure of quality characteristics. The differences lie in the proposed number of levels of hierarchy, as well as in the characteristics themselves of the upper level of the hierarchy, many of which still coincide. The expediency of the hierarchical structure of the SQM is explained, firstly, by the fact that the multi-level structure of quality indicators makes it possible the systematical description of the requirements for the software, allowing stakeholders to set the properties (characteristics) of the software product that they want to see.

In terms of the development trend of the MQST the following can be noted:

- the range of quality characteristics is expanding;
- the number of hierarchical levels of the model is at least two;
- the structure of the characteristics of quality models is complicated due to their greater detailing at the level of sub-characteristics.

The existing SQMs can be divided into the following categories:

- fundamental or basic models that are the result of the work of authors' teams of international authoritative organizations; respectively, this group includes the quality models IEEE, SATC, ISO 9126, ISO 9126-1 and ISO 25010;

- corporate quality models of PS, which were, as a rule, developed in the interests of certain developers or consumers, and are significantly inferior in terms of the nomenclature of characteristics and their relations to the basic software quality models. The following quality models can be attributed to this group of models: McCall, Boehme, Ghezzi, FURPS, Dromer.

Analysis of the SQM, taking into account their evolutionary development, shows that standardized basic quality models are being developed and improved in accordance with the modern achievements of the theory and practice of software engineering, the growing demand for high-quality software products in various fields. They also take into account the main advantages of corporate SQM. In this regard, in the future it seems appropriate to consider only the basic models: ISO 25010, ISO 9126-1, IEEE and SATC.

3 Material and methods

3.1 Main parameters of the SQM

In order to identify the most applicable characteristics and sub-characteristics of the PS in the above standardized INCD, their comparative analysis was carried out. The results of this analysis are presented in Table 2.

Table 2. – Comparison of the main characteristics and sub-characteristics of the basic SQM

No	PS Quality Characteristics	PS Quality Models			
		ISO 25010 model	ISO 9126-1 model	IEEE model	SATC model
A. PS Quality Characteristics					
1.	Functional suitability	+	+	+	±
2.	Performance efficiency	+	±	±	+
3.	Compatibility	+	±	+	±
4.	Usability	+	+	+	+
5.	Reliability	+	+	+	-
6.	Security	+	+	+	±
7.	Maintainability	+	+	+	+
8.	Portability	+	+	+	±
9.	Efficiency	+	+	+	±
10.	Functionality	+	+	+	±
11.	Supportability	±	±	+	+
12.	Requirements Quality	-	-	-	+
13.	Product (Code) Quality	±	±	±	+
14.	Design Quality	-	-	-	+
15.	Implementation Efficiency	±	±	±	+
16.	Testing Efficiency	±	±	±	+
17.	Installation & Checkout Quality	±	±	±	+
18.	Operation & Maintenance Quality	±	±	±	+
19.	Retirement Quality	-	-	-	+
B. Sub-characteristics of PS quality					
1.	Functional completeness	+	+	+	+
2.	Functional correctness	+	+	+	±
3.	Functional appropriateness	+	+	±	±
4.	Time behaviour	+	+	+	±
5.	Resource utilization	+	+	+	+
6.	Capacity	+	±	±	±
7.	Co-existence	+	+	+	±
8.	Interoperability	+	+	+	±
9.	Recognizability	+	-	-	±

10.	Learnability	+	+	+	+
11.	Operability	+	+	+	±
12.	User error protection	+	-	-	-
13.	User interface aesthetics	+	-	-	+
14.	Accessibility	+	-	-	-
15.	Maturity	+	+	-	-
16.	Fault tolerance	+	+	+	-
17.	Availability	+	-	+	-
18.	Recoverability	+	+	-	-
19.	Confidentiality	+	-	-	-
20.	Integrity	+	-	-	+
21.	Non-repudiation)	+	-	-	-
22.	Accountability	+	-	-	-
23.	Authenticity	+	-	-	-
24.	Modularity	+	-	-	+
25.	Reusability	+	-	+	+
26.	Analysisability	+	+	-	-
27.	Modifiability	+	+	+	±
28.	Testability	+	+	+	+
29.	Adaptivity	+	+	+	±
30.	Installability	+	+	+	+
31.	Replaceability	+	+	±	+
32.	Accuracy	+	+	+	+
33.	Compliance	-	+	-	±
34.	Understandability	±	+	+	+
35.	Volatility	-	-	-	+
36.	Attractiveness	+	+	-	-
37.	Stability	-	+	-	-
38.	Fault-proof	-	-	+	-
39.	Expandability	±	±	+	+
40.	Ambiguity	-	-	-	+
41.	Traceability	-	-	-	+
42.	Complexity	-	-	-	+
43.	Sizing	-	-	-	+

Notes: the sign "+" means applicability and full equivalence of the relevant characteristics (sub-characteristics) in the considered PS quality models, the sign "-" means the absence (inapplicability) of the specified characteristics (sub-characteristics) in the considered quality models, the sign "±" means applicability, but incomplete equivalence of the relevant characteristics (sub-characteristics) in the considered PS quality models.

4 Results and discussions

4.1 Comparative analysis of the characteristics of basic SQM

The results of a comparative analysis of the main characteristics and sub-characteristics of basic SQMs revealed the following.

a) In all four analyzed software quality base models, the following quality characteristics are used:

- Functional suitability;
- Reliability;
- Usability;

-
- Security;
 - Maintainability;
 - Portability;
 - Efficiency.

b) Also, in fact, in one form or another, the following quality sub-characteristics are used in all four analyzed basic software quality models:

- Functional completeness;
- Functional correctness;
- Functional appropriateness;
- Time behavior;
- Co-existence;
- Interoperability;
- Learnability;
- Operability;
- Reusability;
- Modifiability;
- Testability;
- Installability;
- Accuracy;
- Understandability.

c) The following quality characteristics and sub-characteristics are applied only in one of the four analyzed basic software quality models:

- Requirements Quality;
- Product (Code) Quality;
- Design Quality;
- Implementation Effectivity;
- Testing Effectivity;
- Installation & Checkout Quality;

- Operation & Maintenance Quality;
- Retirement Quality.
- Compliance;
- Requirements volatility;
- Stability;
- Faultlessness;
- Ambiguity;
- Traceability;
- Complexity;
- Sizing;
- Independence from software platform;
- Independence from hardware platform.

Based on the results of a comparative analysis of the characteristics of various SQMs, it can be noted that the ISO / IEC 25010 model, which is the most modern, perfect and universal, can be adopted as the basic model. In this case, consideration of the rest of the SQM may be useful in the aspect of identifying relevant characteristics applicable to certain types of software. To apply to specific types of software, the basic model must be adapted to the relevant conditions of use by selecting relevant characteristics and sub-characteristics, as well as the possible addition of the necessary quality attributes from other SQMs.

5 Conclusion

Summarizing the results of comparing existing models of PS quality, it can be noted that the problem of describing and evaluating SQ quality remains completely unresolved due to the large variety of PS and the narrow specificity of their individual classes. A comparative analysis of the existing SQM revealed a number of their shortcomings, the main of which can be formulated as follows:

- lack of terminological consistency, which leads to ambiguity in the interpretation of similar terms and the use of different terms to describe equivalent concepts;
- existing models are either abstract, widely applicable, or detailed and narrowly applicable, i.e. the level of detail is inversely proportional to the level of applicability;
- the lack of methods to substantiate the process of building a quality model, often quality models were built subjectively, by intuition;

- low degree of formalization, lack of a rigorous mathematical basis for the description of the formal properties of models and methods for their construction and methods of adaptation;
- the absence of mechanisms for establishing a causal link between the quality of a software product and the principles, methods and technologies of their design;
- lack of a clear and effective system for measuring the quality of a software product.

Practice shows that it is impractical and impossible to develop a universal SQM, which establishes a unified system of quality characteristics, which is unchanged in time and regardless of the scope and object of the application. Users and developers feel the need for the development of an SQM applicable to describe and assess the quality of a particular SQ, taking into account its purpose, specifics and conditions of use. However, standardized SQM is not always fully suitable for assessing the quality of specific SQ. For a specific SQ being developed, it is necessary, based on its functional purpose, features, and degree of importance of individual requirements, to form an adapted quality model based on the basic standardized SQM.

To obtain a comprehensive assessment of the quality of critical software (for example, space-based software - SBS) it is advisable to use different software at the same time. Moreover, to assess the quality of each particular software product, an individual system of characteristics, sub characteristics, attributes, as well as metrics for measuring them should be formed.

The expediency of constructing a basic quality model for SBS on the basis of the standard ISO / IEC 25010 model is shown. However, for application to specific types of SBS, it must be adapted taking into account the features, requirements, application conditions by selecting the relevant characteristics and sub-characteristics, as well as the possible addition of additional attributes quality.

References

- [1] McCall J. A., Richards P.K., Walters G.F., "Factors in Software Quality: Concept and Definitions of Software Quality. Final Technical Report", *National Technical Information Service, Springfield* Vol. 1 (1977).
- [2] Boehm B.W., Brown J.R., Kaspar H., Lipow M., MacLeod G.J., Merritt M.J., "Characteristics of Software Quality", *TRW Series of Software Technology, Amsterdam, North Holland* (1978): 166.
- [3] Grady R.B., Caswell D.L., "Software Metrics: Establishing a Company-Wide Program", *Prentice-Hall* (1987): 275.
- [4] Ghezzi C., Jazayeri M., Mandrioli D., "Fundamental of Software Engineering", *Prentice-Hall, NJ, USA.* (1991).
- [5] GOST R YSO/MEK 9126-93. "Informacionnaya tehnologia. Ocenka programnoi produkcij. Harakteristiki kachestva I rukovodstva po ih primenenyu [Information technology. Software product evaluation. Quality characteristics and guidelines for their use, introduction 1994-07-01]", *M.: Izdatelstvo standartov* (1994): 15.
- [6] IEEE 1219-1993. "Standard for Software Maintenance", *Software Engineering Standards Subcommittee of the IEEE Computer Society* (1993): 45.
- [7] Dromey G.R., "A model for software product quality", *IEEE Transactions of Software Engineering* Vol. 21, No. 2 (1995): 146-162.
- [8] Hyatt L.E., Rosenberg L.H., "A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality", *Proceedings of Product Assurance Symposium and Software Product Assurance Workshop. Noordwijk* (1996): 209-212.

- [9] ISO/IEC 9126-1:2001. "Software engineering", *Software product quality. Part 1: Quality model* (2001): 32.
- [10] ISO/IEC 25010:2011. "Systems and software engineering", *Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models* (2011): 34.
- [11] Suman Manoj Wadhwa, "A Comparative Study of Software Quality Models", *International Journal of Computer Science and Information Technologies* Vol. 5 (4) (2014): 5634-5638 <http://www.ijcsit.com/docs/Volume%205/vol5issue04/ijcsit20140504177.pdf>
- [12] Sanjay Kumar Dubey, "Comparison of Software Quality Models: An Analytical Approach." Sanjay Kumar Dubey, Soumi Ghosh, Ajay Rana *Int. Journal of Emerging Technology and Adv. Engineering* Vol. 2, Is. 2 (2012): 111-119.
- [13] Jarko E., "Sravnienie modelei kachestva programmnoho obespechenia analiticheskii podhod [Comparision of quality models software analytical approach]", *Trudy. M.: IPU RAN. XII Vserossiiskogo sovechaniya po problemam upravleniya VSPU. Moscow* (2014): 4585-4594.
- [14] Gordeev A., Harchenko V., "Evolusiya modelei kachestva programmnoho obespecheniya: Metodica I rezultati analiza v kontekste standart ISO 25010 [The evolution of software quality models: methodology and analysis results in the context of ISO 25010]", *Systema obrabotki informasyi* Vipusk 6 (113) (2013): 13-31.
- [15] Ismail E., "Sovremenniy modeli kachestva programmnykh sredstv I ih osobennosti. [Modern models of software quality and their features]", *MNJ "Poisk"* No 3 (2015): 272-282.
- [16] Larin S.N., Lazareva L.YU., Larina T.S., "Modeli, metody, pokazateli, kharakteristiki i metriki, primenyayemye v ekspertnykh sistemakh otsenki kachestva razrabotki i sozdaniya innovatsionnykh programmnykh projektov. [Models, methods, indicators, characteristics and metrics used in expert systems for assessing the quality of development and creation of innovative software projects. Regional economics, theory and practice]", *Regional'naya ekonomika: teoriya i praktika* T. 15, Vyp. 6 (2017): 1187-1198
- [17] Bass L., Clements P., Kazman R., "Software Architecture in Practice", *2Ed. Addison Wesley* (2003): 528.
- [18] Chang C., Wu C., Lin H., "Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality", *Software Quality Control* Vol. 16, No. 2 (2008): 263-267.
- [19] Sharma A., Kumar R., Grover P.S., "Estimation of Quality for software components: an empirical approach", *ACM SIGSOFT Software Engineering Notes* Vol. 33, No. 6 (2008): 1-10.
- [20] Ajah I. A., & Ugah J.O., "Comparative Analysis of Software Development Methodologies", *International Journal of Advanced Research in Computer Science and Software Engineering* 3(6) (2013).
- [21] Klopper R., Gruner S., & Kourie D., "Assessment of a framework to compare software development methodologies", *The 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries* (2007).
- [22] Jeff Tian, "Quality-Evaluation Models and Measurements", *IEEE Software* Vol. 21, No. 3 (2004): 84-91.
- [23] Deepshikha Jamwal, "Analysis of Quality Models for Organizations", *International Journal of Latest Trends in Computing* Volume 1, Issue 2 (2010).
- [24] Paulk, Cuttis, Weber, "The Capability Maturity Model for Software", *Institute of Electrical and Electronics Engineers* (1997): 427-438.
- [25] Suryanto Nugroho, Sigit Hadi Waluyo, Luqman Hakim, "Comparative Analysis of Software Development Methods between Parallel V-Shaped and Iterative", *International Journal of Computer Applications* 169(11) (2017): 7-11.