ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 3, Issue 2 (Apr. - Jun. 2015)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2015)

Analysis of Aspect Oriented Software Quality (AOSQ) Model

[I]Urvashi Chauhan, [II]Shrddha Sagar
[I]Dept. of CSE, Galgotias University,Greater Noida, U.P, India
[II]School of CSE, Galgotias University Uttar Pradesh, Gr. Noida, UP, India

Abstract
Aspect Oriented Software Development (AOSD) is a new software development paradigm that provides unique and advanced program structuring and modularization techniques. This requires software engineers to define metrics and quality models for enhancing the quality of software. Several authors have defined software quality model for AOSD but none of them have discussed about the scalability of the system. Hence in this paper, we are proposing a new quality model which uses scalability as one of its quality characteristics.

Keywords
Aspect Oriented Software Development, concern, join points, point cuts, advice, Software Quality models, Scalability.

I. Introduction

The complexity of current software systems made software engineers to find new ways of developing, deploying and managing software systems. Nowadays, there are many software development methods that are used by software developers to produce high-quality software systems [1,20].

Earlier, Object-Oriented approach (OOA) was widely used, that uses the concept of Object and has special features like inheritance, data abstraction, delegation and polymorphism. Though OOA has many advanced features but it still has some limitations regarding separation of concern. For example, logging, tracing, or persistence concern of code handling module tends to be scattered and tangled all across the objects of the system [1, 20]. These concerns are known as crosscutting concerns because they cut across other functionality of the program. To address the issue of crosscutting concerns faced by OOA, AOSD has emerged as a new software development approach. It provides a way to separate concerns and at the same time ensure modularity. Basic terminologies related to AOSD systems are as follows:

A concern is an interest which pertains to the system's development, and its operation that are critical or otherwise important to one of the stakeholders [2]. The de facto standard of Aspect-Oriented Programming is AspectJ which is an extension of the Java language. Aspects are AspectJ's unit of modularity for crosscutting concerns. They are defined by these three- pointcuts, advice and introduction. In AspectJ, join points are well-defined points in the program flow, such as method calls, field sets, etc.

Pointcuts describe join points and values at those points.

Advice is a method-like abstraction which defines code to be executed when a join point is reached; pointcuts are used in the definition of advice.

Introduction defines how AspectJ modifies a program's static structure —that is the members of its classes and the relationship between classes.

Pointcuts and advice vigorously affect program flow, and introduction statically affects a program's class hierarchy [3,18].

Silent features of AOSD are discussed below

- Crosscutting concerns are removed from modules and implemented separately as aspects.
- It provides a way to bind aspects with core modules to form a working system.
- It makes it simple to understand core functionality of module by reducing code tangling.
- As aspects can be reused, this approach implements reusability.

II. Software Quality Models

McCall [4] introduced first quality model, the model differentiates between two levels of quality attributes known as quality factors. These are external attributes and can be measured directly. The second level of quality attributes known as quality criteria that can be measured subjectively or objectively. It is also known as General Electrics Model. Boehm [5] introduced another software quality model to overcome the problems of McCall's model. He included Hardware performance which was missing in McCall's model.

FURPS model [6] was proposed by Robert Grady and Hewlett-Packard Co. The model aimed at improving the management of software development processes by software industry. But this model considered only the user's requirements and disregards the developer consideration. The model fails to take into account some of the product characteristics, such as portability and maintainability [7,8,9].

Several quality models were proposed but there was no standard quality model yet. Therefore, ISO/IEC Joint Technical Committee (JTC) [10,13,14, 15] proposed a quality model called ISO/IEC Quality Model. Later , name was changed to ISO/IEC 9126 Quality Model. The ISO 9126 defines 21 attributes that a quality software product must exhibit.

ISO 9126 is a four part standard:

1) ISO/IEC 9126-1 defines an updated quality model.
2) ISO/IEC 9126-2 defines a set of external measures.
3) ISO/IEC 9126-3 defines a set of internal measures.
4) ISO/IEC 9126-4 defines a set of quality in use measures.

Dromey [16] proposed another software quality model called Dromey's Quality Model to integrate Reusability and Process Maturity as characteristics in ISO/IEC 9126 Quality Model.

All the above defined software quality models are either for legacy software or Object-Oriented software but not for AOSD [10]. Kumar et al.[11] proposed first AOSD based software quality model called Aspect-Oriented Software Quality (AOSQUAMO) Model which is an extension of ISO/IEC 9126-1 quality model. This model included four new sub characteristics

- modularity,
- code reducibility,
- complexity,
- reusability

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Castillo et al [12] proposed a conceptual quality model to integrate AOSD concepts, typical requirements engineering ideas, and the new standard ISO/IEC 25030 on software quality requirements. This conceptual model is called REASQ (REquirements, Aspects and Software Quality) Model which is integration of ISO/IEC 9126 and ISO/IEC 25030.

Kumar et al [10] presented an Aspect Oriented Software Quality (AOSQ) Model which is based on Aspect oriented Programming with some extra features and effective modularization. Four new sub-characteristics are integrated: Sustainability, Design Stability, Extensibility and Configurability under Evolvability Characteristics in AOSQUAMO Model.

Table 1: Comparison Chart of Various Software Quality Models

S. No.	Models → Factors ↓	McCall (1977)	Boehm (1978)	ISO9126 (1987)	FURPS (1991)	Dromey (1995)	AOSQUAMO (2009)	AOSQ (2012)
1.	Reliability	A	A	A	A	-	A	A
2.	Correctness	A	–	–	–	A	–	–
3.	Efficiency	A	A	A	–	–	A	A
4.	Integrity	A	–	–	–	–	–	–
5.	Usability	A	–	A	A	–	A	A
6.	Maintainability	A	A	A	–	–	A	A
7.	Testability	A	A	–	–	–	–	–
8.	Flexibility	A	–	–	–	–	–	–
9.	Portability	A	A	A	–	–	A	A
10.	Reusability	A	–	–	–	A	–	–
11.	Interoperability	A	–	–	–	–	–	–
12.	Human Engg.	–	A	–	–	–	–	–
13.	Understandability	–	A	–	–	–	–	–
14.	Modifiability	–	A	–	–	–	–	–
15.	Functionality	–	–	A	A	–	A	A
16.	Supportability	–	–	–	A	–	–	–
17.	Performance	–	–	–	A	–	–	–
18.	Process Maturity	–	–	–	–	A	–	–
19.	Evolvability	–	–	–	–	–	–	A

Table 1 shows the comparison of different quality attributes in various software quality models. The factors which are present in model are marked as 'A' which signifies that this factor has been included in the respective model and the factors which are not present are marked as '–'signifying that these factors are not included in the model.

III. Proposed Quality Model

After analyzing all the above quality models, we have found that all of these models lack scalability which is also an important factor. So, to resolve this problem, we will include scalability into existing AOSQ model. The rest of the characteristics and sub-characteristics are similar to AOSQ model. The definition of new characteristic is as follows:

Scalability: Nowadays, we are using high performance computing so, we need to know whether our system can scale (adjust proportionately) both efficiently and correctly. By integrating scalability, a system can be extended to a larger magnitude. The concept of scalability is desirable in technology as well as business settings. The base concept is consistent – the ability for a business or technology to accept increased volume without impacting the contribution margin. For example, a given program may have a capacity for 1–2000 users, while beyond 2000 users additional equipment is needed or performance will decline [19].Scalability is a crucial software quality factor in infrastructural software due to it being the fundamental basis for all applications[17]. Hence, it can be added as a sub-characteristic of maintainability.

Table 2 :Proposed Quality Model

Characteristics	Sub - Characteristics
Functionality	Suitability
	Accuracy
	Interoperability
	Security
	Reusability
Reliability	Maturity
	Fault tolerance
	Recoverability
Usability	Understandability
	Learn -ability
	Operability
	Attractiveness
	Complexity
Efficiency	Time behavior
	Resource behavior
	Code reducibility
Maintainability	Analyzability
	Changeability
	Stability
	Testability
	Modularity
	Scalability
Portability	Adaptability
	Replace-ability
	Install-ability
	Co-Existence
Evolvability	Extensibility
	Sustainability
	Design Stability
	Configurability

Table 2 shows the characteristics and their respective sub-characteristics of the proposed model. The new characteristic scalability is highlighted.

IV. Conclusion

In this paper, we have studied various software quality models given by different authors from time to time and identified that scalability was lacking in the existing model. It is very essential for today's system to have the ability to accommodate an increasing number of elements to process growing volumes of work gracefully and to be vulnerable for extension. Hence, a new sub-characteristic scalability has been added under maintainability to the AOSQ model. Every proposed model needs evaluation. To evaluate the proposed quality model for AOSD, Analytic network Process (ANP) approach could be used.

References

[1]. Jean-Yves Guyomarc'h, Yann-Gaël Guéhéneuc: "On the Impact of Aspect-Oriented Programming on the object oriented metrics",Proceedings of the 9th ECOOP Workshop on Quantitative Approaches to Object-Oriented Software Engineering (QAOOSE 2005), Glasgow, UK.

[2]. Claudio Sant'anna, Alessandro Garcia, Christina

Chavez, Carlos Lucena, and Arndt v. von Staa "On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework" Proceedings XVII Brazilian Symposium on Software Engineering, (2003).

[3]. Klass van de Berg, Jose- Maria conejero, Ruzanna chitchyan(editors) "AOSD ontology 1.0: pubic ontology of Aspect Orientation" Report of the EU Network of Excellence on AOSD, 2005.

[4]. J. A. McCall, P. K. Richards and G. F. Walters, "Factors In Software Quality - Concept and Definitions of Software Quality," Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, New York, Volume 1, Number 3, November, 1977.

[5]. B. W. Boehm, J. R. Brown and M. Lipow, "Quantitative Evaluation of Software Quality," IEEE Computer Society Press, Page No.: 592 - 605, 1978.

[6]. J.M. Zain "Software Quality Models: A Comparative Study". (Eds.): ICSECS 2011, Part I, CCIS 179, Page No. 46–55, 2011.

[7]. Deepshikha Jamwal, "Analysis of Quality Models for Organizations", International Journal of Latest Trends in Computing, Volume 1, Issue 2, December 2010.

[8]. Divya Prasad Narayani, Poonam Uniyal, "Comparative Analysis of Software Quality Models", International Journal of Computer Science and Management Research", Volume 2, Issue 3, March 2013.

[9]. Suman , Manoj Wadhwa"A Comparative Study of Software Quality Models" (IJCSIT) International Journal of Computer Science and Information Technologies, Volume 5 , 2014.

[10]. Pankaj Kumar "ASPECT-ORIENTED SOFTWARE QUALITY MODEL: THE AOSQ MODEL"Advanced Computing: An International Journal (ACIJ), Volume 3, Issue 2, March 2012.

[11]. A. Kumar, P. S. Grover and R. Kumar, "A Quantitative Evaluation of Aspect-Oriented Software Quality Model," ACM SIGSOFT Software Engineering Notes 7,2009.

[12]. Castillo "REASQ model "Published by ETH Zurich, Chair of Software Engineering, ©JOT 2010..

[13]. ISO/IEC 9126-1: Software Engineering - Product Quality-Part 1: Quality Model, International Organization for Standardization, Switzerland, 2001.

[14]. ISO/IEC 9126-2: Software Engineering - Product Quality-Part 2: External Metrics, International Organization for Standardization, Switzerland, 2002.

[15]. ISO/IEC 9126-3: Software Engineering - Product Quality-Part 3: Internal Metrics,International Organization for Standardization, Switzerland, 2003.

[16]. R. G. Dromey, "A Model for Software Product Quality," IEEE Transactions on Software Engineering, Volume 21, Number 2, Page No.146 - 162, February 1995.

[17]. Deutsch, M. S., and Wills, R. R., "Software Quality Engineering; A Total Technical and Management Approach", Prentic-Hall, Inc. 1998.

[18]. Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., and Griswold, W. G., "An Overview of AspectJ", In Proc. of the 15th European Conference on Object-Oriented Programming, Lecture Notes in Computer Science, Volume 2072. Springer-Verlag, London, Page No. 327-353, June 18-22, 2001.

[19]. Web Site: http://en.wikipedia.org/wiki/scalability.

[20]. Meyer, B: Object-Oriented Software Construction (Book/ CD-ROM) (2nd Edition),Prentice Hall PTR (2000).