Generalized Classification of Software Quality Attributes

K. Paithankar[1], M. Ingle[2]

Abstract

As per ISO 8402 standard, quality of software is the totality of features and characteristics of a product/ service and its ability to satisfy stated/ implied needs. Quality attributes possess the characteristics associated with the system and act as the measures to determine the quality of the software. Each quality attribute has been pertaining to specific feature or property of software and has unique effect Software quality attributes contribute equally for successful software development even though all sub characteristics do not equally affect software product and process. Though several issues related to the quality attributes and the respective classification have been discussed through variety of models and standards, each classification has cited with a specific view. Therefore a strong need has been emerged to systematically categorize quality attributes in an integrated manner. This paper proposes a generalized classification of quality attribute that leads to provide support throughout the software development process to produce quality software in view of product, process, policy, plan and service.

[1] Assistant Professor, Shri Vaishnav Institute of Technology & Science,Indore. E-mail : paithankars@rediffmail.com

[2] Professor & Senior System Analyst, School of Computer Science & Information Technology, DAVV, Indore. E-mail ; maya_ingle@rediffmail.com

1. Introudction

As per ISO 8402 standard, quality of software is the totality of features and characteristics of a product/ service and its ability to satisfy stated/ implied needs. Therefore, quality is defined to be a complex and multifaceted concept [1]. In the context of software engineering, software quality is a measurement of wellness of software design and conformance of the design so as to provide substantial benefit to the end users. Quality attributes possess the characteristics associated with the system and act as the measures to determine the quality of the software. Also, these attributes guide the software engineering process for conformance of all the aspects of software development. It is mandatory for any software development organization to produce quality software; particularly in this competitive era and quality attributes play a major role in determining the quality of software. Each quality attribute has been pertaining to specific feature or property of software and has unique effect. Hence, classification of quality attributes with different perspectives is required for the successful software development thereby providing guidance and ease to conform quality of product, process and service. We discuss various views of software quality attributes and existing classification models of software quality attributes in Section 2. Section 3 describes the proposed quality classification with six major classes based on some specific criteria. These classes of quality attributes have been further divided into subclasses and so on. Finally,

we conclude with the results and discussions in Section 4.

2. Literature Review

Many researchers have presented certain issues related to the quality attributes and the respective classification. As per International Standards Organization (ISO), quality standards cover all the aspects of quality. ISO 9126 is an international standard for the evaluation of software quality and has been covered with four major aspects namely quality model, external metrics, internal metrics and quality in use metrics respectively. Quality model has been described in ISO/IEC 9126-1:2001. ISO/IEC TR 9126-2:2003 has covered the external quality metrics. Internal metrics has been defined in ISO/IEC TR 9126-3:2003. ISO/IEC TR 9126-4:2004 used to define quality in use metrics. In addition ISO/IEC 25000:2005 has also been used as guide for Software product Quality Requirements and Evaluation (SQuaRE). As per ISO 9126-1, the classification of software quality explores in terms of a structured set of characteristics as functionality, reliability, usability, efficiency, maintainability and portability, which have further been divided into quality attributes.

There exists a variety of quality classification models such as McCall's model, Boehm's model, model of Software Assurance Technology Center (SATC), Software Quality Institute model, common subsets model etc. It is observed that there exist three important aspects of software product such as operational characteristics (also known as product operations), ability to undergo changes (product revision) and adaptability to new environment (product transition) [2]. Operations aspect of the system covers correctness, reliability, usability, integrity and efficiency whereas maintainability, flexibility and testability considered being revision aspect elements of a system. Portability, reusability and interoperability attributes deal with adaptability of system. Another model represents a multilevel classification of quality attributes [3] [4]. General utility is at the root level representation which has further been expanded through quality factors maintainability, as-is utility and portability.

Maintainability quality factor covers sub-factors modifiability, understandability and testability. Human engineering, efficiency and reliability have been encapsulated within as-is-utility whereas device independence and self-containedness considered being the sub-factors of portability. Further subdivision of these quality sub-factors includes many crosscutting attributes. The model views the decomposition of quality attributes into source code characteristics. The quality can be viewed in terms of goals such as requirement quality, product quality, testing effectivity and implementation effectivity [5]. Ambiguity, completeness volatility, understandability, traceability, structure, maintainability, reusability, internal/ external documentation, resource usage, completion rates and correctness are the attributes identified to achieve the stated goals. One improvement in ISO-9126 classification includes reusability as the top level attribute with additional subordinate properties. Functionality, reliability, efficiency, usability, maintainability, portability and reusability are defined as the primary classes of attributes. Further subdivision of these classes includes the attributes to be used in other environments [6].The quality attributes in the form of subsets may be applicable for a specific phase of software development. A group of certain attributes such as Reliability, Availability, Serviceability, Usability, and Installability (RASUI) has been referred for effectiveness of the system, whereas a collection of attributes such as Functionality, Usability, Reliability, Performance, and

Supportability (FURPS) has been proved important for software requirements conformance. Reliability, Availability, Scalability, and Recoverability (RASR) is another set of quality attributes recommended for assessing quality of the database. Some quality attributes such as Reliability, Availability, Maintainability and Safety (RAMS) are clustered together to deal with safety-critical systems [2].

Our literature survey reveals that each classification has cited with a specific view. Coverage of limited attributes and their usage have been primarily observed limitations. Further, it has been noticed that software quality attributes contribute equally for successful software development even though all sub characteristics do not equally affect software product and process [7]. Also, certain quality attributes such as functionality, reliability, availability and usability have been common in most of the classifications, but have been composed in different manner. In addition, many more quality factors are identified which have not yet been classified relevantly [8] [9]. In the perspective of literature study, there exists a scope of generalized classification of quality attributes with systematic categorization of attributes in an integrated manner.

3. Proposed Classification

We present a generalized classification of quality attributes with six major classes namely; Runtime attributes, Non-runtime attributes, Business Oriented attributes, Architecture Oriented attributes, Domain Specific attributes and Impact Oriented attributes in view of product, process and service as shown in Fig. 1. We describe each of these classes and its further classification as follows:

Runtime Attributes

This class of quality attributes is strictly concerned with the execution time. These attributes are required to be measured at the time of system execution. At run time it is important to know about the working of the software and its behavior. It highlights on the fundamental process of transformation that software and hardware components of the system perform on inputs to produce outputs. For example response of the system is considered to be of great importance and hence at the time of user-system interaction the time and functional aspects of the software have to be thought about. Therefore, the quality attributes concern with the software in this manner has taken up in runtime class. As shown in Fig. 2, there exists secondary attributes such as security, functionality, interoperability performance, accessibility, usability, availability, and traceability that are used to assess the run time behavior of the system. Security is the first sub-class deals with the ability of the system to resist unauthorized attempts of usage/ behavior modification while providing service to users. It protects information from theft or corruption, or the preservation of availability. One more class of runtime attributes is interoperability. It is the ability of system to cooperate with other systems while in execution and hence concerned with run time behavior of the system. Yet another class of run-time attributes is accessibility, which is the ability to access the functionality of the system, and to get possible benefit of the system, when the system is in operation. Traceability is the ability to verify the history, location, or application of an item by means of documents at the time of system execution.

Functionality is another sub-class of runtime attributes and is the ability of the system to accomplish intended work. It has been further classified into understandability,

conciseness, consistency, clarity (UCCC). Understandability refers to the capability of system being understood and accepted under the circumstances to accomplish any task. Conciseness is concerned with expressing more in few words. Consistency is the property of uniformity of successive results or events and is considered as part of functionality. Clarity refers to the ability of the system that clearly visualizes the concepts, as in thought, understanding and mind. Another sub-class of run time attributes is performance which is the ability of the system to be timely used as desired. It has been further classified into attributes such as utility, response time, throughput behavior of the system, timeliness and structure (URTTS). Utility is the capacity of the system to work as needed. Response time is the time elapsed from submitting an instruction till the first response of the system. Throughput behavior of the system is defined as the amount of work done in a given time. To improve performance of the system maximum throughput is expected. Timeliness being the property of on time plays a major role for better system performance. To get the desired performance of the system, sufficient storage of data and results has necessarily to be provided. Subsequently usability is another sub-class of run rime attributes which is a measure of convenience and practicability of the product for intended users. Ease of use and ease of training to the end users pertaining to the system proved to be of great importance. Thus, usability has to be conformed at run time. It has further been subdivided into set of attributes working to have usable software namely; learnability, efficiency, access control, unambiguty, validity, resilience, customizability, practicability and operability (LEAUVRCPO). Learnability is the ability to know the details of the system without any external support. Efficiency being the property of a system to fulfill the requirement perfectly in short runs of time desired to be attribute of usability. Access control deals with the protection and security of the system when in use. Repeatability defined as the variability of the measurements obtained by one person while measuring the same item repeatedly. Another attribute in usability subset is unambiguity that is the ability of the system being interpreted in the same manner by all the users. It makes the users understand the intended purpose of the system. Validity refers to the logical, analytical or necessary trueness of the system at the time of use. Resilience is considered to be usability attribute being the property of system to energize itself when deformed elastically and then, recover. The software is used by many users for different purposes and satisfaction of the users is important. Customizability is the ability of the software to be changed by the user or programmer as per the need and to provide satisfaction. Practicability is the ability that makes the software usable for a specified purpose. Operability is the major aspect of usability and defined as ability to keep a system in a functioning and operating condition.

Availability is also a sub-class of runtime quality attributes and described as the measure of time when the system is up and running correctly. It is the elapsed time between failures and the time needed to resume operation after a failure. Availability branches into set of attributes such as reliability, sustainability and anomaly management (RSA). Reliability is essential to confirm availability since it is the capability of software to maintain its level of performance under stated conditions for a stated period of time. Sustainability also plays an important role to measure availability as it is the property to uphold the system for the required function. Anomaly management deals with anomalies that may exist while system is running.

Non-runtime Attributes

Attributes of this class do not concerned with run time behavior of the system but play major role in determining quality in offline manner. Also, it has been noticed that the conformance of run time quality depends on the conformance of non-run time quality. It specifies criteria that can be used to judge the operation of a system and the way the system will do it. For example, software performance requirements, software external interface requirements, software design constraints are considered to be nonfunctional requirements of the system and are embodied in the static structure of the software system. Nonfunctional requirements are difficult to test; therefore, they are usually evaluated subjectively with overall characteristics. Fig. 3 depicts such attributes that can evaluate non runtime behavior of the system namely; documentation, manageability, completeness, portability, accuracy, integrity, reusability and testability.

The documentation sub-class of non-run time attributes refers to the process of preparing and providing evidences in the form of communicable material for system investigation. Successful system development has the essential requirement of well documentation. Completeness is concerned with implementing all the capability in terms of sufficient data items, functions, interface and code and hence it is considered as a class of non-run time attributes. The sub-class accuracy has been defined as the degree of conformity of a measured or calculated quantity to its actual value. Integrity is also a non-run time attributes and defined as the ability to separately develop components and make them work together correctly. Components are functioning in specific manner to achieve objectives, at the same time collectively working to achieve common goal of the system. Yet another class of non-run time attributes is testability and has given a great importance as it provides error handling capability to the system. Testing of the system has to be accomplished prior to the execution of the system, thus testability is the major non-runtime attribute.

Manageability is also a sub-class of non-runtime attributes and is the ability of the system to plan, organize resources, direct, administrate and control the overall process of development. Manageability has further extended in a group of attributes, modifiability, maintainability and flexibility (MMF) with equal share of each attribute for the intended purpose. Modifiability is the ease with which a software system can accommodate required changes. Maintainability facilitates updates to satisfy new requirements. The software product that is maintainable is simple, well-documented, and should have spare capacity for processor and memory usage. Flexibility is defined as the ability to adapt to different circumstances.

Another sub-class of non-run time attributes is portability. Portability is the ability of a system to run under different computing environments. The environment is a combination of hardware and software both. A set of attributes, machine independence, system independence, replaceability, installability, adaptability and data commonality has been covered under portability and abbreviated as MSIADR. Machine independence refers to the ability of the software to be used on any machine (i.e. hardware). System independence is the ability of the software to be installed, operated and modified on any system (i.e. hardware and operating system). Presence of these two attributes generates possibility of installability which is the capability of the software product to be installed in a specified environment. Installability is further extended to distributability that is the ability to share the common resources for applications

and users. Adaptability leads to improve portability as it is the ability of the system to be modified by circumstances. Data commonality refers to the availability of common data for all the users and applications. Replaceability is the capability of the system to retain itself after replacement of data structure, function, module or program.

Reusability is referred to be the degree to which existing applications can be reused in new applications and hence identified as a sub-class of non-run time attributes. It has further been classified into a cluster of attributes recommended to measure reusability, such as representation independence, application independence, data encapsulation, function encapsulation and interfaceability (RADFI). For reuse, system has to be independent of representation and application used. Data and function encapsulation are necessary for reusability as it refers to hiding the details about data and function. A major characteristic of reusability is interfaceability, which deals with exchanging information across the common boundary shared by the components.

Business Oriented Attributes

There exist many non-software attributes that influence other software or non-software quality attributes. These attributes, attempt to conform quality in view of business policies. Business objectives are specific statements that give projections about growth or development to companies. these are the stated, measurable targets to achieve business aims. The effective business objectives have to be specific, measurable, agreed realistic and time specific. For example, a business objective could be to increase sales of the product by next year. A combination of attributes such as cost and schedule, economy, marketability, appropriateness for organization, and localization are observed to be quality attributes that affect business system as highlighted in Fig. 4. Cost and schedule is the ability to account cost of the system with respect to time and market, expected project lifetime, and utilization of systems. It plays significant decision making role for planning and executing business policies. Another class of business oriented attributes has termed as marketability. It is the capability to use the system with respect to market competition and to use feedback to increase profitability of the business.

Yet another sub-class of business oriented attributes is appropriateness for organization that has been explored for availability of the human input, allocation of expertise, and alignment of team and software structure. Also, it facilitates business process re-engineering to provide adaptable system. Generality and commonality (GC) are the factors that contribute to measure appropriateness of organization. Generality refers to the availability of the system for the majority of people while commonality is the ability of the software to be used with common features for distinct users. Localization is stated to be the major sub-class of business oriented attributes as any business system has to be localized to realize benefits.

Architecture Oriented Attributes

The quality attributes used to measure structural aspect of the system are known as architecture oriented attributes. There are many common ways of designing computer software modules and their communications. For example, blackboard, Client-server, Database-centric architecture, Distributed computing, Event Driven Architecture, Implicit invocation, Monolithic application, Peer-to-peer, Pipes and filters, Plug-in, Representational State, Transfer, Structured (module-based), Service-oriented, Search-oriented, Space based, Shared nothing, Three-tier model are some of the available architectures

can be used while system development. These are also non software attributes. Fig. 5 illustrates attributes that are considered for assurance of the quality of structural design of the system namely; conceptual integrity, correctness, structured, scalability, extensibility, supportability and self-containedness. Conceptual integrity has been defined as the integrity of the overall structure composed from variety of small architectural components. Therefore, it plays a vital role for building quality software. Accountability to satisfactorily fulfill all requirements of the system is expressed through correctness. Realization of requirements has dependency on design; hence correctness has been covered under architecture oriented attributes. The structured aspect of the system is the capacity to organize activities in well manner. Scalability deals with handling growing amounts of work in a graceful manner, or to be readily enlarged. Extensibility is a system design principle where the implementation takes future growth into consideration. It is a systemic measure of the ability to extend a system and the level of effort required to implement the extension. Another class of architecture oriented attributes is supportability that has been defined as affectivity of system structure to provide service to the intended users. It has been proved that the basis of a desired system is a good architecture and hence essential to provide support of use. Self-containedness is the ability of the system to contain its projected purpose in its definition and hence considered as architecture oriented attribute.

Domain Specific Attributes

This class of quality attributes deals with the specific business and application domain. The objective of defining the domain is developing mass-customized products that reduce the costs, delays, and inflexibility characteristic of software and systems. Also it increases the ability to align business activities to the needs and produce solutions to cater the specific needs. It has further been categorized in three sub-classes that can contribute for the stated purpose such as sensitivity, calibrability and stability as represented in Fig. 6.

Sensitivity is described as the degree to which a system component can pick up something being measured. It has dependency on the area of application and development. Further, calibrability is defined as the ability of the system to recalibrate itself to some specific working range. It has been associated with unique standards and measurements against the standards for evaluating product and process. Stability is the sub-class of domain specific attributes and is the degree to which software can be run over periods of time without crashing or otherwise malfunctioning. It has termed as a characteristic of application area and the plateform used to build the system.

Impact Oriented Attributes

This class of quality attributes is mainly concerned with determining the overall effect of the system from users view point. User satisfaction is the primary goal of any software and has to be cultivated at conception stage of the system. Therefore, software has to be available in such a manner that influences the user without a need of redressal. Hence, Simplicity, communicativeness, maturity, self-descriptiveness, fault tolerance, affect and helpfulness are the major factors contribute to evaluate this aspect of the system as illustrated in Fig.-7.

One class of impact oriented quality attributes is simplicity that is defined as the property of being simple or uncombined i.e. the availability of the system without complexity. Also, a simple system is easy to understand

and use, and hence proved effective. Another class of impact oriented attributes is maturity. It is the state of the system being sensible and fully grown. Hence, maturity has considered to be directly proportional to effectivity. Yet another class of attributes is fault tolerance that is the ability of a system to continue performing operations properly, even in case of failure of some of its components. Since it has been desired for proper functioning of the system, it is considered to be the impact attribute. Installability is also a sub-class of impact oriented attribute as it is the capability of the system of being loaded and used effectively. Affect is the measurement of affectivity of the system over organization wide operations. Any decision has effect over the organizational process. Therefore, affect has been considered to have significant share in view of totality of the system. Communication is another class of impact attributes as it is the ability to convey desired and expected information to the intended users, organizational process or within the system. Finally, we have a class of impact oriented attributes as self descriptiveness. It is interpreted as the capability of the system to sufficiently describe itself and hence shows suitability with impact oriented class.

We consider Helpfulness as further sub-class of impact oriented attributes. It is the degree with which system is ready to help users and hence justify itself to be the attribute of this category. Helpfulness is further classified into a pair of attributes, visibility and survivability and is denoted by VS. Visibility is the transparency of system in context of interface, data and communication. Survivability refers to the ability of the system to persist in the same mode till use.

4. Results And Discussion

In the proposed classification, we have made efforts to reconcile the limitations of the existing classification approaches with the stated aim. Certain attributes such as functionality, performance, usability, portability, reusability have been reorganized in perspective of suitability for improvisation of quality in respective class. Also, the proposed quality classification exclusively focuses on business and economy view of the system. In addition, impact oriented attributes have been incorporated to extend the overall quality of the system. Thus, the classification proves to be beneficial to produce quality software, as it is a unified approach to make use of the attributes for the product, process, policy, plan and service. The attributes have explicitly classified with defined role, but does not account for the crosscutting attributes. Therefore, in view of crosscutting attributes, still there remains a scope of further improvements.

References

[1] D. Garvin, *"What Does Product Quality Really Mean?"*, Sloan Management Review, Fall , PP. 25-45, 1984.

[2] Roger S. Pressman, *"Software Engineering – A Practitioner's Approach"*, Fifth Edition, International Edition.

[3] Rechard E. Fairly, *"Software Engineering Concepts"*, Tata McGraw-Hill Publishing Company Limited.

[4] IEEE standard Glossary of Software Engineering Terminology, IEEE Standard 729-1983.

[5] L. Hyatt, L. Rosenberg, *"A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality"*, NASA/Goddard Space Flight Center, Greenbelt MD, USA 20771.

[6] R. Geoff Dromey, *"Software Product Quality: Theory, Model, and Practice"*, Software Quality Institute,

Griffith University, Nathan, Brisbane, QLD 4111 AUSTRALIA, Software Quality Institute 09/03/98.

[7] Khashayar Khosravi and Yann-Ga¨el Gu´eh´eneuc, *"On Issues with Software Quality Model"*, GEODES - Group of Open and Distributed Systems, Experimental Software Engineering, Department of Informatics and Operations Research, University of Montreal, Quebec, Canada.

[8] *"Trade-off Analysis of Software Quality Attributes"*, Special Issue, Software Quality Journal, Springer Netherlands, November 14, 2005.

[9] Mikael Syahnberg, Claes Wohlin, Lars Lundberg, Michael Mattsson, *"A method for understanding quality attributes in software architecture structures"*, Blekinge Institute of Technology, Ronneby SWEDEN, ACM International Conference Proceeding Series; New York, NY, USA Vol. 27, NewYork, NY, USA.

Author's Biography

Dr. Maya Ingle is Professor & Senior System Analyst at School of Computer Science and Information Technology, Devi Ahilya University, Indore since 22 years. She is also Chair Assessment & Research committee. She is Ph. D. in computer science and M. Tech in Computer Science from IIT, Kharagpur. Her research interest includes Software Engineering, NLP, Speech Recognition, Usability Engineering and Agile Computing. She has published more than 60 papers in national and international journals and conferences.

Kshama Paithankar is Assistant Professor at Shri Vaishnav[SM] Institute of Technology & Science, Indore. She is pursuing Ph. D. in computer science and has 14 years of teaching experience. Her research interest includes Software Engineering and Usability Engineering. She has published more than 05 papers in national journals and conferences.

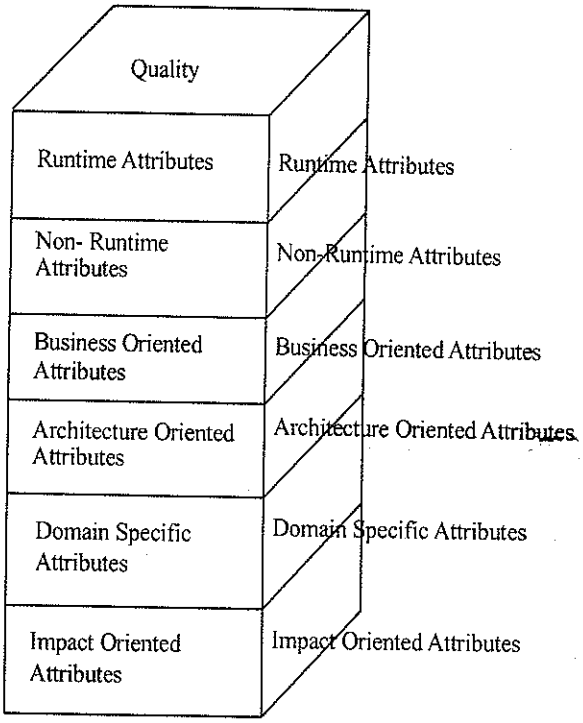


Figure 1: Major Quality Attribute Classes

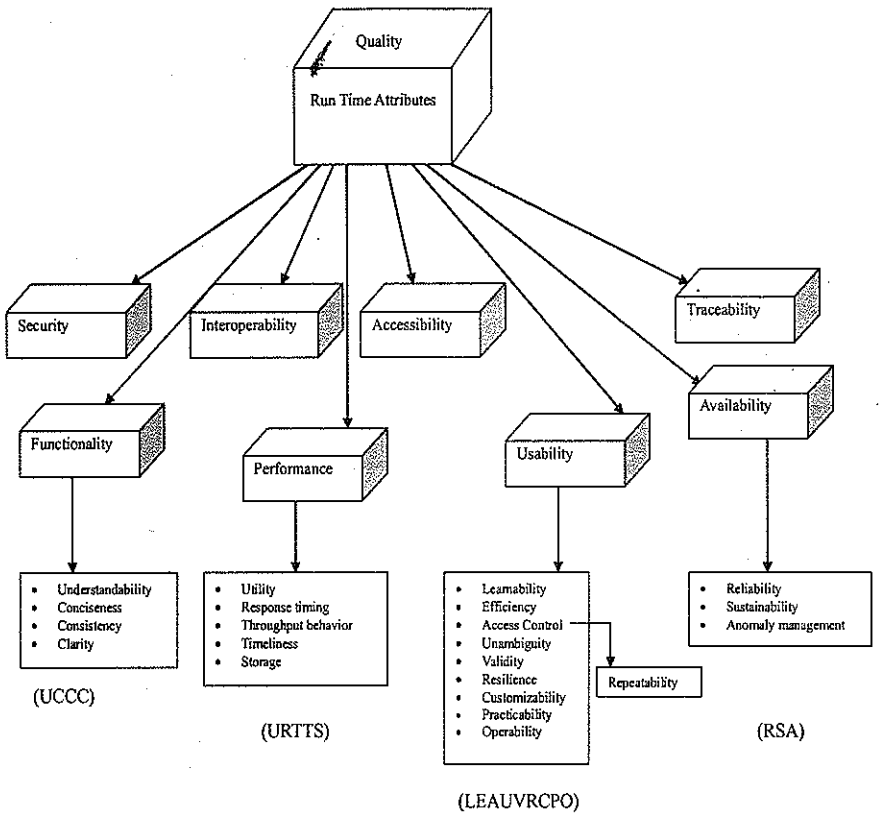


Figure 2 : Run Time Attributes

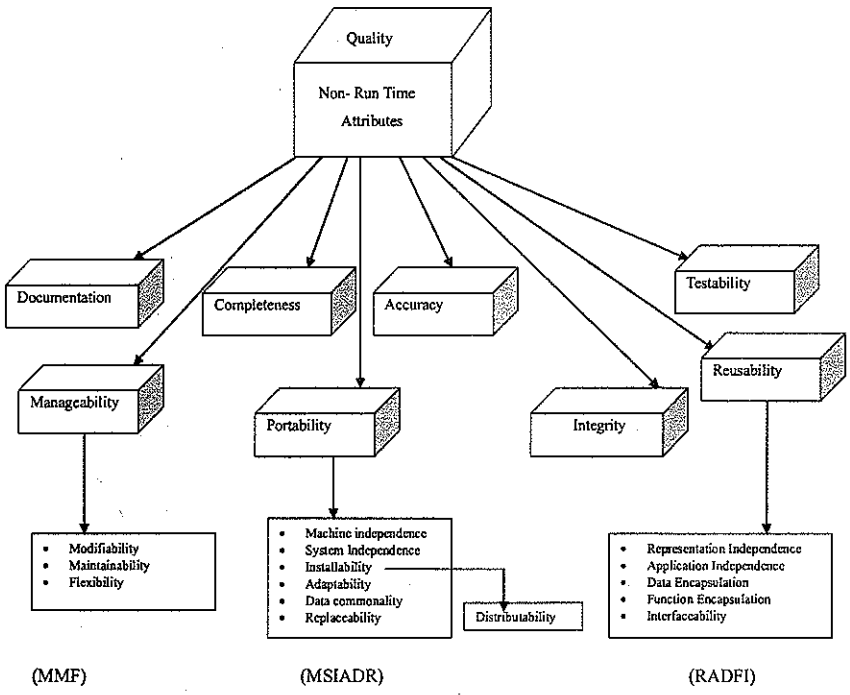
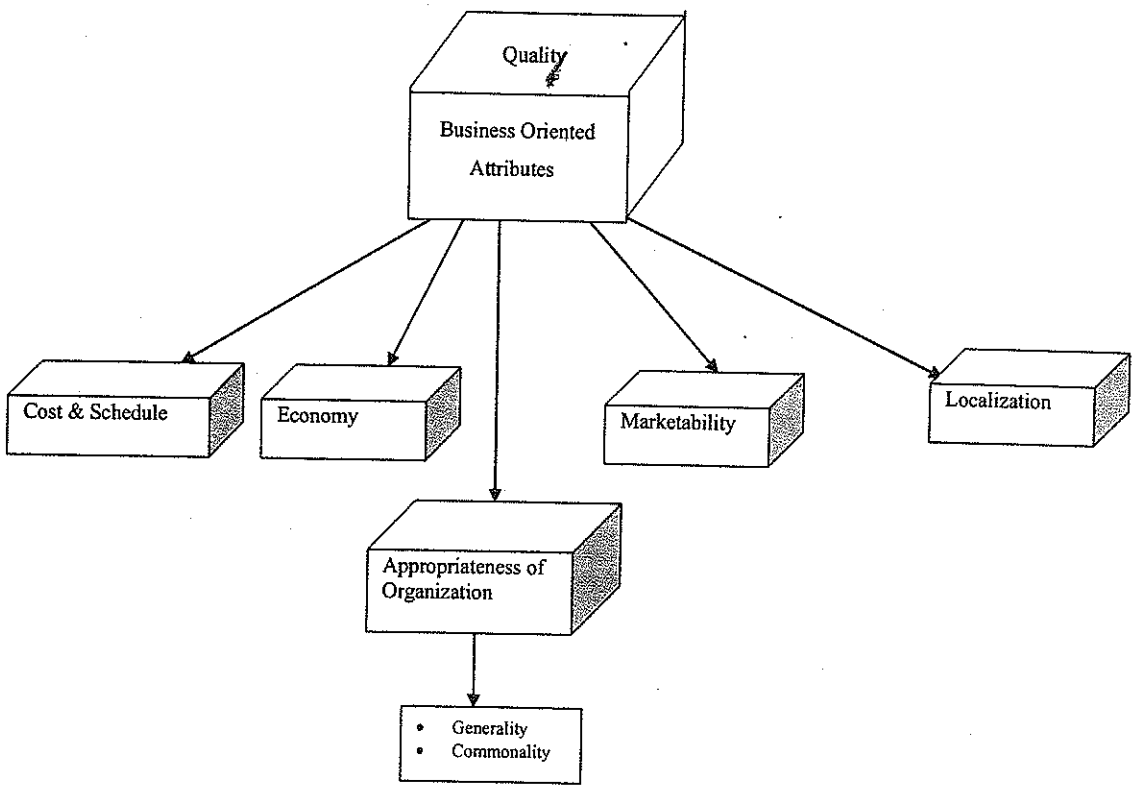


Figure 3 : Non Run Time Attributes



(GC)

Figure 4 : Business Oriented Attributes

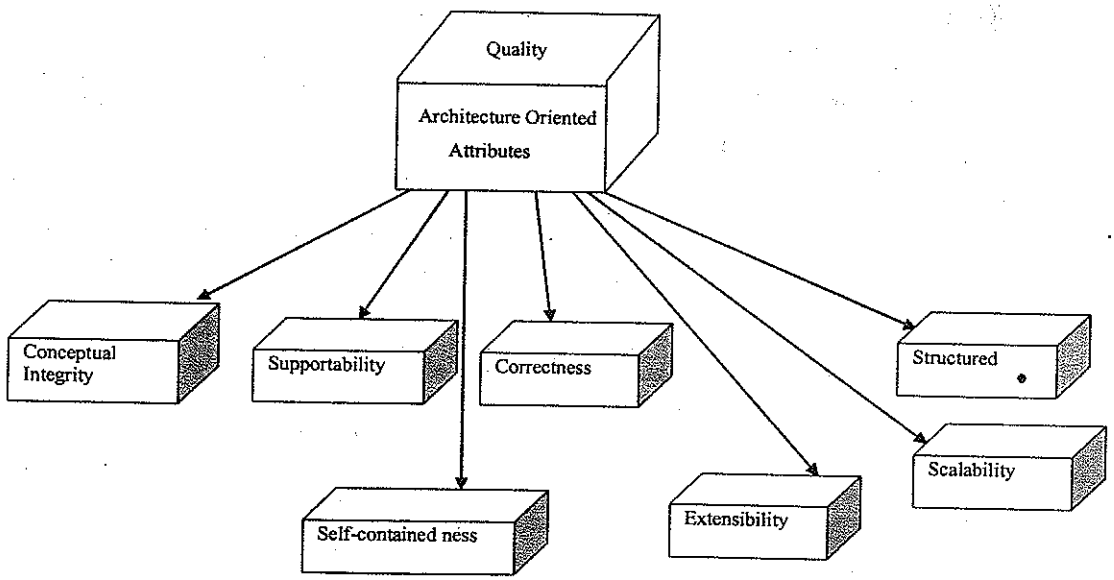


Figure 5 : Architecture Oriented Attributes

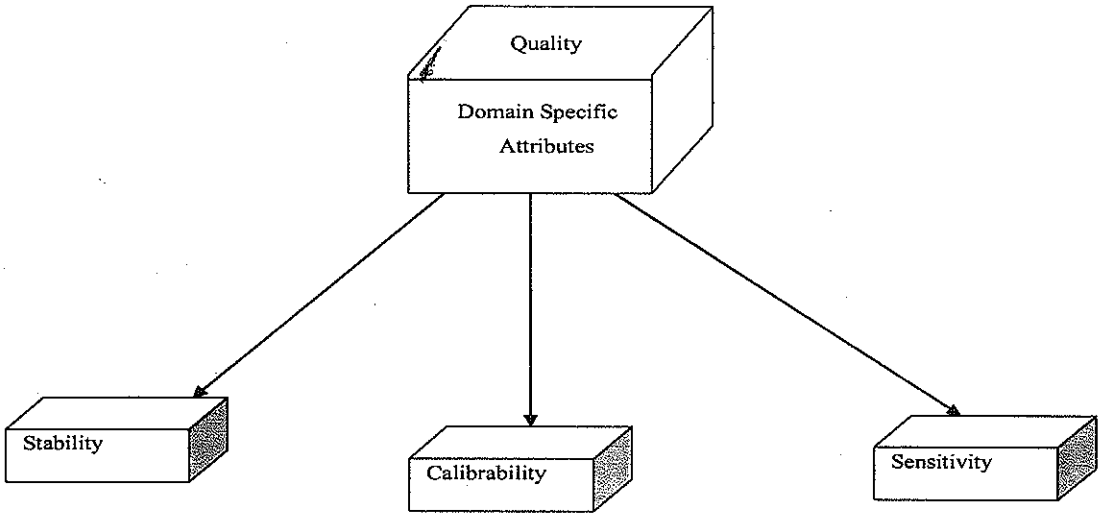
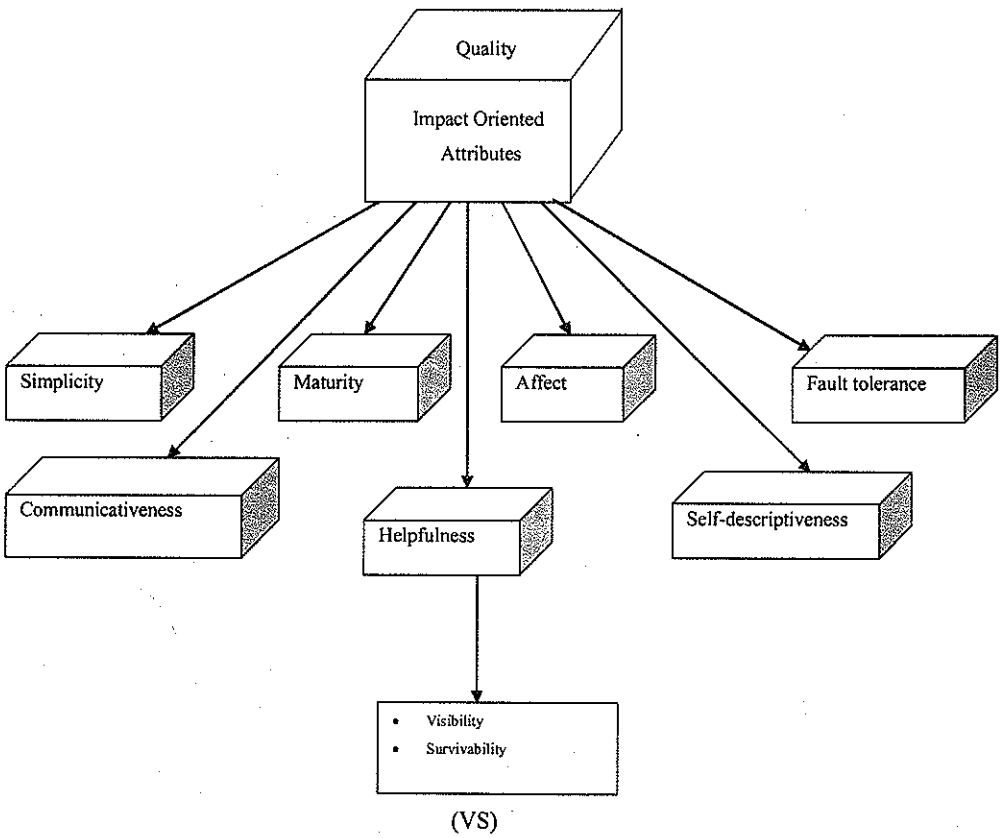


Figure 6 : Domain Specific Attributes



(VS)

Figure 7 : Impact Oriented Attributes