

The Role of Software Architecture and Trusted Computing in Software Design Guiding Principles and Quality Assessment Standard

DIAO Mingguang

School of Information Engineering,
China University of Geosciences,
Beijing, China
dmg@cugb.edu.cn

XUE Tao

School of Information Engineering,
China University of Geosciences,
Beijing, China

ZOU SenZhong

School of Information Engineering,
China University of Geosciences,
Beijing, China

Abstract—In software engineering, software design guiding principles and quality evaluation standard is not only an important theoretical problem but also a very practical problem. Against to some inappropriate dissertation in the popular textbooks, we put forward that the rationality of software architecture and the trusted computing method of software design should be included in the guiding principles and quality evaluation standard of software design. On this basis, we put forward the proposals of the guiding principles and evaluation criteria on software design for further research and discussion.

Keywords—The goal of software design; software design Quality Evaluation Standard; Software Architecture; Trusted Computing Method

I. INTRODUCTION

The objective of software engineering is to produce high-quality software systems or products to meet user needs.

Since the emergence of software engineering, software designed to follow the law, guiding principles and quality metrics has been one of great concern.

Due to the historical limitations, many expositions on guiding principles and quality metrics of software design have improper expression or one-sidedness, some are out of mainstream and the reality of design development in recent years, the outstanding problem is that does not reflect the current focus of software design and update knowledge rationally. The situation is very undesirable for software engineering-related personnel to establish correct idea of software design, so as for how to evaluate the software design quality in the software industry. In this paper, we study the issue and discuss some of key issues in current software design, thus to put forward new proposal to the guiding principles and metrics of software design.

II. THE STATUS OF SOFTWARE DESIGN AND QUALITY EVALUATION PRINCIPLES

Many publications about software engineering and software design all discuss on the technical guidance and evaluation standard of software design. Stated in early discussion, in order to achieve the goal of software design, software design metrics

should have structure hierarchical, modularization of component, design abstraction, high independence of module, interface simple, drivable and repeatable design method and such characteristics^[1]. Then other scholars launch that software design should reflect the system hierarchical, modular, and also should be able to clear the content of data, architecture, interfaces and modules, combination of decomposition and refinement, to use schema integration, thus to ensure the consistency and integration of design^{[2][3]}. Recently, it declines and goes back to the early discussion^[4].

Furthermore, regard to the issue of measuring design model of technical standards, the design model should be a recognized design pattern to build, has a good design characteristics and can implement hierarchy by evolutionary way^[5,6]. Obviously, these design principles or metrics are able to summarize and reflect more comprehensive technical elements and characteristics of the structure of software design, all emphasize the designed structure should be hierarchy, and it's related design techniques are mature, inheritable technology. But, for software engineering design method of component-based software engineering and network era, these guidelines or metrics have not form a guideline that can make sure software design can has good architecture and meet the credibility requirement under the network environment.

However, with the rapid development of software industry, especially with the development of software engineering in network era, the applications oriented by software design develop rapidly, the mainstream technology and methods also develop and change a lot accordingly, so these deals seem to have limitations because of changing times and software engineering industries.

It is generally accepted that software engineering experiences three important stages of development in terms of software design techniques: structure-oriented technique, object-oriented technique and technique based on software architecture reuse. Therefore, the evaluation criteria should be updated as well. In software engineering design based on component and network era, we should use different patterns of software architecture to constitute good architecture guidelines which can guide software design, and with trusted computing to make sure the software designed to meet requirements of

software design under network environment, thus to build a software design guideline based on good architecture and trusted computing.

III. ARCHITECTURE DESIGN SHOULD BE ONE OF THE SOFTWARE DESIGN AND QUALITY EVALUATION PRINCIPLES

Software architecture designed, from the problem domain space to the first activity of the answer domain space, it is directly affect software quality: The division of modules that may affect the level of division of the system; The division of functions and packages that may affect the scalability of the system; Communication protocols between components that may affect the efficiency of the system and etc. Studying the good nature of software architecture, one can understand the structure of the system to meet the demand for quality level, to identify potential quality problems, to be revised to better reflect user needs; the other hand, the system can predict the future quality attributes, design and implementation of the system to provide constructive suggestions.

Good software architecture should serve as guiding principles for software design and quality evaluation of one of the criteria that guide the design and evaluation of software design quality, it should be stressed that the importance of software architecture. A good design should be followed and the use of software architecture research field theory and the latest achievements in the mature. The rapid development of software engineering, Internet-based software applications to provide software services model SaaS has become the latest trend in the development of software engineering, and network environment, the large-scale software systems increasingly complex development needs and environmental requirements to run the software with good quality attributes. Given good software architecture can effectively ensure software quality, software design guidelines in the evaluation of standards and quality it is necessary to add the software architecture of the test well.

Characteristics of good software architecture requires:

- (1) Identified using well-established architectural style or design pattern to create the required design tasks of software architecture;
- (2) Constituted by components, which show the characteristics of good design?
- (3) The evolution of the architecture can be achieved, with good dynamic;
- (4) Easy to implementations and testing later^[7].

General software design should meet these characteristics of software architecture so that it can ensure the software design meets the requirements and become a good software design.

In order to meet the Internet-based software design environment and application requirements, we should consider the software architecture evolution under the network when architecture implementation by evolution, Therefore, at the time of software architecture evolution, we come up with a assessment model of software design which suitable evaluation

of general software design and based on network service environment, as shown in Figure 1.

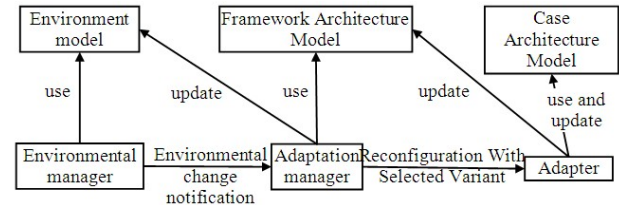


Figure 1. Deducing and evaluating variants of Software Architecture

When the requirements or network environment changes, the evaluating model about variants of Software Architecture can be used for inference and evaluation of software design. When these changes cause the software architecture's changing, this model's way of implementation is in the following manner:

(1) Environmental manager, using existing environment models to examine the environmental requirements of the existing software architecture, find and get the environment changes of software architecture, then tell Adaptation manager.

(2) Adaptation manager gets the information of environmental changes passed by environmental manager, using established framework architecture model to check the environmental changes of existing framework architecture, find the differences between framework architectures after these changes, form the variations of the framework architecture. At the same time, under this change to amend the original environment model, then form a new one. Furthermore, the framework architecture of the new variant will be passed to the adapter and reconfigured.

(3) Adapter uses original instance of architecture model to check the variant information of framework architecture passed by Adaptation manager, analysis the differences between the original instance of architecture model and variant, then reconfigure the architecture. At the same time, amends and forms the framework architecture model after environment changes. And, automatically update the instance architecture model, thus to make architecture adapted to current needs and changes of network environment.

Because in the process of software development, demand or environment may change at any time, especially in today's work in a network environment and service delivery software system, so the final Software Architecture may be differ greatly to the initial. A good architecture, should be designed with a strong flexible and can be realized by dynamic method, meeting new need of software. Of course, the evolution also would be a progressive understanding of the problem domain and the technology method of carrying out system. Whatever, Software Architecture can get the strong support and greatly reduce the workload of altering. Depending on evaluating model about variants of Software Architecture, in the development process based on component, the Process of Software Architecture evolution is shown in Figure 2.

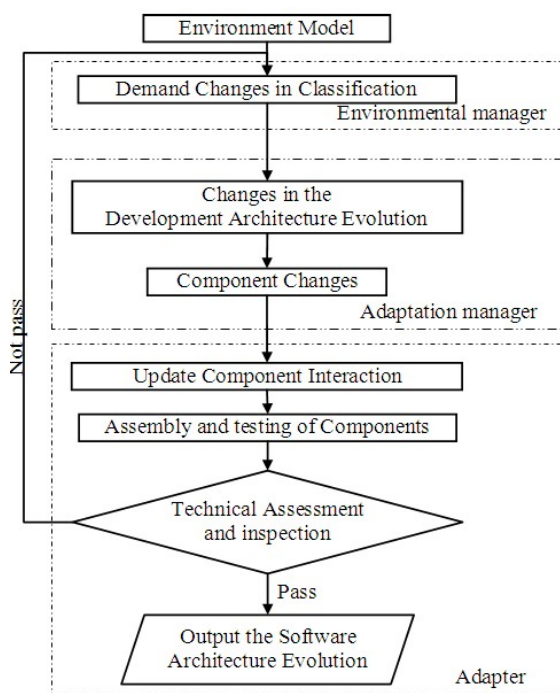


Figure 2. Process of Software Architecture evolution

In the process of software architecture, when demand or environment changes, these changes need to be classified, and evaluate by the Environmental manager of the evaluation model, then form the information of environment changes. Then based on these change of collations, developing software architecture evolution plan, doing assessment analysis by using Adaptation manager, and deciding the final component changes, forming variations of the software architecture. Finally, based on the variant to update interaction of components, then assemble and test new software components. Then use the adapter to do evaluate, form new instance of architecture model in the end.

Under the good architecture criterion, this evolution and implementation of software architecture can be more resilient to the needs and the requirements of software design based on software service provided by internet and network environment's change. Therefore, we consider this good software architecture should be as one of the guiding principles and quality evaluation standards of software design.

IV. TRUSTED COMPUTING SHOULD BE ONE OF THE SOFTWARE DESIGN AND QUALITY EVALUATION PRINCIPLES

Software design follows the method of Software of Trusted Calculation. Credibility in ISO / IEC 15408 standard: Components involved in the Computing; The operation or procedure is predictable in any condition; Withstanding viruses and physical disturbance. In here, we consider we should emphasize two aspects:

First, The design, must be according to the request and specification manual, achieving the anticipated result by the anticipated method. Setting up the anticipated Soft-behavior

set, describing it clearly in design specification manual, offering Soft-behavior authentication codes for testing the software can be trusted or not.

Second, the meaning of credibility requires not only the accuracy and reliability of software, but security is also an important indicator of software quality. Request from the software at all stages throughout the life cycle to take measures to ensure the credibility of the software.

It is because that the Trusted Computing includes five core technologies: authentication key, security output and input, memory shield / protected execution, package storage, remote attestation. Those five core technologies are indispensable conditions of a fully trusted software system. Thus, a good software design should include the five cores of Trusted Computing technologies and development methods must meet the requirements of Trusted Computing. Specifically, existing in the following areas:

(1) Development of Trusted Computing method requires safe input and output of design software. That means it must make sure the credibility of authentication code of users' interactions in the protection of hardware and software and channel of validation.

(2) Memory-Shield / Perform - Protected requires the design can isolate the sensitive areas of memory, can shield illegal access which including the operating system, to ensure data security in memory.

(3) Package-storage requires the design software to protect the data by using the authentication key that comes from using the software and hardware.

(4) Trusted Computing requires the design software should be able to remotely prove that system has ability to perceive the changes of users' computers and uses the certificate to verify hardware and software to ensure system security.

The safe Computing circumstance requires all parts of system are security and Trusted, varying from hardware layer to soft layer. Any component which is disable to prove the security and credibility of its, such as malicious code, virus, Trojans and etc, is excluded from this system. From the core technologies of Trusted Computing, it can be observed that Trusted Computing is a rigorous architecture from hardware to software, so its architecture standards can be used in software development to measure the security and reliability. Meanwhile, the development of Trusted Computing requires modules or interfaces, from hardware to software, to meet the requirement of the Trusted Computing core technology. Therefore, it ensure credibility and healthiness of System by using Trusted Computing to instruct building. So, designing software with the development by using of Trusted Computing provide strong support for system in software layer. That is why the Trusted Computing Software Development can be used as a criterion for evaluating software.

V. CONCLUSION

Software design guidelines and quality metrics, both as software engineering major theoretical issues, but also a highly

practical guidance problem. Design quality assessment is important because it will enable our software team before the software can correct errors, can solve the problems and costs are low. Guiding principles for software design, we present the development of software applications, software, product diversification, technology, software development progress of the new reality, in theory, to take not only to inherit the existing technology base, but also with the times progress and development, improvement and innovative approach, the paper textbooks in the analysis of the current problems existing in the further design of software for now focus on more detailed discussion of the issue, as a guiding principle to adapt the current software design and software design quality provide a basis for evaluation criteria. In the proposed software design quality evaluation of proposed standards, the emphasis on good software architecture and reliable method. The quality of the software architecture directly affects the quality of software design, good architecture is a guarantee of success of software design. As one of the standards of software quality testing, security, public concern, a good security software environment, you need strict requirements in software design and all links

are trusted, so that the software has credibility, as the software security to provide protection.

REFERENCES

- [1] Wang Changyuan , Li HuiJin. Software Engineering. Xi'an: Xi'an Map Press, 2003.11
- [2] Wang Changyu. Software Engineering. Beijing:Tsinghua University Press , 2004.07
- [3] Wen Bin, Liu Changqing, Tian Yuan. Software Engineering and Software Documentation Writing. Beijing:Tsinghua University Press, Beijing Jiao Tong University Press , 2005.08
- [4] Qian Leqiu, Zhao Wenyun, Niu Junyun. Software Engineering. Beijing:Tsinghua University Press, 2007.03
- [5] Xiao Ding, Wu Jianlin, Zhou Chunyan, Xiu,Jiapeng. Software Engineering Models and Methods. Beijing University Of Post And Telecommunication Press, 2008.03
- [6] Zhou Zhenliu, Xu Rongsheng. Research on Architecture and Technologies of Trusted Computing [A].In: Thesis of the Thirteenth National Nuclear Electronics and Nuclear Detection Technology Annual Meeting . 694~697
- [7] Roger S.Pressman , Software Engineering : A Practitioner's Approach, Sixth Edition. The McGraw-Hill Companies Inc. , 2006