Research on System Engineering Method of Equipment Software Quality Management

Qiang Wang¹, Qiang Li ¹ *, Jingwei Shang¹
¹Software Quality Engineering Center, CEPREI, GuangZhou, China

*Corresponding author e-mail: liqiang@ceprei.com

Abstract. Taking the limitations and challenges of current equipment software quality management into account, this paper proposes a system engineering method for equipment software quality management based on the characteristics and development trends of equipment software. Based on the modified Hall model, this paper defines the system engineering of equipment software quality, analyzes and studies the elements, relationships, boundaries and environment of the system from the perspective of software reliability engineering. Finally, the system engineering level of equipment software quality management is quantitatively evaluated by system stability and system effectiveness.

1. Introduction

The war of the future is an informationized war. The precise aggression, system interconnectivity, information interoperability of weapon systems have become the development trend of weapons equipment [1]. The equipment software development process is a complex system engineering, especially the Information Command and Operational System, which aims to integrate the C4I network to a high degree. The Information Command and Operational System is one of the most complicated engineering systems in the modern military, which will be successful developed through the cooperation of hundreds of people. The equipment software development process faces enormous challenges in technology, management and institutional mechanisms. At present, the company has established a quality management system through the certification of ISO 9001 and GJB 5000A. However, with regard to the development of large-scale complex equipment systems, the software quality management needs top-down system design. Developing and maintaining the equipment systematically standardized quantitatively is a necessary guarantee of improving the quality and reliability of software.

In recent years, many countries with more developed software technologies, especially Europe and the United States, have conducted the thorough research on software quality management. Considering the requirements of large-scale complex systems are difficult to quantify and the risk changing with scenarios, tasks, battlefields, threats and emerging technologies, as for the system development and test management, the National Aeronautics and Space Administration (NASA) adopts incremental approach to multi-state identification [2], and the method is gradually introduced into China's equipment research and development (R&D) system. The United States Department of Defense

¹ *Foundation Items: 1. 2018 Industrial Internet Innovation and Development Project, Industrial Internet safety standards system and test environment construction project.
explicitly issued the Integrated Test and Evaluation (IT&E) in DoD 15000. 2, which was released in December 2008. IT&E demands all the relevant organizations (especially the development and operational test appraisal agencies, including contractors and government organizations) work together to plan and implement the various test phases and test activities. IT&E aims to provide shared data for independent analysis, identification and reporting [3].

At present, the level of software quality management in equipment development has improved. However, there are still many problems, which mainly includes the following aspects, 1) The overall relationship of quality activities at various stages of equipment software development has not been proved in detail, the relevance and inheritance of quality activities at various stages are not strong, and quality management is more discrete and chaotic. 2) The relationship among various quality management processes is a static framework structure, rather than an operational structure which has interdependence, interaction, mutual incentive, complementarity and mutual constraint. 3) The systematic overall characteristics of equipment software quality management are not clear, and the overall characteristics formed by multiple quality management processes are not obvious. The most typical example is software reliability engineering. 4) The process of generating and accumulating activity data is passive, the collected data is static, and the information is isolated.

Taking the limitations and challenges of current equipment software quality management into account, this paper proposes a system engineering method for equipment software quality management based on the characteristics and development trends of equipment software. Based on the modified Hall model, this paper defines the system engineering of equipment software quality, analyzes and studies the elements, relationships, boundaries and environment of the system from the perspective of software reliability engineering. Finally, the system engineering level of equipment software quality management is quantitatively evaluated by system stability and system effectiveness.

2. System Engineering and Hall Model

System engineering and system science is the most influential comprehensive basic discipline in the contemporary world. Qian Xuesen, a famous scientist in China, creatively puts forward the system science thought, which is "System engineering is a scientific method of planning, researching, designing, manufacturing, testing and using. And it is a universal significance for all systems" [4]. The core of the thought is to deal with large complex systems with systemic ideas and methods combined with qualitative or quantitative. The basic methods of system engineering are system analysis, system design, and comprehensive evaluation of the system. Specifically speaking, it means using mathematical models and logical models to describe the system, simulating the operation of the system to obtain the optimal combination of the system and the best operating plan. But the shortcoming of this analysis method is that the analysis is getting more and more detailed, and the interrelationship among things is easy to be separated.

With regard to the main problems of traditional system engineering, Hall proposed a three-dimensional shape analysis model of system engineering [5]. In 1974, the US Department of Defense officially released the system engineering standard MIL-STD-499A based on the Hall model. The Hall model is a three-dimensional shape analysis model for solving system engineering problems. The three-dimensional model of the Hall model includes time dimension, logic dimension and knowledge dimension. Time dimension refers to the whole process of systematic engineering activities arranged in chronological order from the beginning to the end. The logical dimension represents the work content and the thinking procedure to be followed in each phase of the time dimension. The knowledge dimension is a subject area in which the two-dimensional system engineering method consisting of “time dimension—logic dimension” can be applied according to the degree of formalization or structuring in the mathematical sense. The time dimension and the logical dimension combine various time phases and logical steps to form a so-called system engineering activity matrix, which is an effective tool for system analysis and design, and provides a unified solution for solving the planning, organization and management problems of large and complex systems. Therefore, the Hall model is widely used.
3. System Engineering of Software Quality Management

To study equipment software quality management from the perspective of system engineering, it is necessary to define the system engineering of equipment software quality management. System engineering method of software quality management is based on system engineering theory, software engineering theory and management requirements of equipment software development, along the path of logical reasoning, analyze the process elements, technical elements, data elements and management elements, carry out quality management of equipment software systematically holistically and collaboratively. Thereby guiding equipment software development to achieve the purpose of solving software quality problems, then develop into the modern system engineering of software quality management.

Drawing on the Hall model, the system engineering of equipment software quality management is mainly oriented to six stages, which include prototype stage, initial stage, positive stage, state identification, column setting, use and maintenance. For each stage of the software life cycle, the whole process quality management of basic, technical and organizational categories is carried out to improve the quality of equipment software. The Hall model of system engineering of software quality management is shown in Figure 1.

![Figure 1. The Hall model of system engineering of software quality management](image-url)

The Hall model of system engineering of software quality management replaces the original "knowledge dimension" of the Hall model with "cognitive dimension", giving the third dimension of the Hall model a new business value. "Cognitive dimension" represents the increasing order of the intellectual hierarchy of people and organizations, reflects the cognitive process of the subjective world transforming the objective world, covers the original scene of "knowledge dimension" and the evolution of this dimension application. The new "cognitive dimension" records the cognitive process of the subjective world and the results of the transformation of the objective world. The understanding of the artificial physics system is deepening and accumulating to form the cognitive flow of "data → information → knowledge → wisdom", which means the dimension of the capacity building of the equipment software development system. The application of IT tools and platforms such as PDM,
knowledge management/knowledge engineering and technology management is an extension and enhancement of the equipment software development and management capabilities.

We can describe the equipment software quality management as such a system: \( S = \{A, R\} \), \( A \) represents the set of all elements in \( S \). \( R \) represents a collection of all relationships between elements. There are no independent elements relative to \( R \) in \( A \).

\[
A = [A_1, A_2, \ldots, A_n] \quad (1)
\]

\[
A_i = [a_{i1}, a_{i2}, \ldots, a_{in}] \quad (2)
\]

For pure software equipment, the boundary of quality management is very clear, but for embedded equipment software, the definition of the boundary is more difficult, and the quality management of the system crosses each other, mutual penetration and mutual dependence. The division of boundaries affects the closure or openness of the system. From the perspective of engineering practice, the open system has good flexibility and strong adaptability; the closed system is relatively simple and easy to control, but difficult to adjust. Considering that our project is still in the exploration stage, closed systems should be used as research objects. Software quality engineering is a very large system engineering. For the sake of simplicity, it is analyzed from the perspective of software reliability engineering subsystem.

4. System Engineering Method of Software Quality Management for Engineering Subsystem of Software Reliability

The existing software reliability engineering study mainly focus on the research of processes and methods. In the standard of IEEE 1633-2016 “IEEE Recommended Practice on Software Reliability”, the local characteristics and practice method of each stage are described according to the process of software reliability engineering, at the same time, a systematic view of the software reliability process is proposed [6]. In contrast to this, this article focuses on the research from the perspective of systems engineering. Based on this standard, the software reliability engineering is expressed as follows:

\[
A : \text{The subsystem engineering of software reliability} = \left[ a_{i1} : \text{System definition} \right.
\]

\[
\left. a_{i2} : \text{Defect importance definition} \right] , \left. a_{i3} : \text{Risk analysis} \right]

\[
\left. a_{i4} : \text{SR tool review} \right] , \left. a_{i5} : \text{Data collection system analysis} \right]

\[
\left. a_{i6} : \text{Develop a SR plan} \right) \quad (4)
\]

And so on, get the elements of \( A_2 \sim A_n \).
To study the Systematic of software reliability engineering from the perspective of system engineering, we must study the overall characteristics of the system, such as diversity and integrity [7]. The diversity of the system means that the system is the unity of diversity and the unity of differences. The system must have intrinsic correlation or coherence. Correlation is also an important source of the system's "living power". These two characteristics determine another important feature of the system, such as integrity. The system is a unified consisting of all its components, with an overall structure, overall characteristics, overall behavior, and overall functionality. In order to reflect the inherent relevance of the system and establish the integrity of the system through relevance, it is necessary to establish a relationship model by defining "relationship R".

The relationship R can be divided into three categories, 1. The relationship with the outside of the system, denoted as $R_o$; 2. The relationship between the elements inside the system, denoted as $R_i$; 3. The relationship between the element and the data, denoted as $R_d$.

\[
R_o = \{ \text{Local impact, Overall impact} \} 
\]

\[
R_i = \{ \text{Association, Dependence, Integration, Sequence} \} 
\]

\[
R_d = \{ \text{Input, Output} \} 
\]

Association: Indicates that some attributes of two elements are related. For example, A is associated with B, which is denoted as: $A \rightarrow B$. 

Figure 2. The subsystem of software reliability engineering

The subsystem of software reliability engineering includes planning, establishing a fault model, implementing during the test phase, implementing during development, implementing during release phase, and release decision support.
Dependency: One of the elements changes, it will affect the change of another element. For example, A affects B, which is denoted as: $A \rightarrow B$.

Integration: A combination of multiple elements determines the characteristics of an element. For example, A, B, and C are summarized to D, which is denoted as: $A/B/C \rightarrow D$.

Sequence: There is a logical order relationship between elements, for example, A is in front of B, which is denoted as $A \rightarrow B$.

Software quality management can be divided into the following stages according to the system process: problem analysis, quality objectives definition, software quality system integration, software quality system analysis, program optimization and program implementation. It can also be classified according to software quality characteristics, Among which the subsystem engineering of software reliability and The subsystem engineering of software security are especially important. At the same time, the system engineering of software quality management can also be divided into the following stages according to quality management techniques: process review, software testing, quality analysis, evaluation and forecasting. The quality management of equipment software is not an independent software process. It is integrated with the system engineering closely, and it determines the complexity of the equipment software quality management process, the coupling with other processes, and the sensitivity to the external environment. Therefore, The definition of environment and boundaries is critical to the quality management of equipment software.

Considering environmental factors, under certain environmental conditions, the system can adapt to the environment and form a stable environment-dependent relationship only if it shows a specific integrity. The environment is defined as System S and Environment E, which refers to a collection of transactions other than S that have non-negligible links with S, ie

$$E = \{x | x \in S \& x \text{ has a non-negligible connection with } S\},$$

where $x_i$ usually means:

- $x_1$: Equipment system requirements
- $x_2$: Technical indicators of the equipment system
- $x_3$: The importance of the system
- $x_4$: Resource conditions (progress, funding, personnel)
- $x_5$: System usage environment
- $x_6$: Model Management Requirements
- $x_7$: Equipment development management program
- $x_8$: test planning

... $x_n$: Other unanticipated environmental factors

The impact of the environment on the system can be local or holistic. It can be discrete or continuous. For continuous effects, we can adopt partial differential equation to calculate the impact of each environmental factor on the system. The greater the impact, the greater the potential improvement opportunities. Taking the planning sub-set of software reliability engineering as an example, through the analysis of the three relationships of $R_x$, $R_i$ and $R_j$, the intrinsic correlation and external connection of the system are determined to provide a basis for quantitative evaluation of system engineering of software quality management through modeling, as shown in Figure 3.
Figure 3. The planning sub-set of software reliability engineering

5. Quantitative Evaluation of System Engineering Level of Equipment Software Quality Management

The complexity of system engineering level evaluation lies in the establishment of evaluation index system. The difficulty in establishing index system lies in the scope of trade-off. In January 2010, the Guide to System Engineering Leading indicator (Second Edition) was jointly released by the International Systems Engineering Council, the Massachusetts Institute of Technology (MIT) Lean Promotion Association, the Practical Software and Systems Measurement Association, and the MIT Systems Engineering Advance Research Association. The guide contains 18 system engineering leading indicators [8]. Compared with the current system engineering measurement tools, the system engineering leading indicators can provide predictions of future development trends by comparing historical information, help to identify weak links, and take timely measures to promote the project to a final success. The indicators in the evaluation system are composed of three parts: the expected status, current status and analytical forecast. And the indicators not only contain technical indicators, but also integrate with management indicators. It is comprehensive and is an overall evaluation of the system that shows a specific integrity.

5.1. System Effectiveness Evaluation

For the moment, the systematic research and practice of equipment software management is not yet mature, and its systematic evaluation should focus on the key indicators, which includes the effectiveness of data collection, accuracy of evaluation, quality prediction ability and quality trend. However, the more comprehensive the evaluation indicators, the increase in the number of indicators will inevitably have a certain correlation between the evaluation results of each index, which leads to too many repeated indicators, and can not correctly and objectively reflect the reality [9]. The
selection and establishment of the evaluation index system is an important basis for comprehensive evaluation, and it is the guarantee for effective evaluation of the system. The core of the selection of the evaluation index system is to ensure the mutual independence of the indicators.

There are \( n \) evaluation indicators, the index vector is \( X = (x_1, x_2, \ldots, x_n) \), and the correlation coefficient of the two indicators is calculated by using the covariance.

\[
\gamma(x_i, x_j) = \frac{\text{Cov}(x_i, x_j)}{\sqrt{Dx_iDx_j}}
\]

The larger \( |\gamma(x_i, x_j)| \), the higher the correlation between \( x_i \) and \( x_j \); Conversely, the lower the correlation.

The correlation coefficient matrix of \( n \) indicator systems:

\[
R = \begin{bmatrix}
  r_{11} & \cdots & r_{1n} \\
  \vdots & \ddots & \vdots \\
  r_{n1} & \cdots & r_{nn}
\end{bmatrix}
\]

5.2. System Stability Evaluation

Stability is an important survival mechanism of the system. The better the stability, the stronger the system's ability to survive. The signs of the quality system engineering have reached a relative stable status, which is mathematically characterized by fixed points, expressed as the average value of a certain stage as a point. For quality engineering systems, fixed points should consider changes in demand, review related data, number of software defects, software reuse capabilities, software reliability indicators, development costs, user feedback (calculated by KLOC, that is, thousand lines of code), etc. At these points, the derivative of all state variables is less than a certain threshold, that is, the system is considered to be stable. The focus of system stability is to solve transient problems and reduce the impact of transients on the overall system. The calculation method of system stability is as follows:

Instantaneous rate of change of fixed point at a certain time:

\[
\overline{x}_t = \frac{\partial X}{\partial t} (t \in [t_1, t_2])
\]

Average rate of change of fixed point at a certain time:

\[
\overline{x}_t = \frac{1}{T} \int_{t_1}^{t_2} \frac{\partial X}{\partial t} dt, (t \in [t_1, t_2]) \leq \varepsilon
\]

Considering time dimension, logic dimension and knowledge dimension, the stability of the system can be analyzed from the three-dimensional space:

\[
\overline{x}_t = \frac{1}{T} \int_{h}^{\infty} \frac{\partial X}{\partial t} dt, \overline{X} + \frac{1}{T} \int_{h}^{\infty} \frac{\partial X}{\partial t} dt, \overline{Z} + \frac{1}{T} \int_{h}^{\infty} \frac{\partial X}{\partial t} dt, Z \leq \varepsilon
\]

If there is a mathematical relationship \( f(x, x, \ldots, x) \) between the selected fixed points, this relationship can be used as a fixed point for comprehensive consideration.

6. Conclusion

This paper presents the system engineering method of equipment software quality management, which realizes the correlation and inheritance of quality activities in each stages of equipment software development. The data flow and information flow between quality activities are continuously interacted and iterated, and closely linked. The operational structure of interdependence, interaction, mutual incentives, complementarity and mutual restraint relationships runs through the entire system.
engineering. By improving the system effectiveness and stability, the viability of the system can be improved, reduce the risk of system operation by locally adding new technology applications, increasing detection or inspection methods, increasing test density or other quality management activities.

7. Acknowledgments
The authors would like to thank the anonymous reviewers for their valuable comments to improve the quality of study.

References
[1] G. S. Wang, G. Luo. Analysis and enlightenment of the US army’s integrated test and evaluation,” J. of the Aca. of Equi. Com. & Tech., vol. 21, pp. 95-98, 2010.
[2] M. R. Woodward and M. A. Hennell, “Strategic benefits of software test management: a case study,” J. of Eng. and Tech. Management, vol. 22, pp. 113-140, Mar. 2005.
[3] Department of Defense. DoD 15000. 2, Operation of the Defense Acquisition System, Washington: Department of Defense, 2008, pp. 50-56.
[4] X. S. Qian, On System Engineering, Shanghai Jiaotong University Press: Shanghai, 2007.
[5] H. B. Duan. “The evolution of traditional system engineering to modern system engineering from the evolution of Hall model,” Intel. Manu., vol Z1, pp 88-95, 2016.
[6] IEEE Std 1633-2016 (Revision of IEEE Std 1633-2008), “IEEE Recommended Practice on Software Reliability,” IEEE, Jun. 2018.
[7] G. Z. Xu, J. F. Gu, H. G. Che. Systems Science, Shanghai Science and Technology Education Press: Shanghai, 2000.
[8] H. B. Pu, K. S. Wang, J. H. Yuan, W. Man, and J. F. Sun. “Eighteen system engineering leading indicator ——a new tool for evaluating the effectiveness of system engineering,” Aero. China, vol. 01, pp. 34-37, 2013.
[9] S. Shi and W. Y. Liu, “Factor analysis method research of evaluation index connection,” Math. in Eco., vol. 27, pp. 23-27, 2010.