

Optimization of Software Project Evaluation System

DING LI¹

(Received December 7, 2019; Revised February 13, 2020; Accepted March 19, 2020)

Online First Publication

DOI: 10.33317/ssurj.146

Abstract— Starting from the problems of imperfect evaluation index system, non-standard evaluation method and single factor of performance evaluation model, this paper applies the unified grey clustering analysis algorithm to establish the index system of software organization status and software project characteristics, i.e., defines the connotation of software project performance. A new network topology design method is proposed and a software project performance evaluation model based on fuzzy neural network is established. An improved particle swarm optimization algorithm is introduced to solve the problem of determining the connection weight coefficient of the evaluation model accurately and efficiently. This paper introduces the basic concept of project performance evaluation and explains the necessity and importance of performance evaluation in the project.

Index Terms— Grey Clustering Analysis, Software Project, Performance Evaluation, Fuzzy Evaluation, Spiral Assessment Model.

I. INTRODUCTION

With the increasing scale and complexity of software projects, software projects needs to be efficiently managed. Simply relying on software engineering technology cannot solve all the problems in the process of software projects. In the era of information economy affairs are unique and no longer simple repetition. The information itself is, constantly changing, dynamic and flexible. Project management is precisely the key means to achieve flexibility and project management in the mode of operation to maximize the use of internal and external resources, i.e., fundamentally improving the efficiency of middle managers. People gradually realize the advantages of the project management mode and adopt it one after another, which has become an important means of enterprise management. By the start of 1980s, project management has been recognized as a viable and effective way to achieve complex enterprise goals and has become a research hotspot in the field of management [1]. "It is impossible to manage without measure," which illustrates the importance of evaluation in management [2]. Software project performance evaluation is a part of software project management and should be included in the whole process. Therefore, project performance evaluation can be divided into pre-evaluation, in-process evaluation and post-evaluation. In the actual operation process, enterprises often only do post-evaluation, or even do not do evaluation at all, so that the project performance evaluation is not included in the whole project management work. This approach makes the software management imperfect in the development process and thus affecting the final software quality.

II. GREY CLUSTERING EXTRACTION AND ANALYSIS ALGORITHM

Software development and other venture assets development first need to address the expected performance and expected risk of these two core issues. Therefore, we use grey clustering analysis theory to forecast. The theoretical basis is that the probability distribution of securities performance in a certain holding time is the basis of every development choice. The developers estimate the portfolio risk according to the expected performance rate of the securities [3]. The developer's decision is only based on the expected performance and expected risk of the securities. At a certain level of risk, the developer expects the greatest performance, i.e., corresponding to a certain level of performance the developer expects the least risk. According to the above assumptions, grey clustering analysis establishes the calculation method of portfolio expected performance, risk and effective boundary theory and establishes the mean-variance model of asset allocation optimization. The model is applied to the evaluation of overall project performance, which can be expressed as follows:

$$\min \delta^2(R_p) = \sum \sum X_i X_j \text{Cov}(R_i - R_j)$$

Where:

R_p = The performance of project portfolio

R_i = The performance of i project

X_i and X_j = The development proportion of i and j

$\delta^2(R_p)$ = Variance (combined total risk) for portfolio development

$\text{Cov}(R_i - R_j)$ = Covariance between two items

This model laid the foundation for modern securities development theory. The model shows that solving the X_i project performance rate under the restrictive condition makes the portfolio risk minimization i.e., $\delta^2(R_p)$. It can be obtained by Lagrange's objective function. The significance is that the developer can pre-determine a desired performance through the model and can determine the proportion of developers in each development project, so that the total development risk is minimized. Different expectation performance has different minimum variance combinations, which constitutes the minimum variance set [4]. It provides a good means to accurately measure the risk and performance of securities development projects, but this model involves calculating the covariance matrix of all assets. Facing hundreds of optional assets, the complexity of the model restricts the practical application.

¹Xi'an Aeronautical University, Xi'an, Shaanxi China. 710077. liding900712@126.com

III. OPTIMIZATION OF SOFTWARE PROJECT RISK PERFORMANCE EVALUATION METHOD

Combined with the previous grey clustering extraction algorithm, the structure of software project evaluation system is optimized. Firstly, a software system operation project evaluation system is established by combining clustering analysis algorithm. By calculating the matrix, the distribution of load points in the software system operation system is obtained. Through the analysis of characteristic data, the failure probability of power cut and short circuit and the standard reference index of system running time are calculated, so as to further estimate the reliability index of running system. The specific estimation method of the reliability index of the system running is shown in the following diagram.

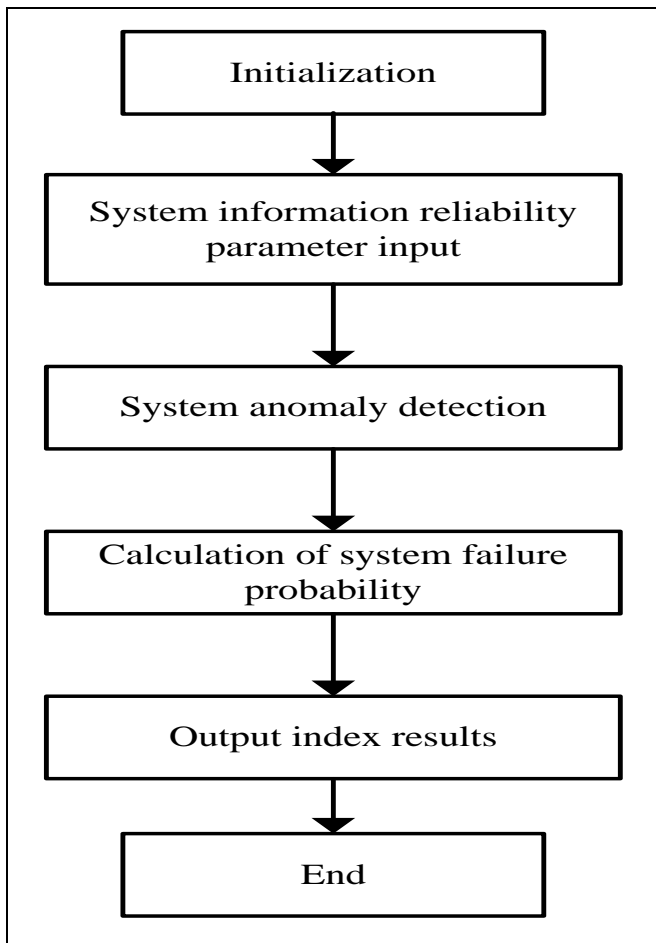


Fig. 1: Operational Reliability Evaluation Process of Software System

The software system operation reliability evaluation process is shown in the diagram i.e., Fig. 1. In the process of evaluating the system operation reliability, three-dimensional point clustering algorithm is used to calculate and evaluate the software information of each module of the software system operation. Combined with the above calculation and analysis method, the software structure reconfiguration strategy is simplified. Because the structure of project evaluation system is complex and influenced by many external factors, it is necessary to protect and restrict the system in the process of system operation structure optimization. The current

information search method is used to track the system capacity and the shortest available system reconfiguration path is detected [5] along with manual switch, so as to achieve the design requirements to ensure system safety. In the running structure of software system, the shortest reconfiguration path should be judged from the optimum and the shortest security path should be selected by comparison. If there are two or more paths with the same length in the system operation structure, the manual switch can be designed according to the number of path nodes. In the system structure design, the system cable is a ring structure to form a network radiation network, so as to achieve the design effect of all-round and accurate use. The 3-point radiation structure of the system is shown in the diagram, i.e., Fig 2.

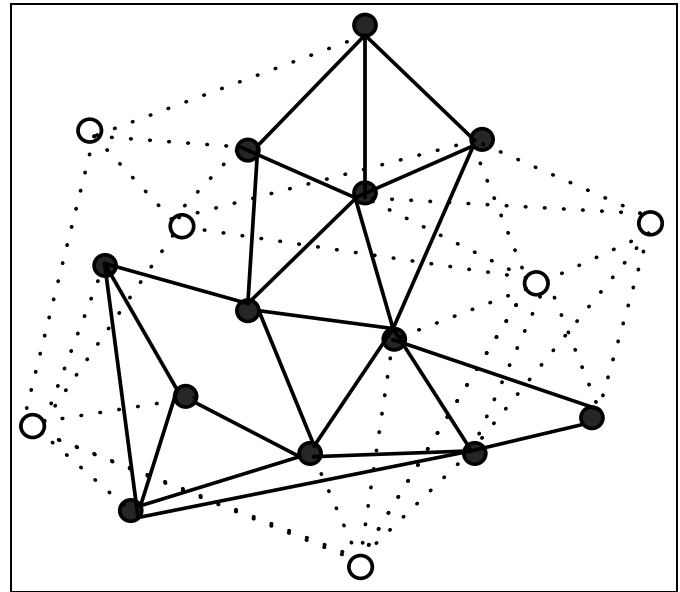


Fig. 2: Operation of 3D Point Radiation Structure in Software System

As shown in the figure, the radial structure of the ring system operation can better ensure, the safe and stable operation of the system, avoid system failure and other issues. Each of the three-dimensional points in the diagram can be directly or indirectly connected by cables to form a ring-shaped extended radiation structure, thus forming a number of project evaluation system load centers. In the three-dimensional point radiation structure of the system operation, the manual load switch and the automatic switch system of the important load project evaluation system are installed under the load center of each project evaluation system to ensure the stable operation of the two load centers and the stable connection with the distribution board of the software project evaluation system. This can achieve the design effect of dual power supply system, to ensure that the project evaluation system can still run normally when the failure occurs. In order to ensure the power supply reliability of the system operation load, the cluster extraction algorithm is used to extract information and evaluate the reliability of the software system operation system structure, which can effectively detect the safety and reliability of the software system operation structure. By analyzing the structure of software system operation system and combining with the process of software system operation reliability evaluation, it is helpful to extract and calculate the

three-dimensional data features of software system operation more accurately. Software project performance evaluation is a tedious and rigorous work, we must try our best to find the program defects, in order to avoid problems in the actual operation, causing unnecessary losses. Therefore, in the process of testing, we need to establish a detailed test plan and test cases, and strictly follow the test plan to test, in order to reduce the randomness of testing and missing problems. From the above description, we can see that a lot of work needs to be done in the whole software development process. This work not only involves personnel, technology, equipment, capital and other issues, but also related to the organizational structure, development methods and project procedures. These factors have varying degrees of impact on the software development process and the risks are different at each stage, so we have to assess them in advance to take action. The relationship between software project objectives is shown in the figure, i.e., Fig 3.

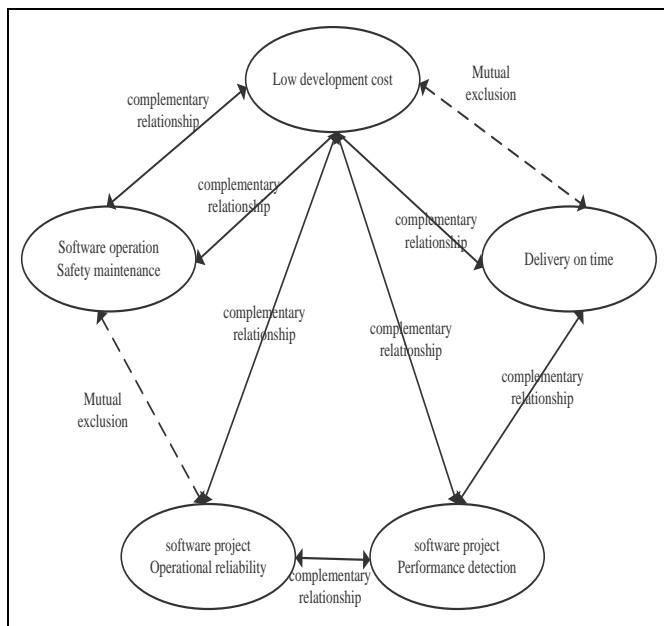


Fig. 3: Relationship Structure of Software Project Performance Evaluation

As with all types of projects, it is often very difficult to attempt to achieve the desired level of these goals in the actual development of software projects. The relationship between these goals is complementary. Obviously, low development costs usually mean a high probability of delivery on time and high-reliability software is naturally easy to maintain. The low development costs affect software reliability, and there is a conflict between good portability and software performance. Therefore, in order to achieve these goals, software engineering principles must be followed in the software development process. One of the advantages of this approach is that it can standardize the development process, reduce systemic risks and at the same time, seek a balance among these objectives, so as to maximize user satisfaction. Linear model divides software engineering activities into top-down steps:

- The development planning

- Requirements analysis and description,
- Software design,
- Program coding,
- Testing and
- Operation maintenance

Each working step links up with each other, like a waterfall, falling step by step. Throughout the development process, each phase undergoes a rigorous review and only when its work is confirmed, can the next activity continue. Therefore, the advantages of this model are that the development process is rigorous, the system design has a good global acceptance and suitable for the development of large-scale systems. The disadvantages are that it is not flexible enough to solve the problem of unclear and inaccurate software requirements; the development cycle is long; the software reusability is low and it cannot be integrated with quality management, risk assessment and other management activities. The prototype model constructs an initial prototype quickly to realize the interaction between the user and the system, then evaluates the prototype with the user and the customer, further refines the requirements of the software to be developed according to the user's modification opinions, and revises the prototype repeatedly until the customer is satisfied. Obviously, the prototype model can overcome the shortcomings of the linear model and reduce the development risk caused by the unclear software requirements. The shortcomings are that the users can mistakenly identify the prototype as the final system and lack patience and it is difficult to manage so it is suitable for small and simple system development. For complex large-scale software the development of a prototype often fails to meet the requirements. The spiral model combines the waterfall model with the prototype model, iterates several times along the spiral and develops a more complete new software version for each rotation from inside to the outside along the spiral, as shown in the figure, i.e., Fig. 4. The four quadrants of the spiral model Cartesian coordinate represent four aspects respectively:

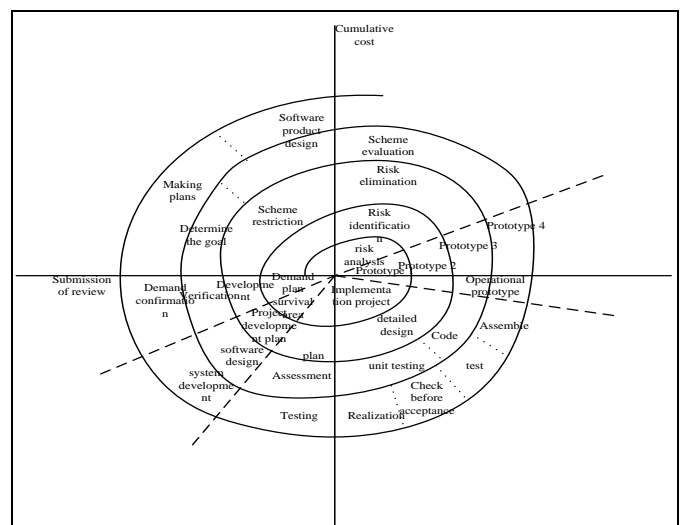


Fig. 4: Spiral Assessment Model

As shown in the figure, the spiral model emphasizes introducing a risk identification, risk analysis, and risk control program before each development phase. It is particularly

suitable for large, complex and high-risk systems but if developers do not have rich experience and expertise they cannot fully identify and assess the risks which in turn will lead to greater losses.

IV. CONCLUDING REMARKS

In this paper, the evaluation system of software system operation project is established. The optimization of software project evaluation system is realized by simplifying the strategy of software structure. In the process of system operation structure optimization, it is necessary to protect and constrain the system. The spiral model greatly improves the effect of system optimization.

REFERENCES

- [1] Ragab, A. M., & Meis, S. (2016). Developing environmental performance measures for tourism using a Tourism Satellite Accounts approach: A pilot study of the accommodation industry in Egypt. *Journal of Sustainable Tourism*, 24(7), 1007-1023.
- [2] Isaksson, L. E., & Woodside, A. G. (2016). Capturing complexity in how configurations of firm Internal Orientations impact corporate social performance outcomes: Breaking from the dominant logic of symmetric-variable to asymmetric-case-based theory and testing. *Australasian Marketing Journal (AMJ)*, 24(4), 300-308.
- [3] Meng, Q., Han, L., Xiao, Y., Li, H., Wen, S., & Zhang, J. (2016). Numerical simulation study of the failure evolution process and failure mode of surrounding rock in deep soft rock roadways. *International journal of mining science and technology*, 26(2), 209-221.
- [4] Nagaraju, S., Kashyap, M., & Bhattacharya, M. (2017). An effective density based approach to detect complex data clusters using notion of neighborhood difference. *International journal of automation and computing*, 14(1), 57-67.
- [5] Gehui, L., Jiefeng, S., & Chuguang, Z. (2000). Application of Fuzzy Logic Toolbox in Matlab Software in Simulation of Fuzzy Control System[J]. *Computer Simulation*, 17(5): 69-72.