

# Research on Software Reliability Model Prediction and Evaluation System

Ru Wang

Wuhan Huaxia University of Technology, Hubei Wuhan 430223

**Abstract.** This paper analyzes the requirements of software reliability model prediction and evaluation system, describes the overall structure of the system, the specific module functions and the design of database. This paper USES JavaScript, Js, Html and other technologies to complete the design of software reliability model evaluation system and analyzes the hierarchical structure of classes and key program packages. Finally, the test scheme is designed to complete the test of the reliability model evaluation system, including the system's function, performance, reliability, security and operability.

## 1 Introduction

In recent years, with the development of the Internet, various industries have begun to develop their own software systems by virtue of the advantages of the open, extensive and Shared Internet. Computer software technology is a key technology of informatization, and its improvement is an important factor in the development of informatization technology [1]. Computer software technology to support the major public service system, e-government system on the market, the normal operation of the financial system and other kinds of service system, and the open sharing under the network environment, the system and the product requires higher reliability and safety guarantee. In today's information age, software failure will give our own life, property, and even the society as a whole to bring serious loss, so the reliability of the software research has attracted more and more the attention of the scholars and experts.

In the process of software product and system development, the product characteristics of the industry are often sacrificed for the reliability of software system. Therefore, the research of software reliability evaluation needs to be more detailed throughout the whole process of software design, development and testing. The research on software reliability assessment is of great significance and therefore has certain challenges [2].

## 2 Software reliability overview

### 2.1 Software reliability

Reliability is the ability of a product to perform specified functions under certain conditions and within a certain period of time. Its probability measure is called reliability. In the process of software product and system development, the product characteristics of the industry are often sacrificed for the reliability of software system. Quantitative analysis of software reliability is an index to measure software

reliability. Software reliability is not just described by a measurement parameter. Software metrics can vary depending on the software properties and scope of application. In order to obtain the software failure probability more accurately, the software reliability method needs the theoretical basis of probability theory and mathematical statistics. Software reliability index is management index and technical index. Management indicators are mainly for software managers. The purpose of management indicators is to measure whether certain reliability activities have been carried out in the process of software system development, whether the reliability distribution of system functions has been taken into account, and whether specific time has been specified [3]. The technical measurement method is a scientific quantitative analysis of software reliability from the perspective of professional technology. So, in order to achieve more accurate software reliability analysis, the reliability of the system, the cumulative failure probability and the average time between failures such as evaluation index for assessment of software reliability testing is an important according to the system design and function realization of subsequent has important significance.

### 2.2 Software reliability model

Software reliability model plays a key role in predicting and evaluating software reliability, which is of great importance to ensure the overall quality of software. It can determine the time required for the software system to fail and achieve the software reliability target according to the relevant parameters of the selected model. The establishment of software reliability model can effectively predict and evaluate the fault behavior of software system. Software reliability modeling can be attributed to the statistical estimation process based on software reliability data. With the elimination of software failures, the failure rate of software system decreases, and the historical change of

software failure rate can be observed according to time. This, in turn, evaluates the reliability of the current software system and the time it will take to complete the specified goals in the future.

### 3 System design

The software reliability model has its own advantages, but it also has some limitations. In order to give full play to its own advantages and make up for its defects, a part of the model system is integrated and sorted out, and a software reliability model prediction and evaluation system is established. Its purpose is to develop a complete software reliability model predictive evaluation system with multiple models, so that the evaluation system is more practical and operable.

#### 3.1 Overall design structure

The system integrates software reliability prediction and evaluation system of several classical software reliability models. When a user logs in to the system platform, software failure can select or input data source (count failure data set and interval data sets), set up reliability prediction analysis process, selecting a model for reliability prediction model parameters have been set, and then performing the forecast analysis, and form in figures and tables show the results of the analysis [4].

Specific function description of software reliability prediction and evaluation system:

(1) The user USES the graph formalized interface for operation. Users can set the data sources (including direct import, from TXT, XLS, or CSV file to import or import) from a database, choose the reliability model, set up the model parameters, set up reliability prediction analysis process;

(2) The system provides four classical reliability models (Schneidewind model, Jelinski-Moranda model, Duane model and Goel-Okumoto model). Through the parameter design interface, users can adjust parameters and optimize the analysis results.

(3) Capable of processing data from multiple sources. Commonly used relational database: SQLserver2008;

(4) Visualization of analysis results. Can display analysis results as tables and graphs and export the results to XLS and PDF files.

The overall structure of the software reliability model prediction and evaluation system is shown in figure 1.

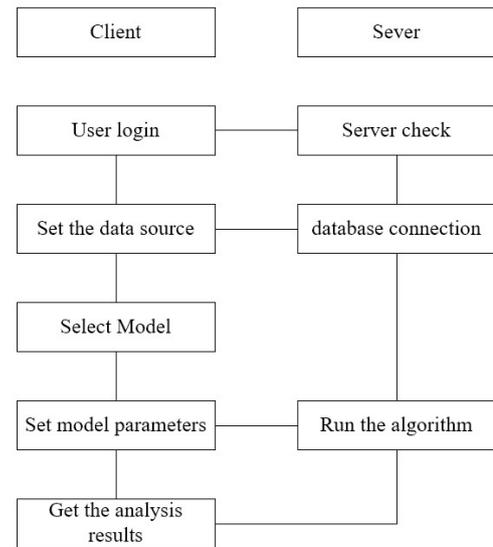


Figure 1. Overall structure of the system.

#### 3.2 Design constraints

(1) Requirements: precision requires floating point data processing

(2) Time characteristics: both update processing time and response time are less than 6 seconds; Data conversion and transmission time less than 3 seconds;

(3) Flexibility:

A. Changes in operation mode: realize an operation that can be used by any external device, such as mouse click and enter, keyboard instruction enters, touch screen touch and touch screen enter, etc. can operate the system;

B. Changes in the operating environment: it can be used on multiple platforms;

C. Change in accuracy and time limit: avoid the impact of precision change on the system and minimize the impact;

D. Planned changes or improvements: easier to improve.

(4) Input and output requirements: input items are user name, password and server address; The output items are user interfaces, graphs, or tables.

(5) Data management capability: the number of records to be managed is more than 10,000. It is divided into a number of tables and volumes; its size is about 100.

(6) Fault handling requirements: when the system failure, to avoid the loss of data, the database will be able to recovery, to ensure the consistency and integrity of data, at the same time on a regular basis to do the data backup is necessary, also requires the system can upgrade the management on a regular basis.

(7) User interface design rules

A. User interface: interface naming should be as scenario-oriented as possible to ensure interface friendliness and internal logic leakage.

B. Try to be consistent, and interface specifications should follow the software interface specifications.

C. Design a complete dialogue process: every conversation in the system should be in the beginning, middle process, and end sequence.

D. Provide a simple error handling mechanism.

E. Provide information feedback: the mouse suspension window and other methods provide users with current software status, software function interpretation and other information.

F. The overall style of the interface should be unified, simple and generous, in line with the requirements of the industry, and reasonable color collocation such as window background font.

(8) Internal interface design

Interfaces need to be established between the functional modules for invocation. Ensures access to internal interfaces from presentation. Layer interface design to the business logic layer and data persistence layer access module interface design should be bigger and bigger, in addition to public module interface design should be allowed to be any other module calls.

3.3 Design strategy

In order to ensure the scalability of the system, the software reliability model predicts and evaluates every subsystem of the system using inherited abstract design. In the algorithm design, the combined algorithm strategy model is adopted. The algorithm is packaged separately to ensure that it is independent of each other and can be replaced at will. Therefore, the change of the algorithm will not affect the customers of the practical algorithm.

Software reuse has been extended to all stages of the software life cycle, which is an important guarantee for improving software productivity. The main idea is to think of the various functions in software as components, and each component is designed by the program as a general means to accomplish the same functions. In the software reliability model predictive evaluation system all subsystems are independent reusable, forming model is encapsulated by the business logic, and each subsystem is a separate and independent can reuse components [6]. From the perspective of business logic, the business logic of the system is encapsulated as an entity, independent of the user interface and database. The purpose of this design is to realize the reuse of the complete business logic of the system and reduce the development and maintenance costs.

3.4 Design idea

Software reliability model prediction and evaluation system [7] adopts modular design. Every module in system operation process, have the ability to input and output the results of the assessment and failure data for input including failure time interval data and time interval data, through certain data conversion mechanism to evaluate the reliability of the software system and display the results. The system selects the reliability model according to the software life cycle, historical project experience, model parameter estimation and model reliability evaluation curve. The system can predict and evaluate the reliability by obtaining reliability data (such as fault data collection, software reliability model evaluation reliability, and cumulative failure probability) and mean time between failures. The data flow diagram of the system shown below is shown in figure 2.

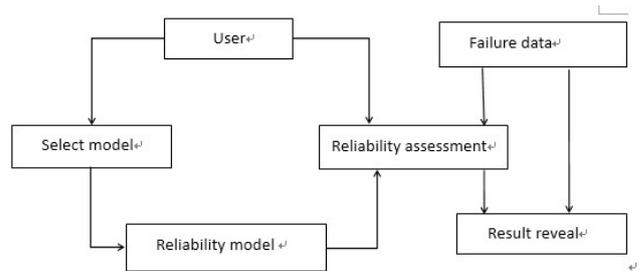


Figure 2. System data flow diagram.

3.5 Database design

The system includes User Info and evaluation result data tables. The following is the conceptual structure design and physical design of the two data tables.

(1) User information data table:

User Info User information mainly includes: User name, password, contact information, address, registration time, work and other basic information to identify a User. The e-r diagram of user information is shown in figure 3.

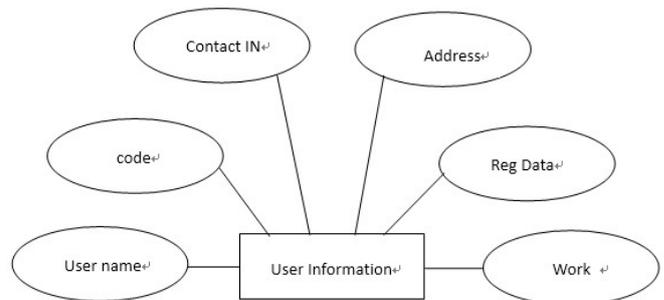


Figure 3. E-R.

(2) Data source data table: evaluation results

Data source includes user, table name, generation time, model name, EX, EY, KS, PL, model noise, and when Basic information such as forward failure efficiency, current reliability, next failure prediction, U diagram, Y diagram, prediction diagram, PL sequence, etc. Database connection class diagram DB-Connection interface, DB-Type abstract class, specific database relational graph (where MYSQL and Oracle cannot implement):

4 Test conclusion

The software reliability model prediction and evaluation system has passed the strict standard test of system function, performance, reliability, security, reliability and operability. Test results show that the software reliability model predictive evaluation system can meet the needs of the business requirements, and system architecture of advanced, simple, fully functional, system security is good, high by sex and operational characteristics, the statistical forecast model using a variety of classic prediction algorithm strict testing, predictive model to calculate the prediction of closer to the real data, the result is correct.

## References

1. Tan min, loose GUI. Development of software reliability model [J]. Journal of computer science, 1990, 33 (5): 382-390.
2. Pham h. Software Reliability [M]. Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, Inc. 1999.
3. Strom R E, Yemini s. Type state: A programming language concept for important software reliability [J]. IEEE Transactions on software Engineering, 2012, 157-171.
4. Chen Si, Wang Shiyan, Sun Jiadze. Reliability measurement model of trusted software based on cloud model [J]. Computer application research, 2014, (09).
5. Wu Jiang. Component based software reuse technology [J]. Scientific outlook, 2016, 26 (15).
6. Carma Mc Clure. Software reuse technology [M]. Machinery industry press, 2003.
7. Liu Wenbin. Research on dynamic fault tree analysis method based on modular thinking [D]. Nanjing university of technology, 2009.
8. Gen Ji, Nie Peng, Qin Zhiguang. Current situation and research of software reliability model [J]. Journal of university of electronic science and technology, 2013, 4: 565-570.