2012 International Conference on Computer Science and Service System

The Software Quality Evaluation Method Based on Software Testing

Liu Wen-hong, Wu Xin
Beijing Institute of Tracking and Telecommunications Technology
Beijing, China
e-mail: wx263@139.com

***Abstract*—In order to improve the effectiveness, visibility and specification of the TT&C software testing, this paper made an in-depth study according to its characteristics. At first, this paper presented a quality assessment model with high reliability and real-time demands getting idea from analytic hierarchy process (AHP). Then, the paper brought forward a dedicated simulation test environment and a generating method of software testing cases based on fault tree analysis (FTA). Next, the paper defined the software testing procedure learning from CMMI demands. At last, the paper gave the performance of quantitative assessment results in the form of a radar chart. Practice has proved that the given model of this paper can represent the software quality objectively, the generating method can improve the sufficiency of software testing cases effectively, and the defined procedure can ensure the specification of software testing availably.**

Keywords- Software testing, software quality evaluation, software process management

I. INTRODUCTION

With the TT&C software more widely used, the requirement for its reliability, security and real-time performance is very high. This is especially true for target tracking software, information processing and control software. In recent years, software quality problems and various risks are becoming increasingly evident. Problems caused by software defects have even accounted for 80% of all the problems. This seriously affected implementation of tasks. To this end, how to improve software quality has become a desiderated problem. Improving software quality should start from two aspects. The first is to promote the capabilities of software development teams. The second is to put software process management in practice. In the process of software validation and verification, testing is the most critical and effective means. Especially, third-party software testing will play an important role on improving software quality and reducing software defects.

In this paper, we introduce how to carry out third-party software testing and how to evaluate software quality based on third-party software testing.

II. THE PROCESS MANAGEMENT OF EVALUATION AND TESTING

According to the characteristics of the TT&C software, our process management of evaluation and testing include [1]: requirement management, project planning, project monitoring and control, configuration management and quality assurance. The activities of evaluation and testing can be divided into the following five stages: test requirement analysis, establishing evaluation model, test design (including test case design, test simulation system design and implementation), testing, software quality evaluation. The relationship between the activities and management of evaluation and testing is shown in Fig.1.

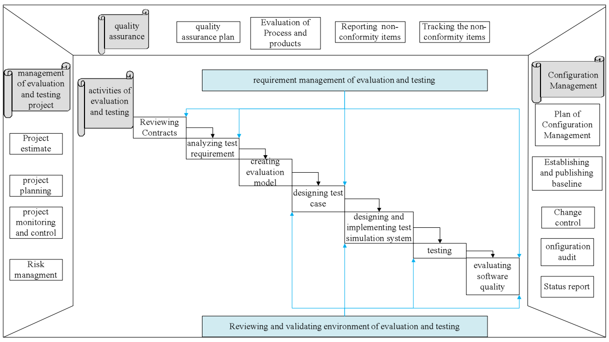
quality assurance
quality assurance plan
Evaluation of Process and products
Reporting non-conformity items
Tracking the non-conformity items
management of evaluation and testing project
Project estimate
project planning
project monitoring and control
Risk managment
activities of evaluation and testing
requirement management of evaluation and testing
Reviewing Contracts
analyzing test requirement
creating evaluation model
designing test case
designing and implementing test simulation system
testing
evaluating software quality
Reviewing and validating environment of evaluation and testing
Configuration Management
Plan of Configuration Management
Establishing and publishing baseline
Change control
onfiguration audit
Status report

Figure 1. The activities and management of evaluation and testing

A. *Requirement Management of Evaluation and Testing*

The objects of requirement management of evaluation and testing are requirements of evaluation and testing, such as the specification, contracts and technology agreements related to the software testing task, the requirement specifications, design specifications and manuals of tested software. According to the test level, the test requirement may vary.

In Practice of the evaluation and testing, the tested software may often change. This may cause many problems and often result in great loss and very serious errors. Therefore, it is necessary to effectively manage test requirements. Test requirement management includes following three steps:

1) Test requirement review. This mainly reviews the completeness and consistency of test requirements. In order to reach consensus on the test requirements, it is necessary to invite the Consignor attending the review.

2) Establishing the test requirement baseline of evaluation and testing projects. The activity can control the test requirements and its change.

978-0-7695-4719-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CSSS.2012.369

IEEE computer society

3) Establishing the traceability matrix of test requirements. The tracing relationship to test requirements should be established in very stage of evaluation and testing.

B. Project Planning of Evaluation and Testing

Project planning of evaluation and testing include:

1) Project estimate: It includes estimate to project size, workload and cost. We can use the number of test items, the number of test cases and the lines of test program to estimate the project of evaluation and testing.

2) Schedule: according to the estimates to schedule.

3) Risk analysis and Evaluation: identifying project risk events. This includes to assess the impact, probability of occurrence and urgency of risk events, to determine the priority of risks and to develop mitigation metrics and disposal strategies.

4) Personnel planning: estimate the staff the project needs. the staff includes: the testing team, the quality assurance team and the configuration management team.

5) Resource planning: planning the resources required for the project, including the resources which are used in test implementation, quality assurance and configuration management.

The planning runs throughout the entire life cycle of the project. The product of planning mainly is evaluation and testing plan (including risk analysis and assessment, staff and resource planning, etc.).

C. Monitoring and Control to the Evaluation and Testing Project

The purpose of Monitoring and Control to the Evaluation and Testing Project is to monitor and control the actual progress of the project so that we can take effective measures if any major deviations occur in the actual progress. The plan of monitoring and control includes the composing of staff, how to monitor and the frequency to monitor. The content can be written to the plan of evaluation and testing.

The monitoring can adopt following forms: daily, weekly and weekly regular meetings. The activities include monitoring workload, schedule, risk and resource. In addition, the validity of evaluation and testing tools should also be tracked. At end of the project, in order to obtain the actual size of the project for follow-up projects to refer, we should analyze tracking data.

D. Configuration Management of Evaluation and Testing

The purpose of Configuration Management of Evaluation and testing is to effectively control the configuration changes, to ensure configuration integrity, accuracy, consistency and traceability.

The main activities of configuration management include: configuration identification, configuration change control, configuration status statistics and configuration audit. Configuration management runs throughout the entire life cycle of the project of evaluation and testing.

E. Quality Assurance of Evaluation and Testing Project

The purpose of quality assurance of evaluation and testing projects is to ensure that the process being used and the work products being constructed comply with the corresponding requirements and standards.

The activities of quality assurance of evaluation and testing projects include:

1) Developing quality assurance plan of evaluation and testing projects.

2) Auditing whether the process and products comply with the used standards, procedures and requirements.

3) Recording audit results to generate quality audit reports.

4) Tracking the non-conformity items found, until the non-conformity items to be properly dealt with.

III. Establishing Evaluation Model

According to the characteristics of the TT&C software, which are high reliability, security and real-time, referring to responding standards, we establish the quality evaluation model of TT&C software. The model defines comprehensive evaluation factors [2], including: software quality characteristics and the degree of software engineering. According to software quality characteristics and sub-characteristics which are defined in the model , each quality requirement of tested software should be concretely analyzed to determine the characteristics and sub-characteristics which should be tested and to develop metric items of each sub-characteristic. When establishing the model, We sufficiently considerate concrete characteristics of tested software to enhance real-time, reliability and security testing. We also invite experts in the field to determine the weight coefficients using a 0-4 score.

The quality evaluation model of TT&C software is shown in Table 1 (in the last page of this document). Using the 0-4 scoring method, we determine the comprehensive evaluation factors, software quality characteristics, software quality sub-characteristics and the weight coefficients of metric items. Among them, the weight coefficient of the software quality characteristics is 0.75, the weight coefficient of software engineering is 0.25.

The progress of software quality evaluation is following:

1) Using (1) and test results, we can calculate the values of metric items, characteristics and characteristics respectively from which we can obtain comprehensive evaluation value of quality characteristics.

$$Vs_{ij} = \sum_{k=1}^{N_{mij}} \omega_{mijk} \cdot v_{ijk} \qquad (1)$$

$$V_i = \sum_{j=1}^{N_{si}} \omega_{sij} \cdot Vs_{ij}$$

$$V = \sum_{i=1}^{N_q} \omega_i \cdot V_i$$

Where, V is the comprehensive evaluation value of quality characteristics, V_i is the metric value of corresponding quality characteristic, V_{sij} is the metric value of corresponding sub-characteristic, v_{ijk} is the metric value of corresponding metric item, N_q is the count of characteristics, N_{si} is the count of corresponding sub-characteristic, N_{mij} is the count of corresponding metric item, ω_i is the weight coefficient of corresponding characteristic, ω_{sij} is the weight coefficient of corresponding sub-characteristic, ω_{mijk} is the weight coefficient of corresponding metric item.

2) The comprehensive evaluation value of software engineering level is calculated using (2):

$$C=\sum_{i=1}^{N_c}\lambda_i C_i \qquad (2)$$

where, C is the value of software engineering level, N_c is the count of all check items of software engineering, λ_c is the weight coefficient of corresponding check item, C_i is the value of the i-th check item.

3) The comprehensive evaluation value(CEV) of tested software is calculated using (3):

$$CEV=0.75\times V+0.25\times C \qquad (3)$$

IV. Test Case Design

In order to obtain accurate data which evaluating software quality needed, we should carry out the necessary and sufficient tests should be carried out. We use fault tree analysis method, equivalence class partition method and boundary value analysis method to design test cases. Fault tree analysis method was first used for system reliability analysis and design. In the designing test cases for failure mode decision software, we adopted this method, which greatly improved the scientificity of test case design and reasonableness of test case distribution.

The process of fault tree analysis method [3] is following: carefully and thorough to analysis software system, to write down the top events, which will possibly cause failure, to the top rectangle, to place all necessary and sufficient cause events, which will directly cause top events occur, into the second row using corresponding event symbols, then to connect top events and direct cause events with appropriate logic gates according to the logic relationships in the real system. So, following the rules of establishing trees to step by step down, the process will finish until all bottom events which are in the lowest row have been written down. Thus, We have established a n-level inversed fault tree, whose root is the top event , whose nodes are middle events and whose leaves are bottom events.

There are 46 fault trees in one test project. Using test-case-design-aided software to optimize test cases, we obtain the minimal cut sets. The number of test cases curtail to 215 from thousands, and they covers all of the original test range.

V. Test Simulation System Design and Implementation

In order to effectively implement the test scheme, it is necessary to develop an appropriate test simulation system. This paper only gives a brief presentation of functional requirements and architecture design of the test simulation system.

A. Functional Requirements

The main functions of Test simulation systems include data simulation, data processing and application, data distribution and capture, test data management and process support.

1) Data Simulation

- Simulation of measure elements of various spaceflight control devices. This requires the noise and wild elements should be easily added and should be flexibly add new devices and new methods.
- Simulation of various device data which are required for spaceflight control.This requires normal data and abnormal data of different devices should be simulated. simulation data can be fixed, it also can be changed by the time.
- It requires that the output timing of a variety of devices be controlled, any device output can be turned on or off, the output combination of different devices can be provided.
- The frame formats of simulation data packages should be flexibly configured and easily modified. The positions, formats, simulation values of all simulation data parameters should be flexibly customized.

2) Data Processing and Applications

- Fault simulation. Fault simulation includes simulation to various aircraft flight failures, and can easily be expanded.
- Analysis to aircraft's trajectory error. It demands to calculate the accuracy of one trajectory of a vehicle relative to another aircraft trajectory.
- Implementation of conversion algorithm of a variety of coordinates, provision of basic math library.

3) Data Distribution and Capture

- The system should provide a variety of data sending and transmission mode, support for multiple network protocols, including point to point protocol, multicast protocol, broadcast protocol, support multiple modes of data output, including serial transmit, time-driven, event-driven .
- The system can read pre-set data, support multiple file formats, including ACCESS, EXCEL, XML, TXT, support multiple package and frame formats that can be packaged according to the configuration file, provide operating interface which can be flexibly configured.
- The system can capture network data of various transmission protocols, support data filtering, which includes filtering by source address, by the data transfer protocol, by the data content of specified location, support data recording; support data display, support filtering to display data, support setting screen refresh frequency.

- The system can analysis the captured data according to the configuration file, support multiple file formats, including ACCESS, EXCEL, XML, TXT. The system can convert information format according to the configuration file
- The system can support computer local time and time uniform system, support for multiple data output modes, including time-driven, event-driven.
- The system can support logging and error alarm.

4) Data Management and Process Support

- The system can provide a unified way of test data storage and query;
- The system can receive all kinds of simulation data in real-time and store them in database to provide public information for the test.
- The system can provide man-machine interface to query, browse, download and upload various types of data, can maintain and manage data, can backup and recovery database.
- The system can identify, statistics and analysis test data to support reuse of test data;
- The system can provide a test platform for more people to cooperate with to provide auxiliary support for the whole test process, including test requirements analysis, evaluation and testing planning, test design and implementation, test execution, regression testing and software quality Evaluation.

B. *Architecture and System Components*

Based on the principle of applicability, usability, flexibility and extension, the Architecture of the test simulation system should be similar with shown in Fig.2.

Within the system, every software can exchange and share data through the network and files. Data Exchange Software is the unified export of external data. The system exchange data with tested software through Data Exchange Software. All software are independent , but also can cooperate with others.

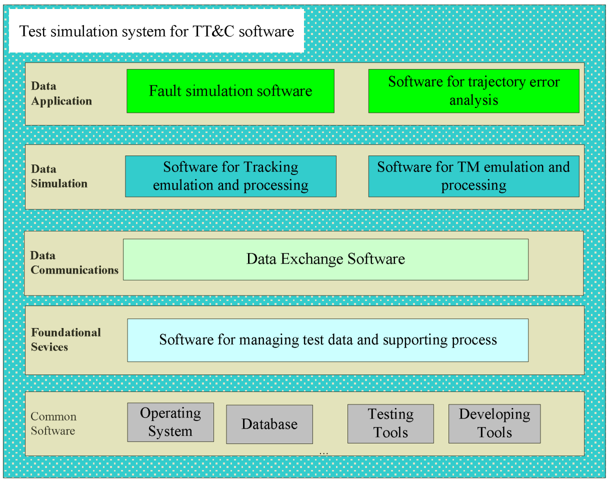
Test simulation system for TT&C software
Data Application
Fault simulation software
Software for trajectory error analysis
Data Simulation
Software for Tracking emulation and processing
Software for TM emulation and processing
Data Communications
Data Exchange Software
Foundational Sevices
Software for managing test data and supporting process
Common Software
Operating System
Database
Testing Tools
Developing Tools
...

Figure 2. The Architecture of Test Simulation System

VI. Instances of Application

Throughout sufficiently analysis to test requirement, we established the evaluation model, as shown in Table 1. Using the evaluation model and fault tree analysis method, we have designed 215 test cases. In support of the test simulation system, we have obtained quality Evaluation data of tested software. We have found 19 major issues of software. This has played important role for improving the TT&C software. As shown in Fig.3, there is the result of the evaluation of failure mode decision software.

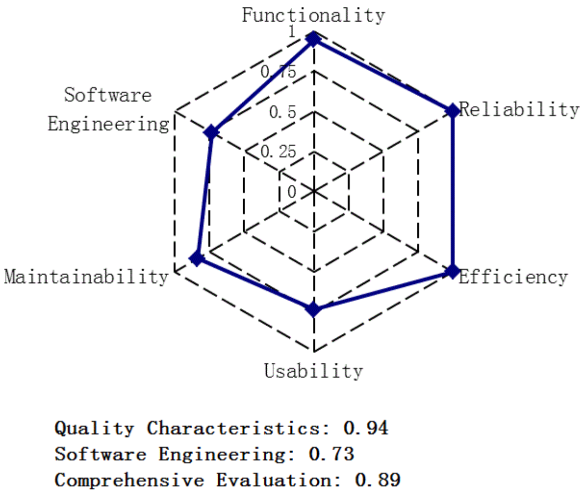

Figure 3. Radar Chart of Software Quality

Dashed circle in the radar chart is the evaluation degree of software quality characteristics. The closer to the center, the lower the evaluation degree of software quality characteristics is. The dashed lines, which radiate out from the center, represent the software quality characteristics and sub-characteristics. The points on these lines are the evaluation values of that software characteristics or sub-characteristics. To connect these points in turn will get the software quality radar chart.

VII. Conclusion

The software quality evaluation model, test case design methods and the test simulation system, which have been established during TT&C software testing, and the scientific management method of test process have strong references in software testing domain, they can guide other software test. The important technologies have been applied in many testing projects, and have produced great effect.

References

[1] WenHong Liu, ZhanWu Wang, XianYin Ma, 2008, Software Testing Process Mangement Method Based on CMM, Beijing:MODERN COMPUTER 2008.2

[2] Wenhong Liu, Zhangwu Wang, Xin Wu, Ying Liu, 2004, Research and Practice on Software Evaluation and Test Technology, Beijing: Journal of Spacecraft TT&C Technology 2004.6

[3] Xiaoting Lu, Pengzhou Zheng, Guowei He,ShengKui Zhen, 1995, Design and Analysis on Reliability, Beijing: National Defense Industry Press, 1995.

TABLE I. THE QUALITY EVALUATION MODEL FOR FAILURE MODE DECISION SOFTWARE

Comprehensive evaluation	**Weight Coefficient**	**Software quality characteristics**		**Sub- characteristics**		**Metric items**	
		Name	*Weight Coefficient*	*Name*	*Weight Coefficient*	*Name*	*Weight Coefficient*
Evaluation of software quality characteristics	0.75	Functi onality	0.29	Suitability	0.41	rate of Functions which are implemented	0.38
						rate of Functions which are correctly implemented	0.50
						rate of non-redundant functions	0.12
				Interoperability	0.16	rate of Data exchange formats which are implemented	0.37
						rate of Data exchange formats which are correctly implemented	0.63
				Security	0.43	rate of Successful access control for software	0.50
						rate of Successful access control for data	0.50
		Reliab ility	0.32	Maturity	0.39	Reliability Evaluation and prediction	0.35
						Test case execution rate	0.35
						rate of intension compliance	0.30
				Fault tolerance	0.39	Rate to avoid misuse	1.00
				recoverability	0.22	Rate of successful Restarting	1.00
		Efficie ncy	0.17	Time characteristics	0.69	compliance rate of Fast response time	1.00
				Resource characteristics	0.31	CPU utilization rate	1.00
		Usabil ity	0.10	understandability	1.00	Compliance Rate of Module cyclomatic complexity	0.23
						Compliance Rate of Module size	0.22
						Compliance Rate of GOTO statements	0.18
						Compliance Rate of Module comments	0.16
						Compliance Rate of Fan-out number	0.21
		Maint ainabil ity	0.12	analyticity	0.75	Rate of successfully finding the reason of Failures	1.00
				changeability	0.25	the degree of easily to change	1.00
Software Engineering	0.25	=	=	=	=	three cases of separation of Project teams	0.19
						training for Development and management staff	0.14
						The use of software tools	0.14
						schedule control of Software development	0.14
						Effectiveness of reviews for each phase	0.20
						execution of configuration management of project	0.19