UDC 004.052

N. KHILCHENKO

National aerospace university named after N.E. Zhukovsky "KhAI", Ukraine

RISK-ORIENTED SOFTWARE RELIABILITY ASSESSMENT

Software risks and risk management model are discussed. Software reliability risks are considered as a part of general software risk model. Software reliability assessment risks concerned with reliability assessment method used are analysed and appropriate risks model is proposed. It is offered the software reliability risk management method based on the reliability assessment method considered.

software reliability, software risk management, software reliability assessment, software reliability models, software reliability management

Introduction

The safety of computer based information and control systems (I&CS) for critical applications (aerospace systems, nuclear power plants, medical systems, etc.) depends on the software reliability. The risks of ensuring appropriate level of reliability and trustworthy reliability assessment are important parts of a general software risks model. Therefore the problem of critical software reliability ensuring and assessment is one of the dominant problems of I&CSs development, examination and maintenance.

It is essential to say, that the trustworthy software reliability assessment is no less important, than its ensuring. Moreover, it is impossible to guarantee the required level of reliability without using the accurate assessment techniques. That is why we should give proper attention to reliability assessment phase and find the way to reduce risks concerned with it.

The used quantitative reliability assessment approach is based on the Software Reliability Growth Models (SRGMs) [1]. The initial model data is the software characteristics as well as software development and testing features and statistic data about faults revealed during testing phase. The output data is reliability measures and reliability predictions.

In the context of reliability assessment the major risk is an accuracy of the reliability measures assessment and an adequacy of the predictions obtained. The midpoint of the assessment is the selection of the most proper model. So there is a risk of unequal model. Moreover, we cannot obtain accurate measures using proper model if we have inadequate or insufficient input data. So, to perform trustworthy reliability assessment we should apply appropriate technique that allows managing all such risks.

There are a lot of approaches and concepts of software risk management [2 – 4] but there is no corresponding attention to reliability risks and their management in literature. The purpose of the research was to analyse known software risk management methods, to construct software reliability risks model and to propose appropriate technique for software reliability risks management on the basis of reliability assessment method used.

Software risk management

Risk is the possibility of loss. It is a function of both the probability of an adverse event occurring and its impact; the impact manifests itself generally in financial loss, time delays, loss of performance, etc. A risk is the precursor to a problem, the probability that, at any given point in the software life cycle, the predicted goals cannot be achieved within available resources. Risk cannot be eliminated from a software project, but it can be managed. Risk management is critical to the success of any software effort and is a strategic aspect of all software projects and especially of I&CS for critical application.

Software risk management is a software engineering practice with processes, methods, and tools for managing risks in a project [4]. It provides a disciplined environment for proactive decision-making to assess continuously what can go wrong; determine what risks are important to deal with; and implement actions to deal with those risks. Risk management planning addresses the strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process.

There are several models available for risk management. Here is one of these that was developed by the Software Engineering Institute and is shown in fig. 1 below.

This model identifies the fundamental risk management functions that must be taken to effectively manage risk: identify, analyze, plan, track, control, and communicate. In all phases of a software life cycle, risks should be assessed continuously and used for decision-making.

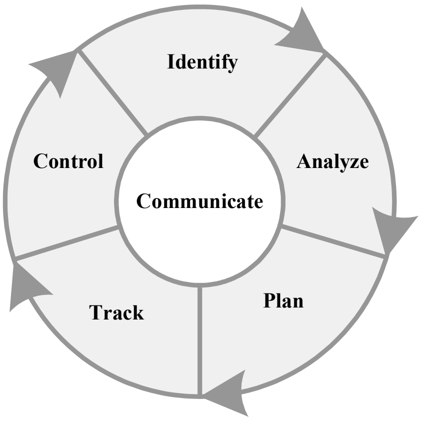


Fig. 1. A continuous set of activities to identify, confront, and resolve technical risk

Let’s consider activities meaning:

– ***Identify***: Search for and locate risks before they become problems adversely affecting the project.

– ***Analyze***: Process risk data into decision-making information.

– ***Plan***: Translate risk information into decisions and actions (both present and future) and implement those actions.

– ***Track***: Monitor the risk indicators and actions taken against risks.

– ***Control***: Correct for deviations from planned risk actions.

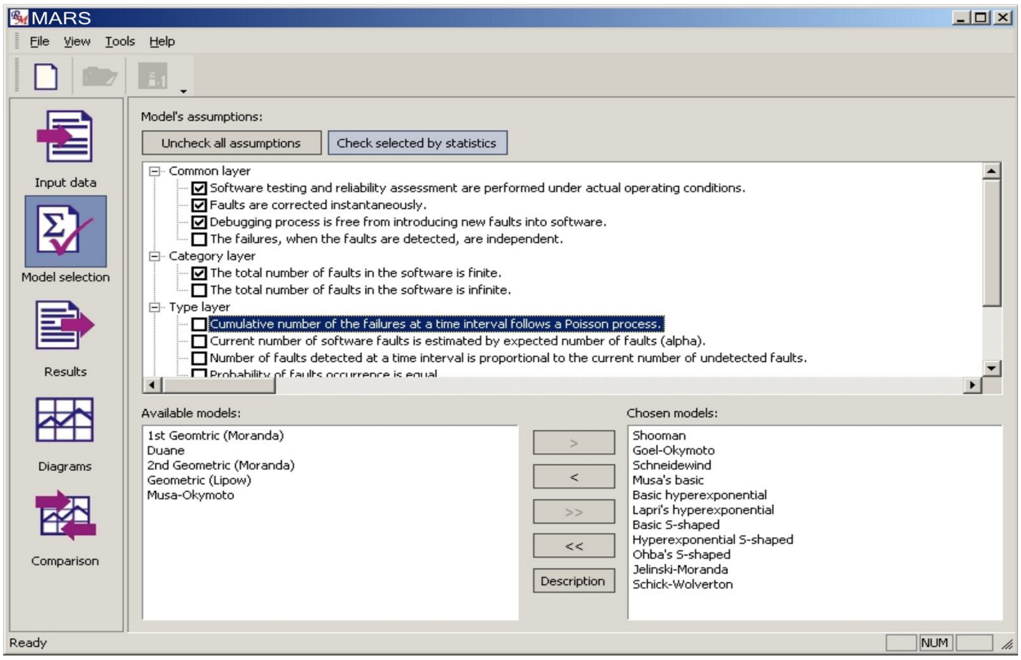
– ***Communicate:*** Provide visibility and feedback data internal and external to your program on current and emerging risk activities.

This model can be applied on the basis of software life cycle phases or/and according to the target software characteristics that are represented by software requirements. Since our interests lie in a software reliability that is one of the major characteristics of I&CSs and its trustworthy assessment let’s try to apply the above model for software reliability management.

Software reliability assessment method and tool

For the quantitative software reliability assessment we use SRGMs [1]. The matrix approach is used to select such models whose assumptions are conformed to software project features in the best way [5, 6]. For that the most typical assumptions are selected from an organic set of assumptions on the basis of analysis of technical and project documentation. These assumptions are used as input for assumptions matrix. The output of this matrix is the single SRGM or a group of models that are complied with the selected assumptions and, therefore, are adequate for analysed software.

The method considered is realized by using the tool for probabilistic software reliability assessment “SRM-Tool”. This tool works under Windows OS and implements graphic user interface. User can enter assessment parameters, examine assessment results, adjusts results view, etc. Also the tool implements interface to communicate with other tools for automatic determination of metrics and models parameters (static software code analysers, tools for collection and handling of technical and project documentation, libraries for statistic data processing). Moreover, tool communicates with database in which assumptions, models and its attributes are stored. The results of software reliability assessment are also stored in the database. The tool screenshot is represented on fig. 2.



Model's assumptions:

- Common layer
  - ☑ Software testing and reliability assessment are performed under actual operating conditions.
  - ☑ Faults are corrected instantaneously.
  - ☑ Debugging process is free from introducing new faults into software.
  - □ The failures, when the faults are detected, are independent.
- Category layer
  - ☑ The total number of faults in the software is finite.
  - □ The total number of faults in the software is infinite.
- Type layer
  - □ Cumulative number of the failures at a time interval follows a Poisson process.
  - □ Current number of software faults is estimated by expected number of faults (alpha).
  - □ Number of faults detected at a time interval is proportional to the current number of undetected faults.
  - □ Probability of faults occurrence is equal

Available models:

1st Geomtric (Moranda)
Duane
2nd Geometric (Moranda)
Geometric (Lipow)
Musa-Okymoto

Chosen models:

Shooman
Goel-Okymoto
Schneidewind
Musa's basic
Basic hyperexponential
Lapri's hyperexponential
Basic S-shaped
Hyperexponential S-shaped
Ohba's S-shaped
Jelinski-Moranda
Schick-Wolverton

Fig. 2. “SRM-Tool”: model selection window

Software reliability risks model

Software reliability risks can be reasonably divided into two major groups: risks of ensuring reliability and risks of reliability assessment. Both of these groups contain general software risks like a risk of exceeding financial, time and other resources while achieving reliability goals. And both of them have special risks like a risk of impossibility of ensuring appropriate (defined by requirements) level of software reliability (as to the first one) and a risk of inability of trustworthy reliability assessment (as to the second one).

The first group risks management should involve the actions to reduce the risk of failures in the software and to increase the efficiency of its detection. The second group risks management should involve activities to accurately assess reliability measures and adequately predict software reliability.

According to our research interests risks from the second group are considered only and risks model for the reliability assessment method used are proposed.

The basic risks of reliability assessment are shown on the fig. 3 below. Two basic risks are concerned with proper model selection and collection of the data about faults detection.

Actually software reliability measures could not be assessed accurate if model does not fit to software concerned. Even if it is proper model accurate assessments could not be obtained if it is not confidence to data about faults detection (they may be inadequate and/or insufficient).

A risk of adequate model selection strongly depends on the selection method. The model selection method based on the assumptions matrix [5, 6] allows us to reduce risk of model conformity to software assessed. But this method introduces a set of particular risks, which are the followings:

– risk of determination realistic characteristics of software itself and processes of its development. This information is one of the most important for correct model selection;

– risk of correct transformation of above characteristics to model assumptions. Unequal transformation affects selection of inappropriate model;

– risk of nonempty set of proper models. No model may fit to a set of assumptions defined.

Moreover the method is applied after (or at the end of) verification phase. All statistic data about software defects revealed and characteristic of the software itself and the processes of its development, verification and utilization are known at this moment. And often there is no model fitted to a set of assumptions promoted. In this case the process should be repeated with reduced set of assumptions rejecting less important of them or even custom model should be developed.

So, it is reasonable to try to start a process of selection (or a preparation to selection) of the model before verification phase. Moreover it would be a good thing to start defining assumptions at the project beginning constricting a set of adequate models step-by-step during all software life cycle.

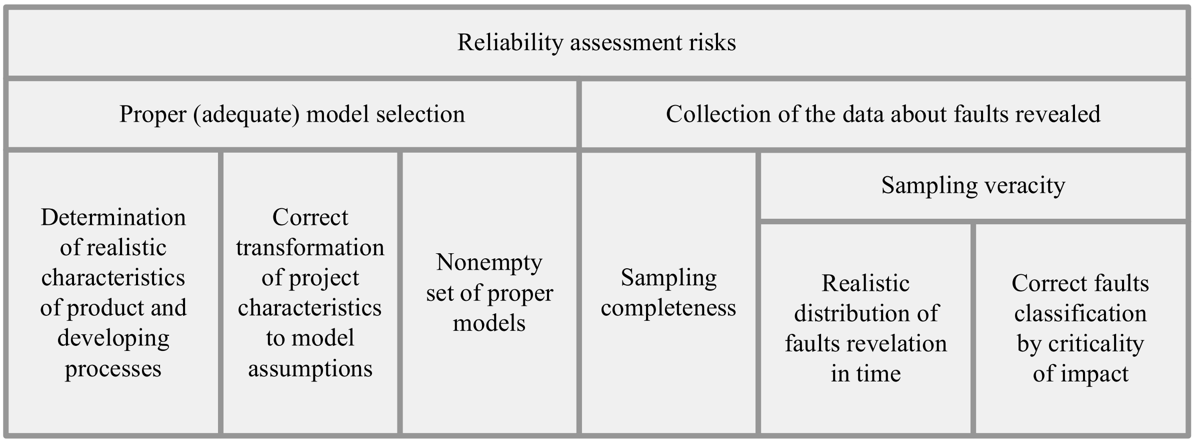

Reliability assessment risks
Proper (adequate) model selection
Collection of the data about faults revealed
Determination of realistic characteristics of product and developing processes
Correct transformation of project characteristics to model assumptions
Nonempty set of proper models
Sampling completeness
Sampling veracity
Realistic distribution of faults revelation in time
Correct faults classification by criticality of impact

Fig. 3. Reliability assessment risks

The second group of reliability assessment risks is concerned with collection of faults detection data. This data is collected during software testing phase. After processing they are represented in the format of faults count per testing interval or in the format of time between faults. Model estimates reliability measures using this data as input. This group contains the following risks:

– sampling completeness. Data incompleteness results in impossibility of SRGMs use;

– sampling veracity relating to reality of faults distribution in time. Unrealistic faults distribution affects false reliability measures and invalid reliability predictions;

– sampling veracity relating to correct classifying faults by influence. Using unsorted faults statistics may affect more optimistic reliability measures and predictions.

As to collection of the data about faults revealed more accurate techniques and close control of the process should be used to obtain realistic data about failures distribution [1]. Moreover, this data should be carefully processed and faults should be classified by their influence on the system operation.

Of cause more critical faults have stronger impact on the reliability measures.

Software reliability assessment risks management technique

As a result of reliability risks analysis we propose the following basic technique for software reliability assessment risks management. Of cause it should be expanded by actions for specific software project and reliability risks. Note that all activities depend on reliability goals defined in the software requirements.

Let’s start from general management activities:

– human resources with appropriate qualification and appropriate time resources should be assigned for software reliability assessment and its management;

– human resources with appropriate qualification and appropriate time resources should be assigned for the phase of system testing and collecting data about faults revealed;

– the process of collection of the data about faults revealed should be carefully planned and controlled.

– Here is a set of technical activities according to the reliability assessment method used:

– assumptions analysis should be started as earlier as possible. It is essential to use information from previous similar projects if it is available;

– the basic assumption about testing software under conditions similar to conditions of utilization should be ensured;

– a set of assumptions that should be achieved should be formed according to the software requirement and developing experience. Appropriate management and technical activities should be planned;

– collecting and processing the data about faults revealed should be made accurately and carefully. Appropriate techniques and resources should be used to obtain realistic distribution of faults in time and correct faults classification by criticality of impact [1];

– estimations should be made using approved tools and techniques.

Conclusion

The risk-oriented software reliability assessment method allows us to control and manage software process in respect to reliability goals. However this approach causes corrections to overall software development process and assumes additional resource and time costs. Advisability of using this method should be properly measured by costs-benefit analysis for the each software product developed. I&CSs are made high demands to reliability and safety, therefore the use of risk-oriented reliability assessment approach seems to be efficient method for achieving reliability goals.

As to advantages relating to reliability assessment method used on the one hand the approach allows specifying assumptions earlier and constricting a set of applicable models gradually. It allows to control a model selection process and to correct it if necessary. On the other hand it gives an opportunity to form a subset of assumptions that can be managed during software development and as a result to correct a set of applicable models.

References

1. Lyu M. Handbook of Software Reliability Engineering. – McGraw-Hill Company, 1996. – 805 p.

2. Reliability Management. NASA Lewis Research Center. – 2004. – 37 p.

3. Barry W. Boehm. Software Risk Management: Principles and Practices // IEEE Software. – Jan/Feb, 1991. – Vol. 08, no. 1. – P. 32-41.

4. Software risk management: A Practical Guide. Department of Energy Quality Managers. – SQAS21.01.00. – 2000. – 31 p.

5. Kharchenko V., Tarasyuk O., Sklyar V., Dubnitsky V. The Method of Software Reliability Growth Models Choice Using Assumptions Matrix // Proceedings of 26th Annual International Computer Software and Applications Conference (COMPSAC). – Oxford, England, Aug. 2002. – P. 541-546.

6. Kharchenko V., Tarasyuk O., Gorbenko A., Khilchenko N. A metric-probabilistic assessment of software reliability: Method, Tool and Application // Proceedings of East-West Design and Test Workshop (EWDTW’05). – Odessa, Ukraine, 2005. – 123 p.

Поступила в редакцию 16.03.2006

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.