A Survey on Software Reliability Demonstration

Wei He, Jianyang Ding, Xiaomei Shen, Xinyu Han and Longli Tang
China Institute of Marine Technology and Economy, Beijing, P. R. China

E-mail: wei.he@live.cn

Abstract. Software reliability demonstration is the process of making scientific decisions on the development and management of software through logical reasoning with sufficient evidence and rigorous methodology. It is the first and foremost task in software reliability engineering. There is much literature on various aspects that aim to facilitate software reliability demonstration. In this paper, we provide an overview of the related concepts, principles, activities, standards and techniques. Besides, we discuss key issues that are pertinent to software reliability demonstration and presents directions for future research.

1. Introduction
The influence of the information industry on a global scale has increased remarkably during the past decade. Software systems have penetrated into various industrial fields and promoted the continuous progress of these fields [1]. Information equipment has become one of the main forces of the industrial revolution. It is the biggest difference that the role of software has changed between information equipment and traditional mechanized equipment. Nowadays, software has been the main approach to realize information exchange and control logic, which works as the “brain” of equipment. It is responsible for monitoring and controlling each part of the equipment. The overall reliability of such systems depends largely on its software. Hence, reliability is deemed as one of the most important focus for any software system [2].

Errors, which may lead to defects, are inevitable in software design and implementation. Once a user run the software with some specific input parameters or execution process, defects are possible to be activated and cause software failures, which may lead to unexpected loss of property or even lives [3].

In recent years, the loss caused by software reliability problems has increased dramatically. Due to insufficient reliability, a large number of software release was delayed, the budget of was exceeded, and even the entire project was terminated [4]. Software systems used in key areas such as chemical engineering, nuclear power, aeronautics and astronautics will cause a catastrophe if a fatal software
defect is triggered [5, 6]. As a result, many scientists consider that software reliability should have a higher priority than functionality and performance [1].

It is the basis of equipment development to demonstrate the indexes according to the user demand. Each index should be demonstrated at the proposal stage. The equipment reliability demonstration should start from the system level reliability requirements and refine gradually to a suitable product level.

For software systems, the first and foremost task of reliability engineering is to determine a set of reasonable software reliability indexes. To be specific, these indexes should be proposed according to the requirements and characteristics of the equipment, and taken as constraints in the whole process of software development.

The essence of software reliability index demonstration is to put forward the reliability requirements of software from the user's point of view, including the selection of reliability parameters and the determination of parameter values. These requirements will directly affect the design scheme, implementation approach, schedule and cost of a software project [7].

Nowadays, most of the software reliability requirements are not clearly defined, and a comprehensive software reliability parameter framework has still not been established. There is not a systematic guidance on how to determine the software reliability indexes. When analyzing the reliability of an equipment, it is usually to assume that the software will not affect the reliability of the whole equipment, and ignore the influence of software reliability. This assumption is not the case in reality. The reliability analysis result obtained from this assumption is obviously not reliable. Even if some software reliability indexes are considered in certain projects, these indexes, mainly based on past experience or similar products, are highly subjective and hardly to be reasonable.

In recent years, much more attention has been paid on software reliability demonstration. Some qualitative or quantitative techniques (such as the system balancing method, the release date method, and the life cycle cost optimization method) have been proposed for determining software reliability indexes. With some techniques already in existence and others continually being proposed, as well as with advances being made both from an academic and industrial perspective, it is important to overview and categorize current techniques in software reliability demonstration in order to provide a comprehensive survey for those already in this research area as well as those interested in making contributions to it.

There is much literature on various techniques that aim to facilitate software reliability demonstration. Despite the fact that these techniques share similar goals, they can be quite different from one another and often stem from ideas that originate from diverse disciplines. Due to space limitations, we group techniques into appropriate categories for collective discussion with an emphasis on the most important features of these techniques.

The remainder of this paper is organized as follows: we begin by introducing the basic concepts of software reliability demonstration in Section 2, describing primary principles for software reliability demonstration in Section 3. In Section 4, we depict a framework about the procedure of software reliability demonstration, followed by a discussion of critical aspects, such as standards and techniques, on software reliability demonstration in Section 5. Finally, conclusions are presented in Section 6.
2. Basic Concepts of Software Reliability Demonstration

Demonstration is a process of analyzing research problems, it can provide a basis for decision-making. Once a decision is made upon the demonstration result, the demonstration result will be the guidance to implement the decision.

Demonstration is generally composed of three elements: proposition, evidence, and method of demonstration. The "proposition" is a clause that needs to determine its authenticity or judge its feasibility in the proof. In the field of science and technology, proposition is the goal, requirement, plan, or measure, of a certain problem. It is the core of the entire demonstration process and is mainly used to clarify the question of "what is to demonstrate". The "evidence" is all kinds of data used in the process of proving the proposition. It answers the question of "what can be used for demonstration". The "method" is the connection between the proposition and the evidence. It solves the problem of "how to use evidence to support the proposition".

Reliability demonstration is the process of making scientific decisions on the development and management of equipment through logical reasoning with sufficient evidence and rigorous demonstration methods. As an important part of the equipment demonstration work, equipment reliability demonstration is the process of determining the requirements on reliability indexes for a brand new equipment according to user requirements, and avoiding reliability problems of similar equipment.

With respect to equipment reliability demonstration, the proposition of demonstration is the reliability level of the equipment; the evidence of demonstration includes all kinds of data and related information such as literatures, standards, questionnaire and experimental results; the method of demonstration is the logical reasoning approach to determine equipment reliability indexes, including analogy, comparison, deduction, causal analysis, etc.

The automation level of modern equipment improves all the time. Some software has even empowered the equipment "intelligence" in terms of autonomous perception, object recognition, collaborative control, inference prediction, planning and decision-making. Meanwhile, the scale and complexity of the software have increased dramatically, making software development and quality assurance increasingly difficult.

According to the ISO/IEC/IEEE 24765 standard, "software reliability" is defined as the probability of software running without failure for a specified time under specified conditions [8]. Software reliability indexes are used to describe the requirements for software reliability with quantitative parameter values, which consists of two elements, namely reliability parameters and their values. Random sampling and statistics can be used to evaluate the reliability indexes by means of running the software.

The first task of software reliability engineering is to determine software reliability indexes, which will directly affect the design scheme, implementation approach, schedule and cost of the whole software project. Each index is regarded as a kind of measurement on software reliability in performing requirement analysis, design, development, and testing. Software development cycle, cost, and quality should be coordinated from the perspective of software engineering whether the software project is independent or belonging to a complex large system.
3. Principles for Software Reliability Demonstration

In the demonstration stage, software reliability indexes are determined according to the type, the importance and the operating profile of the software. The demonstration process of software reliability indexes should follow the following principles.

3.1. The principle of requirement satisfaction
User requirements are the main basis for the demonstration of software reliability indexes. The minimum acceptable value of the software reliability index is necessary to meet basic usage requirements. After the reliability growth in the trial and refinement stage, the software reliability index value in the mature stage can reach up to 1.2 to 3 times to the minimum acceptable value.

3.2. Principle of suitability
The selection of software reliability parameters should be based on the current user requirements and possible improvement directions in the near future. At the same time, the determination of software reliability indexes should be practical, considering not only the needs of all potential usage scenarios but also the existing technologies and management skills. Suitable software reliability indexes can promote the developers to continuously improve the quality assurance capability.

3.3. The principle of independence
A standardized hardware reliability parameter framework have been established, while the software reliability parameter framework is still under study. When conducting software reliability demonstration, software reliability indexes should be determined at first, and then the correlation with the hardware reliability indexes should be considered. Finally, the indexes are balanced to achieve an overall optimization.

3.4. The principle of economical efficiency
Improving software reliability indexes requires time and labor costs. With the increment in software reliability, software development and management costs grow significantly. When determining software reliability indexes, one should not only pursue the advancement of the indexes, but also consider the economical efficiency. It is a better way to perform a comprehensive prioritization of multiple software reliability indexes for choice, and select the fewest indexes that users care about as much as possible to achieve a good cost effectiveness.

3.5. Principle of feasibility
Feasibility must be taken into account when demonstrate software reliability indexes. The indexes should be measurable and evaluable using existing techniques, preferably supported by open source or commercial tools.

4. Main activities of Software Reliability Demonstration
The main activities of software reliability engineering include reliability aim identification, reliability requirement determination, reliability design and analysis, reliability testing and verification, reliability measurement, reliability engineering management, etc. [9].
Each activity of software reliability engineering is closely related to software reliability indexes. As shown in Figure 1, software reliability indexes go from scratch to refinement, implementation, verification and maintenance. The identification of software reliability requirements in the first stage of Figure 1 refers to the reliability aims and demands of the software from user’s perspective, and is the basis for subsequent software reliability activities. In the process of software development, it is crucial to have an accurate understanding of the software reliability requirements and conduct software reliability engineering.

**Figure 1. Software reliability engineering framework**

The procedure of software reliability index demonstration is shown in Figure 2. First of all, we should collect relevant data, including the equipment demonstration data and the software runtime data.

**Figure 2. Procedure of software reliability demonstration**

The procedure of software reliability index demonstration is shown in Figure 2. First of all, we should collect relevant data, including the equipment demonstration data and the software runtime data.
Then, we should analyse the data and clarify the user requirements. The third step is to determine a set of software reliability indexes. After that, the fourth step is to refine the indexes. In the fifth step, a feasibility analysis is conducted to ensure that the indexes are practical. The sixth step, we should determine the timing and method to verify the indexes. Finally, a software reliability demonstration report should be carefully produced.

(1) Data collection
Software reliability demonstration must be supported by sufficient data. At least two kinds of data related to the software should be collected. On one hand, demonstration data of the equipment where the software is located, including user requirements and use schemes of the equipment, is a kind of important data. These data directly affects the reliability requirements of the software, and can be deemed as a set of preconditions of the software reliability requirements. On the other hand, the relevant data of similar software is needed, which can be referenced in defining the failure modes and operating profiles.

(2) Data analysis
After collecting the required data, we should organize expert meetings to analyse the data. The following issues should be analyzed:

① The user requirements and task requirements of the software;
② The operating profile of the software;
③ The definition of software failure modes;
④ The selection of software reliability parameters;
⑤ The method of determining software reliability indexes.

(3) Preliminary determination of software reliability indexes
According to the method of determining software reliability parameters adopted in the expert meeting, a set of preliminary software reliability indexes are proposed.

(4) Refinement of software reliability indexes
Improve the software reliability indexes if necessary. For example, the proposed preliminary software reliability index is the mean time between failure (MTBF) and the reliability of the software system is expected to be measured by a failure rate, then a parameter conversion should be performed to obtain a more applicable index. When the software is composed of multiple components, the preliminary indexes needs to be allocated to each component.

(5) Feasibility analysis of software reliability indexes
Economical feasibility analysis and technical feasibility analysis should be performed on the software reliability indexes to ensure that the indexes can be put into practice.

(6) Determination of the timing and method of software reliability index verification
For each software reliability index, the timing and method to verify the index must be determined. Verification testing procedures and parameter evaluation models should be provided if necessary. A software reliability indicator demonstration report should be made according to the above procedure.

5. Critical Aspects on Software Reliability Demonstration
5.1. Software Reliability Demonstration Standards
At present, the main basis for carrying out equipment reliability work is IEEE Std 982.1 Dictionary of Measures of the Software Aspects of Dependability [10], ISO / IEC 25023 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Measurement
of System and Software Product Quality [11] and IEEE Std 1633 Recommended Practice on Software Reliability [12].

IEEE Std 982.1 provides a standard dictionary of measures for assessing and predicting the reliability, maintainability, and availability of any software system. In particular, it applies to mission critical software systems.

ISO / IEC 25023 defines quality measures for quantitatively evaluating system and software product quality in terms of characteristics defined in ISO / IEC 25010 and is intended to be used together with ISO / IEC 25010. It can be used in conjunction with the ISO / IEC 2503n and the ISO / IEC 2504n standards or to more generally meet user needs with regard to software product or system quality.

IEEE Std 1633 describes the techniques for assessing and predicting the reliability of software, based on a life-cycle approach to software reliability engineering. It provides information necessary for the application of software reliability measurement, and establishes the basic principle for assessing and predicting the reliability of software. The recommended practice prescribes how to perform software reliability assessments and predictions.

However, there is currently a lack of technical standards that can be used specifically to guide software reliability demonstration.

5.2. Software Reliability Demonstration Techniques

Regarding the determination of software reliability indexes, existing studies have proposed some techniques, including the system-balancing method, the release-date method, and the life-cycle-cost optimization method, etc. [9].

The system-balancing method is mainly used for the reliability allocation of software modules based on the overall reliability requirements of the system. The basic idea of this method is to balance the development difficulty of different modules. Lenient reliability indexes are assigned to the complex modules. By this means, the overall reliability requirements of the system are met, while minimizing the workload of processing complex modules. The release-date method is mainly used for software systems with fixed release date. The reliability indexes of such software systems are determined by available resources and funds. The life-cycle-cost optimization method assumes that the cost of software testing is related to the failure rate, and the improvement of software reliability can be obtained through more tests [13].

However, in the current work, many people still have misunderstandings about software, thinking that software is embedded in a target hardware device and confusing software development and hardware device development. At the same time, considering the limited software reliability demonstration techniques, most projects do not have clear definitions on software reliability requirements. There is a lack of systematic guidance on how to determine software reliability indexes [14,15]. Because of the above misunderstandings and technical shortcomings, when performing equipment reliability demonstration, the top-level reliability indexes for the entire equipment are usually determined and the indexes are rarely allocated to each software, sometimes even assuming that the software will not affect the reliability of the entire equipment, thereby ignoring the software reliability. This assumption is not appropriate in most situation. Even if some software reliability indexes are proposed in certain projects, the indexes are mainly based on past experience or similar
product requirements. It is subjective and difficult to ensure that the indexes are scientific and reasonable.

6. Conclusion
Under the modern information technology, higher software reliability is demanded. As an important part of information equipment, software should be carefully designed and implemented with good reliability. In particular, software reliability demonstration, as the basis of software reliability engineering, should be conducted at first. The resulting software reliability indexes will directly affect the software design scheme, development methodology, development schedule and whole cost. Therefore, researches on software reliability demonstration is urgently needed to be conducted. Techniques are highly demanded to tackle the issues about how to scientifically determine software reliability indexes and how to evaluate the reasonableness of the indexes, etc.

In order to meet the needs of software reliability demonstration, modern information technologies, such as machine learning and system simulation, can be employed. Automated (or semi-automated) tools are also needed to be developed to support software reliability demonstration. Both the academic and industrial society can take an active part in these researches and make substantial contributions to promote the software reliability.

References
[1] Taylor R N and van der Hoek A 2007 Software design and architecture: the once and future focus of software engineering Proc. IEEE Future of Software Engineering (Los Alamitos: IEEE Computer Society) 226-43
[2] Glass R L 2002 Software Engineering: Facts and Fallacies (Boston: Addison Wesley)
[3] Tassey G 2002 The economic impacts of inadequate infrastructure for software testing (Gaithersburg: National Institute of Standards and Technology)
[4] Littlewood B and Strigini L 1993 Validation of ultrahigh dependability for software-based systems Communications of The ACM 36(11) 69-80
[5] Guo J Yang M Zou B Zhang Y Yang J and Dai X 2018 Nuclear safety-critical digital instrumentation and control system software: reliability demonstration Annals of nuclear energy 120 516-27
[6] Yates S W and Ali M 2006 A Bayesian Approach to Reliability Demonstration for Aerospace Systems. IEEE Rams 06 Reliability & Maintainability Symposium (Los Alamitos: IEEE Computer Society)
[7] Musa J D Iannino A and Okumoto K 1987 Software Reliability: Measurement, Prediction, Application (New York: McGraw-Hill)
[8] ISO/IEC/IEEE 24765 2017 Systems and Software Engineering – Vocabulary
[9] Lyu M R 1996 Handbook of software reliability engineering (New York: McGraw-Hill)
[10] IEEE Std 982.1 2005 Dictionary of Measures of the Software Aspects of Dependability (Los Alamitos: IEEE Computer Society)
[11] ISO/IEC 25023 2016 Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Measurement of System and Software Product Quality (Geneva: International Organization for Standardization)
[12] IEEE Std 1633 2016 *Recommended Practices on Software Reliability* (Los Alamitos: IEEE Computer Society)

[13] Sandoh H 1991 Reliability demonstration testing for software *IEEE Transactions on Reliability* **40**(1) 117-9

[14] Tal O Mccollin C and Bendell T 2001 Reliability demonstration for safety-critical systems *IEEE Transactions on Reliability* **50**(2) 194-203

[15] Lyu M R 2007 Software Reliability Engineering: A Roadmap *Proc. IEEE Future of Software Engineering* (Los Alamitos: IEEE Computer Society)