FAST ABSTRACT: A User Friendly Software Reliability Analysis Tool based on Development Process to Iteratively Manage Software Reliability

Myungmuk Kang, Taewan Gu, Jongmoon Baik
Korea Advanced Institute of Science and Technology
335 Gwahangno, Yuseong-gu, Daejeon, 305-701, Republic of Korea
{mmkang, gutaewan, jbaik}@kaist.ac.kr

Abstract

Software Reliability has been regarded as one of the important quality attributes because low reliable software systems have high possibilities of causing serious problems such as the loss of human life, catastrophic mission failures, and the waste of valuable resource investments. In order to improve software reliability, the degree of current software reliability needs to be measured and managed by using an automated evaluation tool in a convenient way throughout the whole SDLC (Software Development Life Cycle). However, general open tools such as CASRE (Computer Aided Software Reliability Estimation) and SMERFS (Statistical Modeling and Estimation of Reliability Functions for Software) cannot be used within the early SDLC phases. In order to improve the drawbacks of the open tools, we developed the SRTpro (Software Reliability Tool professional) which can predict and estimate software reliability throughout the whole SDLC in a convenient way.

1. Introduction

Among software quality attributes, reliability has been regarded as the most important factor since software reliability is associated with faults and failures representing the largest cost element in software development life cycle. Software reliability is the probability of failure-free operation of software intensive systems in a specific environment for a specific time period [1]. In order to develop high reliable software, it is necessary to control and manage software reliability iteratively throughout the life cycle. For this reason, several software reliability analysis tools have been developed and used. CASRE [2] and SMERFS [3] are the most widely used open tools to support measuring and analyzing software reliability. By using these open tools, current and future software reliabilities of software systems can be measured and estimated with ease.

Nevertheless, the open tools have limitations as follows: First, these tools can be used only during testing phases. Second, failure data manipulation is inconvenient. Third, the tools support results based on failure information without the project related information such as project name, description, and date. However, in order to iteratively manage software reliability, it is necessary to manage software reliability throughout the whole SDLC and record project information.

The primary purpose of the research is to propose a new open tool to iteratively manage software reliability throughout the whole SDLC.

2. SRTpro

In this section, we introduce the new open automated tool, called “SRTpro”, to iteratively manage software reliability. Figure 1 is the screenshot of SRTpro. SRTpro consists of three major parts in the internal architecture. The each part is explained as:

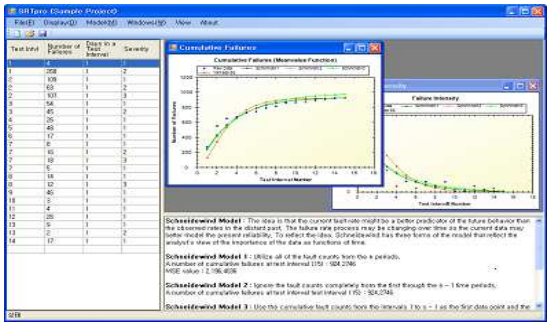

Figure 1. SRTpro

1. Data and Report Files

A data file can be used at the first time to enter users’ input data and to save all the input data and model selections. A report file can be created with all data and analysis results with graphs in html format.

2. Data Management

One of the most important structures is data management because it contains and processes all data in memory. Therefore, when users load a file or create a new project, data management keeps all users’ input

and modified data. In addition to these, users' selections are also managed. Especially, a report function depends on data management.

3. Prediction and Estimation Models

We have implemented estimation models based on CASRE and SMERFS. We have also implemented several prediction models to predict software reliability at the early phases. Currently the following models have been implemented.

- Prediction models
  1. RL-TR-92-52 Model [5]
  2. Musa Basic Method [5]
  3. COQUALMO [4]
  4. Industry Data [5]

- Estimation models [6]
  1. Non-homogeneous Poisson Process Model (NHPP)
  2. Schneidewind Model
  3. Yamada S-Shaped Model
  4. Generalized Poisson Model
  5. Musa Okumoto Model
  6. Musa Basic Model
  7. Jelinski Moranda Model
  8. Littlewood and Verrall Linear Model
  9. Littlewood and Verrall Quadratic Model
  10. Non-homogeneous Poisson Model (TBE)

Table 1. Overall comparison of SRTpro and two existing open tools

Tool name	Supplier	Development language	Available models		Parameter estimation		Performance and usability		
			Estimation models	Prediction models	Maximum likelihood	Least squares	reliability or failure rate	Total failures	Remaining failures
SRTpro 1.0	SPIRAL Lab in KAIST	C#	O	O	O	X	O	O	O
CASRE 3.0	Jet Propulsion Laboratories	FORTRAN	O	X	O	O	O	O	O
SMERFS^3	NSWC	FORTRAN	O	X	O	O	O	O	O

Performance and usability								
Severity selection	Project information	Data addition	Data modification	Data transformation	User input and selections store	Result update	Compatibility with other tools	Available phases
O	O	O	O	X	O	Immediately	LDRA	SDLC
O	X	X	X	O	X	By reselection	X	testing
O	X	△	△	O	X	By reselection	X	testing

△: possible, but inconvenient

3. Overall Comparison of Tools

Table 1 provides an overall comparison of SRTpro and the two exiting open tools. Through Table 1, tools related information, available models, parameter estimation, and performance and usability of the tools can be identified.

4. Conclusion

We proposed the new open tool, called "SRTpro", to iteratively manage software reliability and deal with limitations. From the comparison of SRTpro and two existing tools, we confirmed that the new open tool can provide user convenience and project management for assessing and improving software reliability.

The next version would provide more models and functions such as constant value change for researchers to make models fit many projects or study. We would continue to improve performance and usability of SRTpro through testing and feedbacks from many users after the release.

ACKNOWLEDGMENT

This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement). (IITA-2009-(C1090-0902-0032))

5. References

[1] John D. Musa, Software Reliability Measurement, Prediction, Application, p.1-29, 1987.
[2] Allen Nikora, CASRE–A Computer-Aided Software Reliability Estimation Tool, http://www.openchannelfoundation.org/projects/CASRE_3.0.
[3] William Farr, Oliver Smith, SMERFS – Statistical Modeling and Estimation of Reliability Functions for Systems, http://www.slingcode.com/smerfs/drfarr.php, 1996.
[4] Deun Lee, SunitaChulani, Barry W. Boehm, COQUALMO, USC/CSE, 2001.
[5] Peter B. Lakey, McDonnell Douglas Corporation, St. Louis, System and Software Reliability Assurance Notebook, Rome Laboratory, pp. 7-4, 7-10, 7-11
[6] M.R. Lyu, Handbook of Software Reliability Engineering, IEEE Computer Society, p.71-117, 1996.