CASRE - An Easy to Use Software Reliability Measurement TooI

Allen P. Nikora, M.S.
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Mail Stop]25-233
e-mail :
bignuke@spa1.jpl.nasa.gov

Michael R. Lyu, Ph.D.
Bellcore
445 South Street
Box 1910
Morristown, NJ 07692-1910
e-mail. :
lyu@bellcore. com

William H. Farr, Ph.D
Naval Surface Warfare Center
Code B-10
Dahlgren, VA 22448
e-mai 1 :
wfarr@s850.mwc.edu

Abstract:

Many different software reliability model.s have been proposed since the publications of the first models in 1971 [1,2]. Sufficient work has been done to demonstrate that these methods can be used to monitor reliability growth over a useful range of software development projects. However, the use of software reliability measurement as a management tool is not as widespread as it might be. Part of the problem is that there is no known method for identifying the model most applicable to a development effort. prior to test [2]. This can be mitigated, however, by executing several models at once and using statistical. methods to identify the model that is most likely to produce accurate results [1,2]. Recent work has also shown that more accurate predictions can be produced by combining the results of several models in a linear fashion [3] or by recalibrating a model based on its predictive bias [4].

Another major issue in software reliability measurement is the ease-of-use of currently available tools. Several tools are currently available [5], but many of them have properties making them difficult for non-specialists to use. For instance, the primary outputs of several tools are in the form of model. parameter values. Without knowledge of the mathematical details of the model s, the results cannot be interpreted. Although software developers and managers may have a general knowledge of failure rates and mean time to failure, there will be many who are not familiar with the details of software reliability models, and have no time to become acquainted with them. Because of this, many of the currently-available tools will be of limited use to these individuals.

Operating the tools may also be difficult. Many tools have command-line interfaces, and do not take advantage of currently-available high resolution displays and pointing devices that would allow the construction of menu-driven or direct-manipulation user interfaces. Command-line interfaces can make it more difficult for users to remember the specific steps required to accomplish a task, and also make it easier for operations to be performed out of sequence. A pull-down menu or direct-manipulation interface, on the other hand, can assist users by explicitly listing the available operations and restricting their use to circumstances in which meaningful results would be obtained.

Finally, most currently available tools display their outputs in tabular form or in character-based graphics. In measuring software reliability, it is useful to be able to see high resolution displays of failure rate curves, reliability growth curves, and the cumulative number of failures. Displays such as these would allow managers to easily predict future failure behavior and see their relationship to the time and effort required to achieve the failure rate or reliability required for the software to be released.

We have implemented a software reliability measurement tool., CASRE (Computer Aided Software Reliability Estimation) that addresses these ease-of-use issues. CASRE incorporates the mathematical modeling capabilities of the public domain tool SMERFS [6], and is being implemented in a Microsoft Windows environment. The command interface is menu driven; selective enabling and disabling of menu options guides users through the selection of a set of failure data, execution of a model., and analysis of model results. Input to the models is simultaneously displayed as text and as a high-resolution display that can be controlled to let users to view the data in several different ways (e.g. interfailure times, cumulative number of failures) . Model predictions and statistical evaluations of a model's applicability may be superimposed on the plot of the data used as input to the models. CASRE also incorporates our earlier findings that prediction accuracy may be increased by combining the results of several models in a linear fashion. Users can define their own model combinations, store them as part of the tool's configuration, and execute them in the same way as any other model.

This type of tool would be particularly useful to software development organizations searching for ways to more effectively manage their development resources. For many projects, software reliability measurement has been shown to be an effective way to manage resources during the testing phases [1]. Specifically, it can help managers answer the following questions:

1. When will the software be ready for release?

2. How much time and effort will be required to complete testing?

3. What will the impacts to users be if the desired reliability is not achieved?

Since CASRE has been designed with the non-specialist in mind, we believe it will wider acceptance among managers and developers than tools requiring detailed knowledge of the models.

References

1. J. D. Muss, A. Iannino, K. Okumoto, Software Reliability: Measurement, Prediction, Application, McGraw-Hill Book Company, New York, New York, 1987.

2. A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of Competing Software Reliability Predictions," IEEE Transactions on Software Engineering, vol.. SE-12, no. 9, pp. 950-967, Sep, 1986.

3. M. R. Lyu and A. Nikora, I'Applying Reliability Models More Effectively," IEEE Software, vol. 9, no. 4, pp. 43-52, July, 1992.

4. S. Brocklehurst, P. Y. Chan, B. Littlewood, and J. Snell, "Recalibrating Software Reliability Models," IEEE Transactions on Software Engineering, vol. SE-16, no. 4, pp. 458-470, April, 3990.

5. G. E. Stark, "A Survey of Software Reliability Measurement Tools," proceedings of the IEEE International Symposium on Software Reliability Engineering, pp. 90-97, Austin, TX, May 15-17, 3991.

6. w. H. Farr and O. D. Smith, "Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS) User's Guide," NSWC TR 84-373, Revision 2, March, 1991..

Acknowledgements

Part of the work performed in this abstract was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a NASA contract. The implementation of CASRE is being supported by the Air Force Operational Test and Evaluation Center (AFOTEC) under Task Order RE-182, Amendment 655, proposal 80-3417.