ESTIMATING AND PREDICTING ERROR DETECTION TRENDS AND RELATIVE MANPOWER UTILIZATION*

JAMES D. ARTHUR
The Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

e-mail:
arthur@vt.edu

RUTH K. STAPKO
Unisys/GSFC
Bldg. 6, Code 300.1
Greenbelt, MD 20771

e-mail:
rstapko@pop300.gsfc.nasa.gov

LINDA H. ROSENBERG
Unisys/GSFC
Bldg. 6, Code 300.1
Greenbelt, MD 20771

e-mail:
Linda.H.Rosenberg.1@gsfc.nasa.gov

LAWRENCE E. HYATT
NASA Goddard Space Flight Center
Bldg. 6 Code 300.1
Greenbelt, MD 20771

e-mail:
Lawrence.E.Hyatt.1@gsfc.nasa.gov

ABSTRACT

The work described in this paper outlines an investigative effort focusing on estimating the number of errors remaining in a software product and the effort required to detect those errors. More specifically, we provide a description of a computational model that, through a non-linear optimization process: (a) determines an error detection trend in a given development effort, (b) provides an estimate of the errors remaining in the product and the time required to detect those errors, and (c) relates the error detection effort to current and future manpower requirements. The model is simple to use, requiring only error counts and detection dates as input, and is grounded in the time-tested theory underlying Musa's Basic Reliability Model [1] and the Rayleigh Manpower Utilization Curve [2]. The model has been developed under the auspices of the Software Assurance Technology Center (SATC) at NASA/GSFC. Data sets from various NASA projects were used in formulating and tuning the model. For research purposes, the model was initially developed within an Excel framework; a more robust implementation is currently underway.

Area Keywords: Software Engineering, Error Estimation and Prediction, Modeling and Analysis

INTRODUCTION

Within a software development effort, and in particular during Integration and Testing, two questions are frequently asked: *"To what extent is the product error-free?*", and "*How much time and effort is required to detect and remove the remaining errors?*". Clearly, neither question can be answered with absolute certainty. Nonetheless, the ability to answer these questions with some acceptable level of confidence is highly desirable. First, knowing the extent to which a product is error-free and the critical nature of its application, we can judge when it is time to terminate testing and begin using it. Secondly, in the event that errors are judged to be present, we can perform a cost/benefit tradeoff analysis to estimate when the product will be ready for use, and at what cost. Unfortunately, current models that help answer questions like the above often require diverse sets of data elements that are often difficult to capture, and/or contain computations that are sensitive to data accuracy [3]. Moreover, most (if not all) models reported in the literature address only question 1 or question 2. We have found none that addresses both questions concurrently. The error trending and manpower utilization model described below relates manpower requirements to error detection using a minimal input data set and a relatively simple, yet robust, set of computations. The results described in the remainder of this paper builds on the work initiated by Waterman [4].

MODEL OVERVIEW

The error trending and manpower utilization model uses Musa's Basic Reliability Model as the computational foundation. Musa's Basic Reliability Model is defined by:

* Funding and direction for the work presented in this paper has been provided by the Software Assurance Technology Center (SATC) at NASA/GSFC.

$$f_M(t) = V \bullet \left(1 - e^{-g(t)}\right) \quad \text{where } g(t) = \left(\frac{L}{V}\right) \bullet t,$$

t denotes time, and L and V are constants denoting the initial failure rate and total system failures, respectively.

From an error trending perspective, our model employs a straightforward implementation of the Musa Model to compute a nonlinear approximation to the cumulative errors found to date. Rather than using the probabilistic approach to determine model parameters, however, we chose to seek values for L and V through an "iterative" process that employs a non-linear optimization technique to fit the model-generated cumulative error distribution curve to the actual cumulative error distribution curve. A sum-of-difference-squared is used to determine the "best fit." Using the computed values for L and V, other functions defined by Musa allow us to estimate: (a) the number of errors remaining in the product, and (b) the time needed to detect the remaining errors.

To relate manpower utilization to detecting the errors remaining in the product, we use a modified version of the Musa Model where the linear term in the exponential is replaced by the integral of a Rayleigh function:

$$f_R(t) = K \bullet \left(1 - e^{-h(t)}\right) \quad \text{where}$$

$$h(t) = \int_1^t \left(R_1 \bullet \left(\frac{t}{R_2}\right) \bullet e^{-\left(\frac{t^2}{2 \bullet R_2^2}\right)} \right),$$

t denotes time, and K, R_1, R_2 are constants denoting total system errors, and curve form parameters, respectively.

Justification for replacing g(t) with h(t) in f_M *and* expecting feasible results to ensue is based on the following:

(1) In the Musa Model g(t) is a non-decreasing function. The speed at which Musa's Model converges to a projected total number of product errors is directly proportional to the slope of g(t) -- the greater the slope the faster the convergence.
(2) In general, an increase in staffing levels translates into a rise in testing activities, and subsequently, a derived increase in the rate at which errors are detected. Moreover, the cumulative manpower utilization function is also non-decreasing.

Hence, replacing g(t) with a cumulative manpower utilization function, guarantees that the Musa Model will eventually converge and that that convergence will be related, in some sense, to manpower utilization. Because the Rayleigh curve models manpower utilization [2, 5], we define h(t) to be the integral of the Rayleigh function (i.e., the cumulative manpower utilization function) and use it to "drive" the convergence rate of the Musa Model.

To determine appropriate values for the two Rayleigh curve form parameters (R_1 and R_2), however, we must again employ a non-linear optimization process that iteratively converges to the appropriate set of values. Recognizing, however, that the relationship between the cumulative manpower utilization curve and the cumulative error distribution curve is tenuous at best, we chose a "best fit" criteria for the optimization process that provides the most flexibility in determining values for R_1 and R_2. That is, we use a sum-of-difference-squared to determine "best fit", and concurrently introduce a synchronization constraint that requires the two curves to have a single point in common -- that is, the *projected* errors found to date must equal the *actual* errors found to date.

Having determined appropriate parameter values for the Rayleigh function, both the manpower utilization and cumulative manpower utilization curves can be computed, and then extrapolated to include the manpower needed to detect the remaining errors.

MODEL APPLICATION

The previous section provides a brief overview of the fundamental mathematics and assumptions underlying the error detection and manpower utilization model. The subsections below illustrates how the model is used to predict such quantities.

Ensuring that the Data is Amenable to Estimation

Error estimation and prediction using the Musa and Modified Musa Models require that the error detection rate be decreasing for the reported time period. Intuitively, this stipulation implies that the detection rate will eventually approach 0, and subsequently, that the Musa model will converge to a total product error value. Clearly, the larger the negative slope, the faster Musa's model will converge. The following three steps enable one to determine if a given data set is amenable to estimation:

(1) Plot the actual error detection rate as a function of time,
(2) Compute a linear least-squares fit for the error detection rate, and
(3) Examine the slope of the resulting line.

If the slope is negative, the data can be used for estimation purposes. If the slope is 0 or positive, then the data cannot be used to estimate error trends.

Figure 1 illustrates the results of plotting the error detection rate and a least-squares fit for that rate. Note that the slope of the linear (least-squares fit) line is negative, indicating that the data is amenable to our estimation approach.

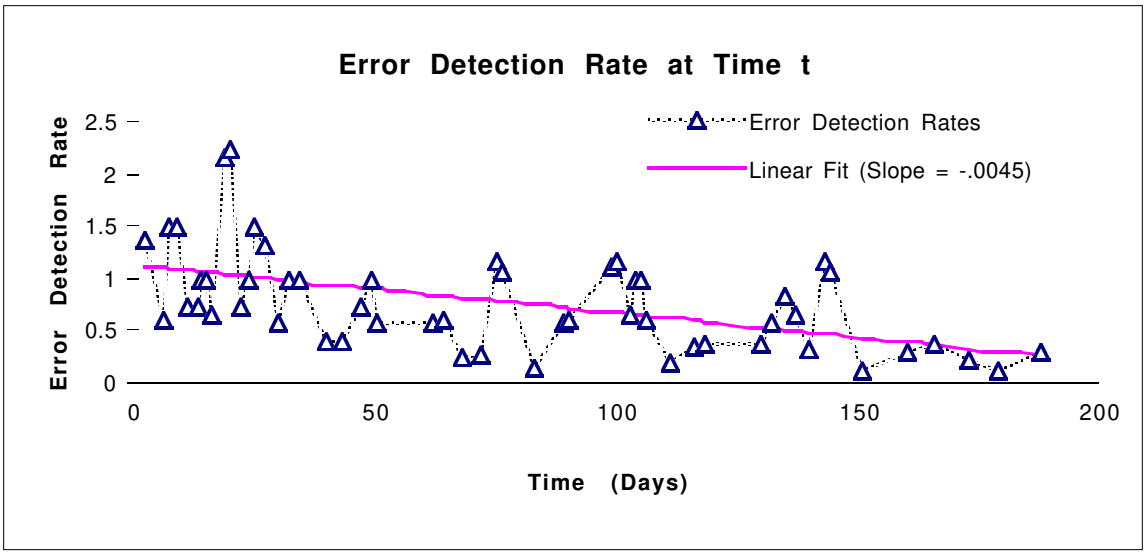

Figure 1
Ensuring that Data Set is Amenable to Estimation

Estimating the Total Number of Errors in the Product

Computing an estimate for the total number of errors in the product requires that we identify values for *L* and *V* (as defined in Musa's Basic Reliability Model) such that the non-linear curve defined by the points (t , $f_M(t)$) best approximate the non-linear cumulative error distribution curve defined by the input data. The value determined for *V* is, in fact, the projected (total) number of errors in the product.

As described earlier, a non-linear optimization algorithm is employed that varies the values of *V* and *L* while seeking to minimize the sum of the squares of the differences between the points defined by (t , $f_M(t)$) and the corresponding points of the cumulative error distribution curve defined by the input data. Figure 2 depicts the results of one such approximation -- the projected errors are those defined by the Musa's Basic Model, the plot of the actual errors detected represents the cumulative number of errors reported over time. Note that both plots pass through (188, 91). The resulting value for V is 99 and provides an estimate of the total number of errors residing in the product.

Estimating the Number of Errors Remaining in a Product and the Time to Remove those Errors

Having determined values for *V* and *L*, Musa provides us with additional formulas for estimating the number of errors remaining in a product and the time required to detect them. The following two modified versions of his formulas compute those two values, respectively.

(1) $$Err_{rem} = \left(\frac{\hat{V}}{\hat{L}}\right) \bullet \left(\left(L \bullet e^{-\left(\left(\frac{\hat{L}}{\hat{V}}\right) \bullet (t_{\max})\right)}\right) - x\right) - \left(Err_{fnd} - f_M(t_{\max})\right)$$

where $\hat{V}$ and $\hat{L}$ represent the final values of *V* and *L* determined by the non-linear optimization process,

t_{max} = total number of days over which the data spans,

$Err_{fnd} = \sum_{i=t_1}^{t_{\max}} Err_i$, the total number of actual errors found to date,

x = 0.001.

(2) $$T_{req}(Err_{rem}) = \left(\frac{\hat{V}}{\hat{L}}\right) \bullet \ln\left(\frac{1}{1 - \frac{Err_{rem}}{\hat{V} \bullet e^{-\left(\left(\frac{\hat{L}}{\hat{V}}\right) \bullet t_{\max}\right)}}}\right)$$

where T_{req} = time required to fix remaining errors, Err_{rem}

ln = natural log function, and

$\hat{V}$, $\hat{L}$, and t_{max} are as defined above.

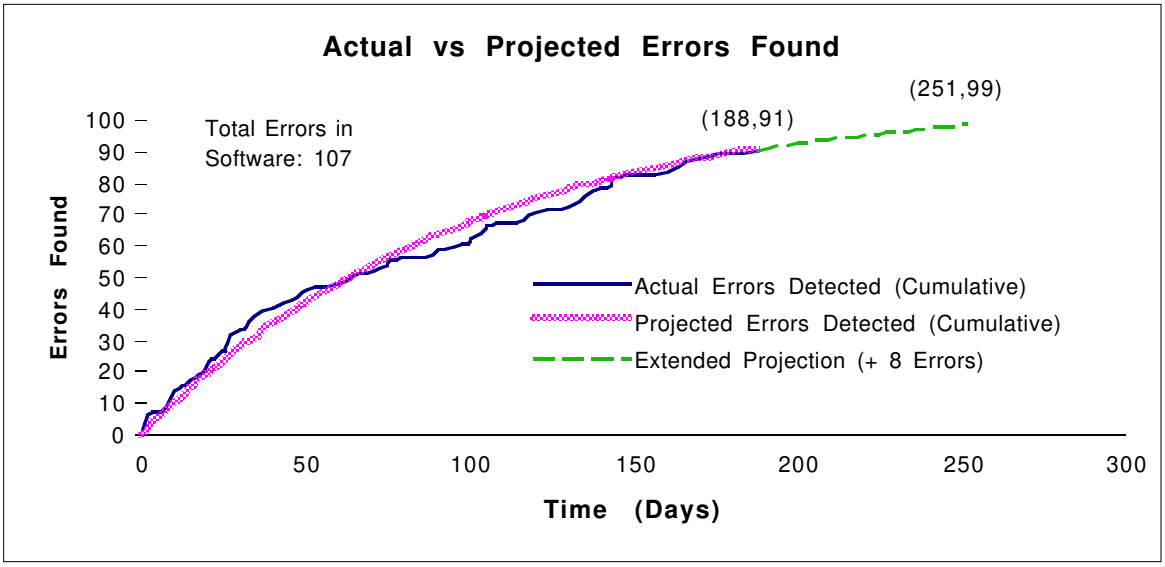

Figure 2
Fitting Projected Errors to Actual

Note that Figure 2 reflects the facts that: (a) 8 additional errors remain in the product, and (b) it will take 63 more days to detect them. Both quantities can be determined by using the identified values of *L* and *V* in the evaluation of T_{req} and Err_{rem}.

Computing the Cumulative Manpower Utilization Curve

Computing the cumulative manpower utilization curve involves a two-step computational process:

(1) Using $f_R(t)$ as defined above, we determine the appropriate values for *K*, R_1 and R_2 through a non-linear optimization process that varies their values subject to the constraints that: (a) the sum of square of the difference between the points defined by $f_R(t)$ and the corresponding points defined by cumulative error distribution curve is minimized, and (b) $f_R(t)$ and the actual cumulative error distribution curve intersect at time t_{max}.

(2) Using the values for R_1 and R_2 determined in step 1, determine the cumulative manpower utilization curve by computing the integral of the Rayleigh curve:

$$CRay(t) = \int_1^t \left(\hat{R}_1 \bullet \left(\frac{t}{\hat{R}_2} \right) \bullet e^{-\left(\frac{t^2}{2 \bullet \hat{R}_2^2} \right)} \right)$$

If we compute CRay(t) over the time frame defined by input data, we can expect to see a curve reflecting the *relative* manpower used over that time period. Using the estimated time to detect remaining errors, we can then extend the cumulative manpower utilization curve to provide a picture of both "where we are" and "where we would like to be" in the error detection process.

Figure 3 depicts a cumulative manpower utilization curve that also has been extended to include the time to detect the remaining errors (63 days). Note, however, that the y-axis is not provided in terms of staffing levels. These values are, in effect, *relative values* and must be interpreted in the context of the project staffing levels known to-date.

Exploring Cost Benefit Tradeoffs

The above sections have described the methods for: (a) estimating how many errors are remaining in the product and how long it will take to detect those errors, and (b) plotting the associated manpower utilization and cumulative manpower utilization curves. These estimates and plots are based on a Musa Model that relies substantially on an exponential function and the "speed" with which it reaches an asymptote. The slower the Musa Model approaches the asymptote, the more the estimated time (and manpower) will be required to detect the errors. This phenomenon reflects the current wisdom that as the number of remaining errors become fewer, those that remain are the more difficult ones to detect. Consequently, the time to find them increases at a non-linear rate.

In non-safety systems the time and manpower required to detect and remove *all* product errors can be cost prohibitive. What is needed, therefore, is the ability to explore a feasible solution that enables one to reach a

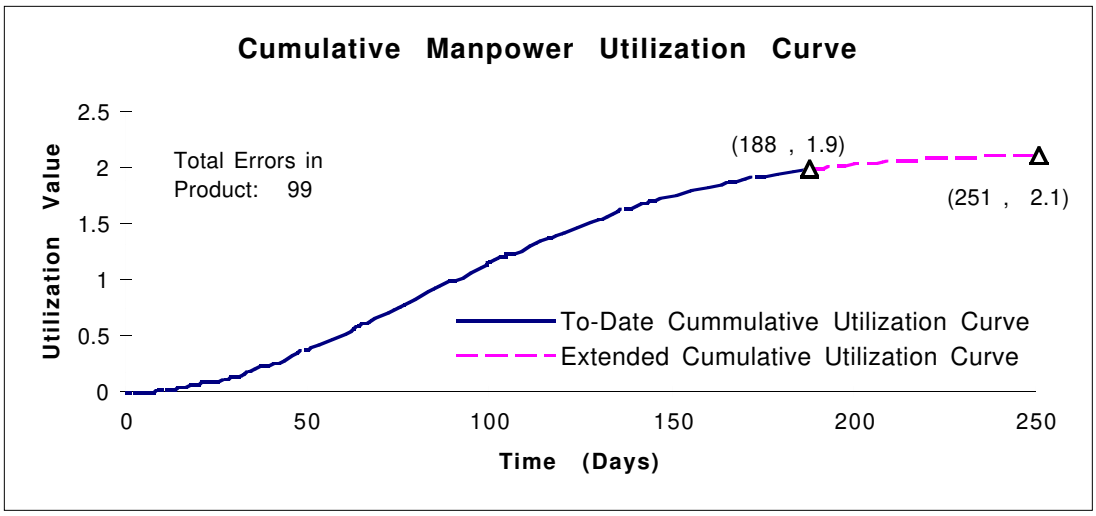

Figure 3
Cumulative Manpower Utilization Curve

balance between costs and leaving errors in a product. To accommodate this need the model allows one to

- determine how many of the remaining errors one desires to remove from the product,
- compute the time it would take to detect those errors, and
- compute the corresponding manpower utilization and cumulative manpower utilization plots.

For example, let X be the number of errors the user desires to *remove* from the product. We can determine the time required to detect those errors by computing:

$$Y = T_{req}(X).$$

We can then visually observe the required manpower by computing and plotting CRay(t) over the interval $[1, t_{max} + Y]$.

CONCLUSIONS

This paper presents a computational model that enables the user to (a) predict the total number of errors in a product, (b) estimate the total number of errors remaining in a product, and (c) determine the relative manpower effort required to remove those errors. Model principles are based on computations defined by Musa's Basic Reliability Model. The data set used to illustrate the model capabilities came from an actual project being developed under the direction of NASA/GSFC. Because time and cost tradeoffs are also of particular interest, the model incorporates additional computational capabilities to address the critical questions: "*What is the relative manpower cost to remove a portion of the remaining errors?*", or from a different perspective, "*When should I stop testing and begin using the product?*".

The model described in the paper has been used with several different project data sets with consistent and confirming results. While some drawbacks still exist in using the model, e.g. applying the model to a data set whose error detection rate is barely negative, the potential benefits of such a model warrant a continued investigation into the evolution of a robust mathematical basis and an implementation that supports that evolution -- both are currently underway.

REFERENCES

[1] Musa, J.D., A. Iannino and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, Professional Edition: Software Engineering Series, McGraw–Hill, New York, NY., 1990.

[2] Norden, P. V., "Useful Tools for Project management," *Management of Production*, M.K. Starr, (Ed) Penguin Books, Inc., Baltimore, MD., 1970, pp. 71-101.

[3] Triantafyllos, G., S. Vassiliadis and W. Kobrosly, "On the Prediction of Computer Implementation Faults Via Static Error Prediction Models," Journal of Systems and Software, Vol. 28, No. 2, February 1995, pp. 129-142.

[4] Waterman, R.E and L.E. Hyatt, "Testing - When Do I Stop?", (Invited) International Testing and Evaluation Conference, Washington, DC, October 1994.

[5] Putnam, L.H., Putnam D.T., and L.P. Thayer, "A Tool For Planning Software Projects", The Journal of Systems and Software, Vol. 5, No. 1, January 1984, 147-154.