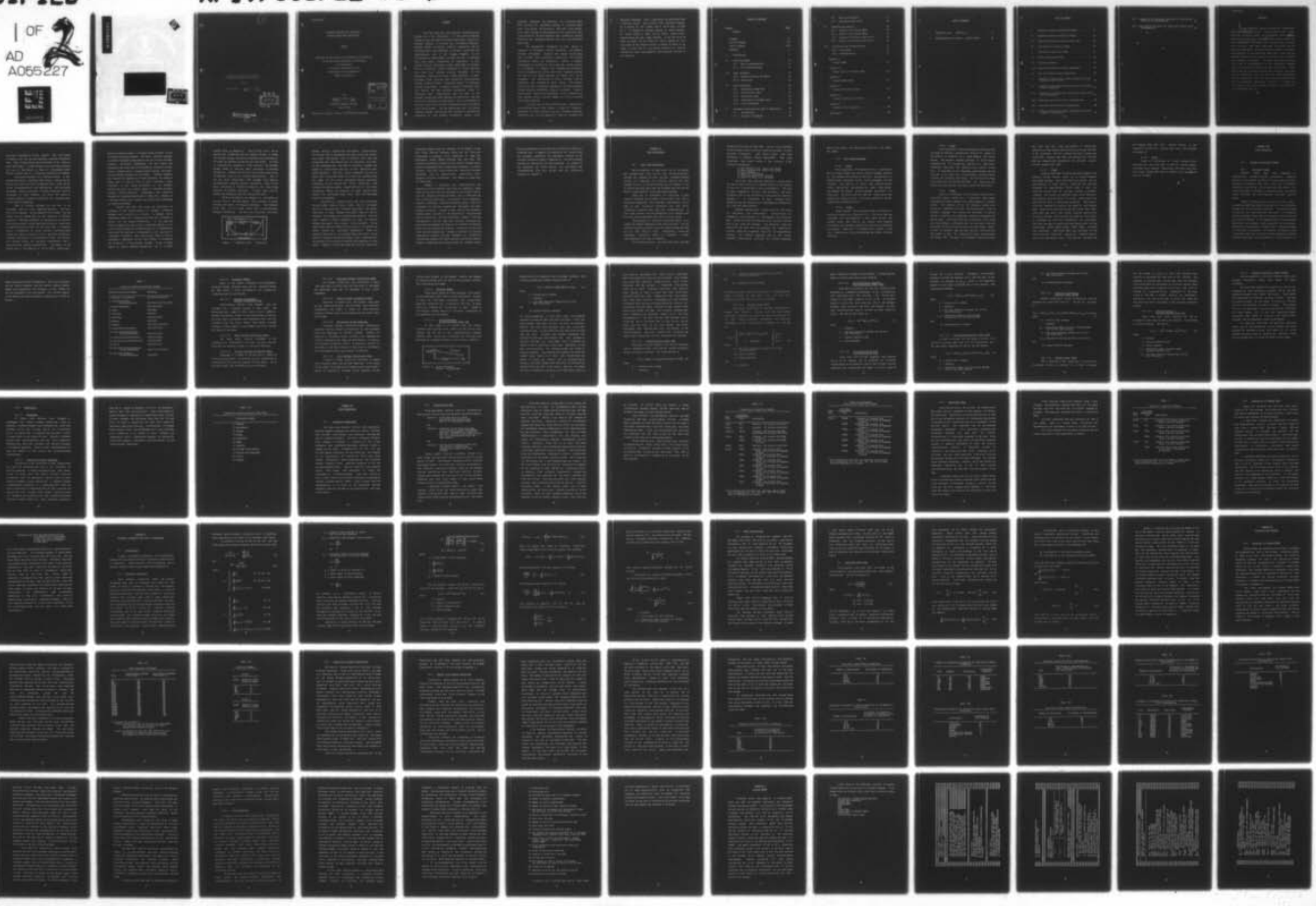
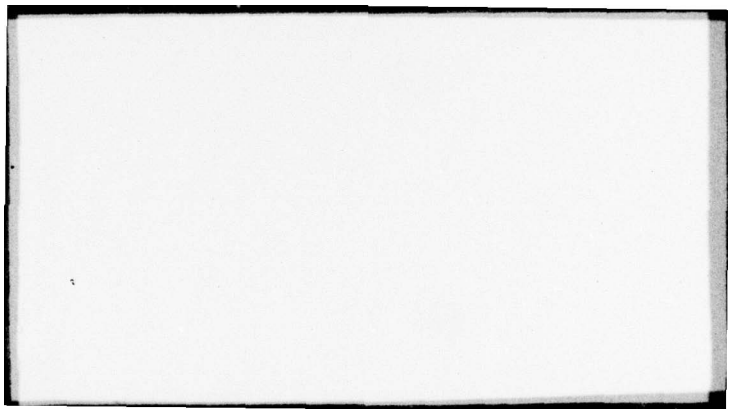
AD-A055 227 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
SOFTWARE RELIABILITY: MODELLING TIME-TO-ERROR AND TIME-TO-FIX.(U)
MAR 78 S G CASTLE

UNCLASSIFIED AFIT/GCS/EE/78-2 NL





(1)

SOFTWARE RELIABILITY: MODELLING TIME-TO-ERROR AND TIME-TO-FIX

THESIS

GCS/EE/78-2 Steve G. Castle
Capt. USAF

DDC
RECEIVED
JUN 19 1978
E

Approved for public release;
distribution unlimited.

78 06 13 165

GCS/EE/78-2

SOFTWARE RELIABILITY: MODELLING TIME-TO-ERROR AND TIME-TO-FIX

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institite of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and or SPECIAL
A

by

Steve G. Castle, B.A.
Captain USAF

Graduate Computer Science

March 1978

Approved for public release; distribution unlimited.

Preface

Over the past ten years Software Reliability has emerged as a new discipline. It has gained attention because of the vast amount of money being spent annually on computer software by private and governmental agencies. Numerous models to predict software reliability and its attributes have been presented by various individuals and organizations. One of the biggest setbacks in the verification of these models has been the lack of software error data bases. Recognizing the importance of this subject, Rome Air Development Center (RADC/ISIS) at Griffiss AFB, N.Y. contracted several companies to collect software error data from software projects developed for the government. RADC has now built a large repository of data. Utilizing files from the RADC repository, discrimination between several software reliability models by hypothesis testing is performed. A computer program is presented which will predict the best software reliability model for particular data. Time-to-error data was used in the software reliability model testing and time-to-fix data was used in the modelling of time-to-fix. Due to limitations of the data, some of the software reliability models considered could not be tested. For the models that could be tested, computer program subroutines were written to estimate the parameters of the software reliability models. After

parameter estimation was completed, the Likelihood Ratio Test was used for hypothesis testing to determine which model best suited the particular data base being tested. Also, model testing was implemented for the time-to-fix data using eleven of the more commonly used probability density functions, the Kolmogorov-Smirnov Test, and the Likelihood Ratio Test.

The theoretical foundation of this thesis is comprised of reliability theory, statistics, and computer software development. However, this thesis avoids going into too much detail concerning statistical and software reliability theory and formulation. Rather, a complete bibliography is provided for those desiring further details on development of formulas and mathematical techniques. With one exception the actual code of the computer programs and much of the output produced for thesis will not be presented. The contribution of these listings would be negligible when compared to their volume. The computer source code used in parameter estimation and discrimination between software reliability models is included. Usage of this program requires only knowledge of the difference in time between errors and the number of errors occurring during each interval.

A project such as this cannot be done without moral and technical support from others. I thank Dr. Thadeus L. Regulinski, my thesis advisor, for his invaluable guidance, assistance, and his willingness to share his academic and

technical knowledge. Also, I appreciate the assistance that I received from Dr. Paul Moranda of the McDonnell Douglas; Dr. E. Nelson, Mr. Tom Thayer, and Mr. Myron Lipow of TRW; Mr. H. E. Willman of Raytheon Company; my thesis sponsor, Mr. Allen Sukert of RADC; and my thesis readers, Capt. Edward Reeves and Capt. James Peterson. A day did not go by where my thesis did not effect my family, whether it be a late night at the computer center or asking for quiet in the house. I would like to give special thanks to my wife and children for their encouragement, understanding, and most of all patience.

Table of Contents

Chapter / Section			Page
	Preface		ii
	List of Figures		vii
	List of Tables		viii
	Abstract		x
I.	Introduction		1
II.	Data Description		8
	II.1	Basic Characteristics	8
	II.2	File characteristics	10
III.	Model Selection		14
	III.1	Software Reliability Models	14
	III.2	Time-to-Fix	30
IV.	Data Preparation		33
	IV.1	Preparation Background	33
	IV.2	Time-To-Error Data	34
	IV.3	Time-To-Fix Data	39
	IV.4	Computation of Elapsed Time	42
	IV.5	File Limitations	43
V.	Parameter Estimation and Test of Hypothesis		45
	V.1	Introduction	45
	V.2	Parameter Estimation	45

V.3 Model Restrictions 51
V.4 Likelihood Ratio Test 52

VI. Execution And Results 56
VI.1 Execution of Program SRMOD 56
VI.2 Results from Program SRMOD 57
VI.3 Execution of Program Time-To-Fix . . . 61
VI.4 Results from Program Time-To-Fix . . . 62

VII. Conclusions and Recommendations 71
VII.1 Conclusions 71
VII.2 Recommendations 74

Appendix A
Program SRMOD 79

Appendix B
Sample Input for Program SRMOD 110

Appendix C
Program SRMOD Output 112

Appendix D
Program Time-To-Fix Input 153

Appendix E
Program Time-To-Fix's Output 155

Appendix F
Software Error Categories 159

References 166

List of Figures

1 "Bathtub" Curve Ref([7]:3) 4

2 De-Eutrophication Process Ref([21]:482) . . . 20

List of Tables

I	Potential Software Reliability Models	17
II	Selected Software Reliability Models	29
III	Time-To-Fix Density Functions (Ref [18])	32
IV	Time-To-Error Input File Table	37
V	Time-To-Fix Input File Table	41
VI	Model Selection By Group	59
VII	Selection Summary	60
VIII	Modelling Time-To-Fix Within Categories	65
IX	Data Point Spread Within Categories	66
X	Modelling of Time-To-Fix Within Categories by the Number of Data Points	66
XI	A Sample of Distributions Selected For Time-To-Fix Within Categories	67
XII	Distribution Selection For Time-To-Fix Within Major Error Categories	67
XIII	Modelling Time-To-Fix Within Subcategories	68
XIV	Data Point Spread Within Subcategories	68
XV	Modelling Time-To-Fix Within Subcategories by the Number of Data Points	69

XVI A Sample of Distributions Selected For Time-To-Fix Within Error Subcategories 69

XVII Distribution Selection For Time-To-Fix Within Error Subcategories 70

Abstract

Various competing software reliability models were tested ~~utilizing~~ using time-to-error data extracted from a large data base provided by Rome Air Development Center, (RADC), Griffis Air Force Base, ~~New York.~~ (N.Y.) Only those models for which appropriate information was available from the RADC data base were used. Each model that was tested had its parameters estimated and discrimination between the models was performed by the Likelihood Ratio Test. Code for the computer program used in parameter estimation and Likelihood Ratio Testing is included. The same large data base was also used in determination of the probability density functions governing the random variable time-to-fix. Time-to-fix data was extracted from the data base and was sorted by error categories. Eleven of the more common probability density functions, along with the Kolmogorov-Smirnov Test and the Likelihood Ratio Test, were used to isolate the distribution that best modelled software time-to-fix for each particular data set.

SOFTWARE RELIABILITY: MODELLING TIME-TO-ERROR AND TIME-TO-FIX

Chapter I

Introduction

Each year billions of dollars are being spent on the contractual purchase and in-house development of computer software packages. The future does not look much brighter, especially considering that the cost of software is soaring. The cost of planning, coding, verifying and maintaining software has far exceeded the cost of computer hardware. Unfortunately, more often than not, the final software product is far too unreliable.

> "The world's most carefully planned and generously funded software program was that developed for the Apollo series of lunar flights. The effort attracted some of the Nation's best computer programmers and involved two competing teams checking the software as thoroughly as the experts knew how to make it. In the aggregate about $600 million was spent on software for the Apollo program. Yet, almost every major fault of the Apollo program, from false alarms to actual mishaps, was the direct result of errors in computer software (Ref [26]:52)."

This thesis addresses the problem of unreliable software. The first objective of the thesis is to test the numerous competing software reliability models. Many models

have been presented by various authors. Each has claimed his model to be the one that descibes software reliability best. Most of these models were formulated on the basis of small data bases. A large software error data base provided for use in this thesis by Rome Air Development Center (RADC), Griffiss AFB, N. Y. permitted testing of the models.

The second objective is to determine which, if any, probability density functions (pdf) govern time-to-fix for entire error files and error categories within files. As far as can be determined no such analysis has been previously performed. Further discussion of these objectives will be enumerated in later chapters, but first the basics of reliable software, software reliability, and software time-to-fix must be understood.

What is reliable software and how can it be produced? This is a very important question. The first part of the question can be answered fairly simply and the answer can be agreed upon by the majority of software engineers. "Reliable software" is considered to be software that produces output within specified tolerance limits over a specified period of time. On the other hand, a "software error" is considered to be any software condition which causes the output to deviate from the specified tolerances. "Software reliability" is the probability that the software will perform within the specified requirements over a specified time interval (Ref[24]:5-2). The answer to the second part of the question is not so easily agreed upon.

There are several schools of thought being promoted on how to produce reliable software. One school believes strongly in better development techniques as the key to more reliable software. Structured programming, top down design and modularity are terms that are expounded by followers of this school. Another school proposes better testing and proof-of-correctness techniques as the answer. Computer hardware has long had the capability of fault tolerance, the state of being able to recover from an error condition. Fault tolerance is now being considered in the software engineering field as a viable means of obtaining reliable software (Ref[17]:230-232). Undoubtedly, each school of thought has something significant to offer and a manager of a software project must consider all options when purchasing or producing software.

Despite the best laid plans and procedures as promoted by the various schools of thought, a software package with no errors is a rarity. A program can be repeatedly tested, but "Testing only shows the presence of errors, not the absence (Ref[6]:33)". In the past the software manager has had no data to determine whether or not a new software system is reliable enough to be implemented, other than the instinct of a programmer or the assurance of a software contractor. How does he know that he has not purchased a nightmare? He has had no means of estimating the reliability of the software package. It was a simple matter to define software reliability, but it is quite

another thing to measure it. Even if there was a way to measure the reliability and the manager decided to accept the software system, how does he determine the mean-time-to-fix after the software system has been delivered? In order to ascertain the answers to these questions the software manager needs to know the basic principles of software reliability. Software reliability could someday easily play an important role in his decision making policy. But, it was not until recent years that attempts were made to mathematically descibe software reliability. Some of the basic principles of software reliability are still being disputed by software engineers. A closer look into its history and background may help explain this fact.

Software reliability has emerged as a discipline of its own only in the past decade. Early attempts to model software were based on hardware principles. The "bathtub" curve (Fig. 1) is a model often used in demonstrating hardware reliability. The curve can be divided into three

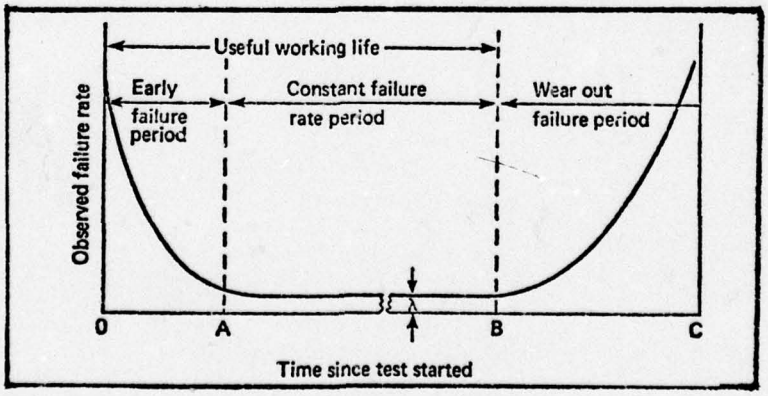


Figure 1. "Bathtub" Curve Ref([7]:3)

phases: burn-in, normal-life, and wearout. During burn-in the failure rate is decreasing with time; during the normal-life phase the failure rate is constant with time; and during wear-out the failure rate is increasing with time. As time increases over the life of a software package, the number of errors found per unit of time (detection rate) should decrease. This most closely resembles the burn-in phase. But, it has been shown in at least two projects that the number of new errors introduced during correction procedures is not trivial (Ref [7]:18, [24]:4-45). Some software reliability models assume there are no new errors introduced. Before any criticism is waged against these models, the environment under which they were developed should be recognized.

Collection of software errors and their attributes has only taken place in recent years. Many of the models had to be developed and validated using small data bases. There were no large software error data bases to use for analysis. This brings out an interesting question. From the large number of models available, how does the software manager know which one to choose for his project? Is one model better than all others for all data or are particular models suited for particular situations? These are questions that have to be answered before any faith can be placed in software reliability models. These questions will be answered in the process of meeting the objectives of this thesis. However, a large data base was required before any

reasonable answers could be obtained. In an attempt to get full value from its software dollar, the Air Force has established a software error repository at Rome Air Development Center (RADC), Griffiss AFB, New York. The repository presently consists of over 27,000 software error reports accumulated from at least seven different Air Force projects. Each error record within the repository contains attributes such as date-of-error, date-of-fix, error category, and error criticality. This thesis is based upon the above data.

Chapter 2 describes the characteristics and background of the different files from the repository. These files are the best that are available at the present time to test the software reliability models and to determine the pdfs governing time-to-fix for software error categories. Chapter 3 identifies software reliability models that were considered for testing purposes and delineates the model selection criteria. Details on the time-to-fix validation program, which was used in the selection of the governing pdfs of time-to-fix, are also presented. Chapter 4 includes information on how the data from the data repository was extracted and prepared for usage by the software reliability models and the time-to-fix program. Limitations of the data files are also pointed out. Chapter 5 outlines the program construction process that was necessary to validate and discriminate between the usable software reliability models. Chapter 6 summarizes the enormous amount of computer output

that was generated by this study and presents the results in condensed form. A summary is presented of the results from the parameter estimation and hypothesis testing of the software reliability models. Summaries of the distributions selected for time-to-fix are also presented. In support of the objectives set forth earlier, the conclusions and recommendations that were derived from the results are contained in Chapter 7.

Chapter II

Data Description

II.1 Basic Characteristics

Man's reluctance to admit error may be one reason why software error data has not been collected earlier. Besides, the computer has been the best scapegoat mankind has had in a long time. It's been too easy to say, "The computer did it." If the ever-rising cost of software is to be halted, man must admit that the mistakes were his and not those of some mythical, error-generating computer.

There was no precedence for collecting a large volume of software errors before, and the RADC files mark a significant beginning. Most assuredly, improvements can be made in data collection procedures, and each error-collecting contractor made many such recommendations. In fact a new, shorter, and more meaningful error category list has since been established (Ref [24]: 3-18 thru 3-20).

The selection of software reliability models to be tested was largely dependent on the data available. Not all software reliability models could be tested because the data did not lend itself to them. Consequently, before any discussion concerning models is presented, a basic understanding of the data is required.

As mentioned earlier, six data files from the RADC

repository were used as input data. The six files represent 27,000 software errors from five different projects by four different contractors. Each record within each file represents a Software Problem Report(SPR). With a few exceptions, each record within a file consists of the following information:

1) Date software error report was opened
2) Date software error report was closed
3) Error category
4) Degree of criticality
5) Test phase when error occurred
6) Module in which error occurred.

One of the most important aspects to realize about the data is that it was not collected solely for the purpose for which it is being used in this investigation. The data is being accepted as the best data available. The error categories used in the collection of this data are listed in Appendix F. It represents 20 major software error categories and 164 software error subcategories (Ref[24]:3-3 thru 3-8) and (Ref[7]:45-50).

Seperate subfiles were created from the RADC files for time-to-fix and time-to-error. Time-to-error was computed as the difference in time between two successive SPRs. Within each SPR, the date the error occurred and the date it was fixed was recorded. From these two dates the time-to-fix for each error was computed. The beginning of the test phase was considered "day one" for computational purposes. All subfiles beginning with an "EF" prefix represent time-to-error files and all subfiles beginning

with an "FF" prefix are time-to-fix files (e.g. EF1, EF4A, FF1, FF5G).

II.2 File characteristics

II.2.1 File1

File1 represents 4519 software errors collected by TRW on a large command and control software package (Project 3). The software was written by TRW and a subcontractor in JOVIAL/J4 and contained 115,346 source statements and 249 routines. The majority of errors were detected during formal testing over approximately an eight month period. The command and control software package was tested in a batch mode. The entire file was used for time-to-error and time-to-fix analysis and the files were called EF1 and FF1 respectively (Ref[24]:2-2,2-3).

II.2.2 File2

File2 contains the operational data for Project 3, which was referred to in Section II.2.1. The error data was collected over approximately a nine month period. The records within this file do not contain error category information and therefore time-to-fix could not be extracted by category. There were 451 software error records in the file. Time-to-error data was extracted and called EF2 (Ref [24]:2-2).

II.2.3 File3

File3 consists of records representing SPR data from a project by TRW and a subcontractor (Project 2). Again, it was written in JOVIAL/J4 for a large command and control system. The software package consisted of 96,931 source statements and 173 routines and operated in a batch mode. Originally there were 436 software error records in the file. It consisted of two independent modifications and the file had to be split into two autonomous subfiles before time-to-error analysis could begin. The time-to-error files thus formed were called EF3A and EF3B. For time-to-fix the entire file was used and was simply called FF3 (Ref [24]).

II.2.4 File4

File4 is a collection of 2165 software errors from a real-time control system for a land-based radar system developed by Raytheon Company over approximately a 37 month period. The majority of the 109 software modules were written in JOVIAL/J3, with the remaining modules coded in assembly language. The project was developed in several builds. A build was an independent computer development and coding phase. Sufficient data was not recorded in the files and the necessary information was not available from the contractor on how to break out the file into builds. Again, it must be stressed that the data was not collected with this study in mind. The time-to-error data was extracted and called EF4. Two means of collecting time-to-fix data

were used. One used a 365 day calendar to compute the difference in time between when the error was found and when it was corrected. The file created from this process was called FF4A. Another file, FF4B, was created from fix data that was computed and recorded by the contractor using a workday calendar (Ref [29]).

II.2.5 File5

File5 was generated by Boeing Aerospace Company from Software Problem Reports (SPRs) on an avionic software development package. There were 2036 error reports over an approximately 27 month period. JOVIAL/JB was used and accounted for approximately 40,000 lines of code. The remaining 80,000 lines of code were written in assembly language. The project was developed using seven functions, each function being an entity of its own. Time-to-error was extracted for each of these seven functions. The time-to-error was computed by counting the number of days between SPRs. These seven files were called EF5A, EF5B, EF5C, EF5D, EF5E, EF5F, and EF5G. Also, Boeing computed time-to-error by using a formula that totalled the number of daily hours of equipment use since the beginning of the test phase. This data was used to extract the time between SPRs and the files EF5AH, EF5BH, EF5CH, EF5DH, EF5EH, EF5FH, and EF5GH were generated. Time-to-fix data was extracted for the entire file and called FF5. Boeing also computed time-to-fix by the hundredths of hour. This data was extracted and

was labeled FF5H (Ref [7]). Further details on the preparation of this file and the other files are presented in Chapter IV.

II.2.6 File6

File6 is a collection of 11,728 software errors collected over the years 1967 to 1971 for Project Apollo, the manned lunar landing. Regretfully, this data base could not be used, because the dates of software error occurrences were not recorded.

Chapter III

Model Selection

III.1 Software Reliability Models

III.1.1 Potential Models

Numerous models have been suggested to stochastically represent software reliability. Listed at the end of this section in Table II are the models and their respective references that were considered for possible inclusion in this study. By no means does the absence of a model from Table II indicate its relative value. The models listed are those for which adequate literature could be found.

Software reliability can be measured in two ways: reliabilty growth and interval reliability. The distinction between the two is extremely important. Reliability growth is just what its name indicates, the growth in reliability as time increases, where time is measured as starting on the first day of testing. It is expected that as time proceeds, more errors are removed, thus increasing the reliability. Ideally all errors should be found and a reliability equal to 1.00 would be achieved. For interval reliability, time is measured since the last error and the probability density function represents the distribution of time-to-error. An

interval is the time between two successive errors. The probability that an error will be found (unreliability) increases as the time since the previous error increases. Thus, the reliability is high in the beginning. But, as the software package or program is exposed longer and longer since the previous error, the probability of the package performing to specifications (reliability) decreases. Mathematically this can be stated by

$$R(x,T) = P[X(t) = 1, T \leq t \leq T+x]$$

where X(t) = 1 if the software is operating at time "t" and X(t) = 0 otherwise. Inherent in this formulation is that at "T", the starting point of the interval, the software is operating and the duration of the interval is "x" (Ref [1] : 8). Knowledge of interval reliability would enable the software manager to ask what the reliability of a package was after being exposed to "x" number of hours or days. Reliability growth would enable him to release a package when it reached a predetermined level of reliability. The uses and combinations of these two types of reliability are endless. But the crucial point being made is their difference, and that interval reliability and reliability growth are two different concepts.

The software reliability models listed in Table I were based upon many different principles. In order to produce meaningful results, testing of the competing software reliability models was narrowed to only include

those based upon interval reliability. With this constraint and the lack of adequate data for certain models, several models were eliminated from inclusion in this thesis, as will be discussed in the next section. For further details on the models that were eliminated, references are given in Table I.

Table I

Potential Software Reliability Models

Software Reliability Model	Reference
1) Shooman's Exponential	Ref([22],[23],[27])
2) Jelinski-Moranda De-Eutrophication	Ref([8],[14],[15],[23])
3) Schick-Wolverton	Ref([30],[23])
4) Weiss	Ref([28])
5) Corcoran	Ref([3])
6) Markovian	Ref([23],[25])
7) Nelson	Ref([16])
8) Weibull	Ref([23],[20],[27])
9) Lipow-Schick-Wolverton	Ref([12],[23])
10) Lipow-Jelinski-Moranda De-Eutrophication	Ref([12],[23])
11) Modified-Schick-Wolverton	Ref([12],[23])
12) Reliabilty Growth Modelling	Ref([9],[23])
13) Mills	Ref([21],[9])
14) Jelinski-Moranda-Geometric De-Eutrophication	Ref([14],[15],[23])
15) Modified Geometric De-Eutrophication	Ref([23])

III.1.2 Eliminated Models

Seven of the models considered were eliminated, primarily because necessary data was not contained within the RADC files. A brief discussion on the reasons for eliminating each is given below.

III.1.2.1 Shooman's Exponential Software Reliability Model

Unfortunately, several data elements were not available for a complete testing of this model. The processing rate, number of total hours in test, number of machine instructions, number of runs terminating in failure, and total time of successful and unsuccessful runs were not available from the files. Rather than making a biased estimate of these inputs, it was thought best not to include this model in the study.

III.1.2.2 Weiss Software Reliability Model

The Weiss model required knowledge of the probability of correcting an error and the test time for successes and failures, neither of which was available.

III.1.2.3 Corcoran Software Reliability Model

Knowledge of N number of trials with N_i number of failures and N_j number of successes and the probability of correction of the error was required for this model. As in the Weiss model, the information was not available.

III.1.2.4 Markovian Software Reliability Model

The primary reasons for the elimination of this model were that it did not fall into the scope of the study and sufficient time was not available to give a fair assessment.

III.1.2.5 Nelson Software Reliability Model

There was no control over the original test input for the modules from which the data bases were collected. Consequently, the number of inputs for which execution failures occurred could not be examined and this model could not be used.

III.1.2.6 Reliability Growth Modelling

As stated earlier, reliability growth modelling is concerned with the growth of reliability as time increases since the beginning of software testing. However, the majority of the models listed in Table I are dependent on interval reliability and model the distribution of time-to-error between successive errors. As mentioned earlier, it was deemed in the best interest of this study to investigate only those models based upon interval reliability.

III.1.2.7 Mills Software Reliability Model

A number of known errors can be seeded or tagged into a program, whose code has been completed. On the basis of the number of indigenous and induced errors found after a period of testing, an estimate can be computed for the

initial error content of the program. Seeding and tagging of software modules were not done for the projects involved, thus eliminating this model.

III.1.3 Selected Models

Eight models remain for which adequate data exists. In the subsections that follow a brief description is given of each. For each model a detection rate is specified, where the detection rate is defined as the rate at which errors are detected for a given unit of time. Chapter V will present further details on the actual computation of the parameters for the models.

III.1.3.1 Jelinski-Moranda De-Eutrophication Model (JM)

In this model it is assumed that the initial detection rate is given by $N\phi$, where N is the initial error content, and phi (ϕ) is the proportionality constant (Ref[14]: 327). As an error is removed, the detection rate decreases in a stepwise fashion as can be seen in Figure 2.

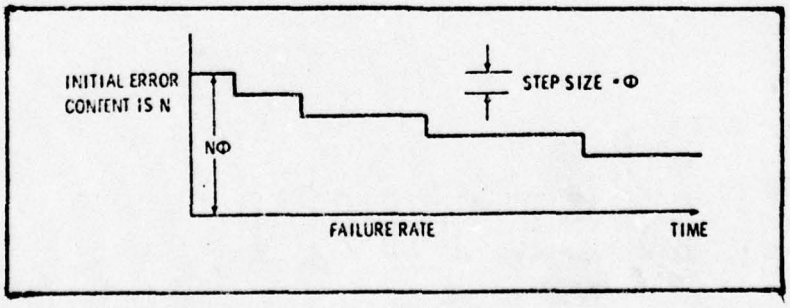


Figure 2. De-Eutrophication Process Ref([21]:482)

Between errors the detection rate is assumed constant. The resulting probability density function (pdf) is

$$f(t_i) = \phi[N-(i-1)]EXP\{-\phi[N-(i-1)]t_i\} \quad (1)$$

where

N = initial error content

i = interval

t_i = the time separation between the (i-1)st and "i"th error

and

ϕ = proportionality constant

Two basic assumptions of this model point out potential problems in using the RADC data with this model. First, the model assumes that there is only one error per interval. However, the data were collected such that errors were recorded by days. On any particular day there could be more than one error. Jelinski and Moranda recognized this conflict in data they used (Ref[8]: 475). They proposed to use each SPR date as a "stop date" and to disregard the number of errors occurring on this date. Regardless of how many errors occurred, the software system has failed to function properly. This approach was accepted for model JM. Another model based on JM takes into consideration the number of errors per interval and is discussed later. Secondly, the model assumes that each error is corrected and removed before the next error occurs. This was not always the case as revealed by the data in the RADC files, where an

error could be discovered, but other error(s) could have been found before the first one was corrected. This dilemma could have been caused by nonfatal errors, which do not completely halt the software. A remedy for this problem could not be found. Intervals could have been based on weeks instead of days, in hopes that most errors were corrected within the week. Inspection of the files showed that there still would be overlapping, and much valuable information would be lost by consolidating the errors. For some files the most common interval of time between "stop dates" would become one week. This produces problems when model parameters are estimated. Taking all of this into consideration, it was felt that the best approach was to retain the day as the basic unit of time and to use the model with the data as is, even though the errors overlapped. This approach does limit the validity of the results produced. Only with further error collection and actual model testing will the full implication be realized.

III.1.3.2 Schick-Wolverton Model (SW)

This model is quite similar to JM, but the detection rate is dependent on the time interval between successive errors, as well as N and phi. The resulting pdf is

$$f(t_i) = \phi[N-(i-1)]t_i EXP\{-(\phi/2)[N-(i-1)]t_i^2\} \quad (2)$$

where

N = initial error content

i = interval

t_i = the time separation between the (i-1)st and the "i"th error

and

ϕ = proportionality constant

The input data were used as "stop dates", disregarding the number of errors at each stop. Also, the data were accepted, realizing that the errors do overlap.

III.1.3.3 Weibull1 Model (WM1)

This model was applied to software reliability by Wagoner (Ref [27]:D-1). Wagoner used the least square method for estimating the scale and shape parameters, but for this thesis parameter estimation was performed differently as will be presented in Chapter V. The pdf is:

$$f(t_i) = \begin{cases} \alpha/\beta\ (t_i/\beta)^{\alpha-1} EXP[-(t_i/\beta)^{\alpha}] & \begin{cases} t_i \geq 0 \\ \beta > 0 \\ \alpha > 0 \end{cases} \\ 0 \quad \text{elsewhere} & \end{cases} \qquad (3)$$

where

t_i = the time separation between the (i-1)st and "i"th error

β = scale parameter

α = shape parameter

and

i = interval

Again, inputs are treated as "stop dates", disregarding the number of errors occurring at each interval.

III.1.3.4 Jelinski-Moranda Geometric De-Eutrophication Model (JMG)

This model is similar to JM, but the detection rate decreases in a geometric progression on the occurrence of each individual error (Ref [14]:327). The initial error content is no longer fixed at N, as in JM. An infinite number of errors can occur over the life of the software package. Only one error can occur per interval of time, thus restricting the data to be used as "stop dates" as before. The pdf for this model is:

$$f(t_i) = Dk^{i-1}EXP(- Dk^{i-1}t_i) \qquad (4)$$

where

i = interval

t_i = the time separation between the (i-1)st and "i"th error

D = initial detection rate

k = ratio constant

III.1.3.5 Lipow-Jelinski-Moranda De-Eutrophication (LJM)

Lipow (Ref [12]:1-7) has suggested that numerous errors in an interval can be accounted for by making modifications to the detection rate of the JM model. The new detection rate incorporates the number of errors found up

through the (i-1)st interval. Throughout the parameter estimation process the quantity (i-1), that was used in the JM model, is replaced by (n_{i-1}), which is the cumulative number of errors found up through the (i-1)st interval. The resulting pdf becomes

$$f(t_i) = \phi(N-n_{i-1})EXP[-\phi(N-n_{i-1})t_i] \qquad (5)$$

where

N = initial error content

i = interval

t_i = the time separation between the (i-1)st and "i"th error

n_{i-1} = cumulative number of errors found through the (i-1)th time interval

and

ϕ = proportionality constant

III.1.3.6 Lipow-Schick-Wolverton Model (LSW)

In order to account for any number of errors in an interval in the SW, Lipow (Ref [11]:1-7) modified the SW in the same manner suggested for the JM. The pdf becomes

$$f(t_i) = \phi(N-n_{i-1})t_i EXP\ [-(\phi/2)(N-n_{i-1})t_i^2] \qquad (6)$$

where

N = initial error content

i = interval

n_{i-1} = cumulative number of errors found through the (i-1)th time interval

t_i = the time separation between the (i-1)st and "i"th error

and

ϕ = proportionality constant.

III.1.3.7 Modified Lipow-Schick-Wolverton Model (MLSW)

Another variation of SW and its detection rate was presented by Lipow (Ref [12]:1-7). The pdf for the model is:

$$f(t_i) = \phi(N-n_{i-1})[T_{i-1}+(t_i/2)]EXP\{-\phi(N-n_{i-1})[T_{i-1}+(t_i/2)]t_i\} \quad (7)$$

where

N = initial error content

i = interval

n_{i-1} = cumulative number of errors found through the (i-1)th time interval

t_i = the time separation between the (i-1)st and "i"th error

T_{i-1} = cumulative time through the (i-1)st error

and

ϕ = proportionality constant

III.1.3.8 Weibull2 Model (WM2)

WM2 is identical to WM1, except that it accounts for the number of errors in interval "i". In order to account

for the number of errors in each time interval, each interval is replicated for each error in the interval. For instance, an interval that was two days long and contained five errors would be represented by five data points, where each data point has an interval of two days. Each error is treated as a separate entity, independent of the other errors in the same or another interval, and represents a data point. The pdf is the same as Eq.(3), but there are many more data points considered in the parameter estimation process.

III.1.3.9 Modified Geometric De-Eutrophication Model (MG)

Sukert (Ref [23]: 19-21) modified the JMG by allowing for more than one error per interval. It was based on Lipow's formulas. The pdf is:

$$f(t_i) = D(k^{n_{i-1}})EXP[- D(k^{n_{i-1}})t_i] \qquad (8)$$

where

i = interval

D = initial detection rate

k = ratio constant

n_{i-1} = cumulative number of errors found through interval "i-1"

t_i = the time separation between the (i-1)st and "i"th error

III.1.4 Software Reliability Model Summary

With the addition of the extra Weibull model, nine software reliability models were chosen for model validation.

The first four models only required "t_i" and "i" for estimation of their parameters, treating "t_i" as the time separation between "stop dates" (SPR dates). These models will henceforth be classified as Group I models. The last five models account for the number of errors per SPR date and will be classified as Group II models. They require "t_i","i", and the number of errors occurring per SPR date. For these models, "t_i" is the time separation between successive SPRs. Table II lists the nine models, their acronyms, and the parameters requiring estimation.

It should be noted that one additional model was found late in the development of this thesis, that could possibly be included in Group II of Table II. It is the Geometric Poisson Model by Moranda (Ref [14]:330). There was not enough time to include the model in this study.

Table II

Selected Software Reliability Models

Group I
Input Requirements - Length of each interval
(number of days between each pair of successive errors)

Model name	Model acronym	Parameters
Jelinski-Moranda De-Eutrophication	JM	N and phi
Schick-Wolverton	SW	N and phi
Weibull1	WM1	α and β
Jelinski-Moranda-Geometric De-Eutrophication	JMG	D and k

Group II
Input Requirements - Length of each interval and the number of errors within each interval

Model name	Model acronym	Parameters
Lipow-Jelinski-Moranda De-Eutrophication	LJM	N and phi
Lipow-Schick-Wolverton	LSW	N and phi
Modified-Lipow-Schick-Wolverton	MLSW	N and phi
Weibull2	WM2	α and β
Modified Geometric De-Eutrophication	MG	D and k

III.2 Time-to-Fix

III.2.1 Background

If within each software error category or subcategory the random variable time-to-fix could be modelled, a software manager could determine the worst case estimate on how long it would take to fix a particular error. Of course, this is assuming that the type of error is known, which is not always the case. Ideally, a software manager could use established pdf's for error categories, determine the worst case estimate, and thereby, eliminate the haphazard guesswork, which so often underestimates fix time and results in cost overrun and strained working relationships.

III.2.2 Time-To-Fix Program Selection

Time-to-fix is the difference in time between when the error was discovered and when it was corrected. It could be measured in central processing time, days, weeks, or any of various ways. The data in the RADC files was recorded by days. No information could be found on other previous attempts to model time-to-fix. A computer program (Program Model) written by T. L. Regulinski (Ref [18]) was used as the basis for determining the pdf that governed the time-to-fix data. Program Model accepts time-to-fix data and estimates the parameters for the probability density functions listed in Table III. After parameter estimation,

each pdf is tested for goodness of fit by the Kolmogorov-Smirnov Test (Ref [5],[10],[13]). Each distribution is either accepted or rejected at a significance level of 0.20. Of those accepted, the Likelihood Ratio Test (Ref [4] and [19]) is used to select the one that best models the particular time-to-fix data set. A modification to Program Model allowed a data file containing two or more different error categories to be sorted by error category and subsequently have a distribution selected for time-to-fix within each category. Henceforth, this new program will be called Program Time-To-Fix.

Table III

Time-To-Fix Density Functions (Ref [18])

Distribution Name
1) Uniform
2) Exponential
3) Weibull
4) Normal
5) Log-normal
6) Gamma
7) Logistic
8) Extreme Value-Largest
9) Extreme Value-Smallest
10) Pareto
11) Laplace

Chapter IV

Data Preparation

IV.1 Preparation Background

Six files were received from the RADC repository. Separate computer programs were written to extract the data from the RADC files because each contractor recorded the data in different formats. The files contained virtually every category in Appendix F. However, not all of the categories were considered legitimate software errors in respect to this study. Categories "L", "T", "U", "V", and "X" were deemed irrelevant for this study and were deleted as the time-to-error and time-to-fix files were being created. Respectively these categories are user requested changes, operator errors, questions, hardware errors, and non-reproducible errors. None of these are true software errors, even though they definitely do impact the development of a software package. Also, many records in File1, File2, File3 and File4 were classified as explanatory software problem reports (SPRs). These records were not representative of an error in the executable code and therefore, were not included in the time-to-error and time-to-fix files.

IV.2 Time-To-Error Data

Three approaches could be taken for extraction of time-to-error data for the nine software reliability models.

Option 1) Treat each file as a whole and extract the time-to-error data in a chronological order.

OR

Option 2) Determine if the files were made up of separate software development entities, independent of others in the file, and then create time-to-error data in a chronological order for each subfile.

OR

Option 3) Isolate every program in each file and create a chronological time-to-error subfile for each program.

Option number three was discarded, because in a large system, any particular program may sit idle for a long period of time. Time between successive errors would be meaningless unless this idle time could be accounted for. Idle times for particular programs could not be obtained from the contractors. Also, much data could not be used, because many programs only had one or two errors recorded. Modelling with such a few number of data points would produce results of questionable value.

A compromise between Option 1 and Option 2 was required. File1, File2, and File4 could not be split into separate entities and were used in their entirety, with time-to-error being computed chronologically for the entire file.

File4 had numerous records which did not contain the date the error was found. As recommended by RADC and the contractor, each of these records arbitrarily had the date that the error was found set equal to the date that the error was fixed. Option 2 was used in the creation of subfiles for File3. During the creation of the software package from which the error data for File3 was collected, two independent software subprojects, called "modifications" by the contractor, were implemented. For each of these "modifications" a time-to-error file was created, each having its time-to-error computed independently from the other's. Similarly, File5 had to be split into seven different subfiles, each representing time-to-error data for a particular functional group of software. Each functional group was independently developed to fulfill a particualar requirement for the master avionics system. Each group had its own software and a time schedule that was unique. All the files created thus far have had time-to-error computed as the difference in days between two consecutive errors. In addition, File5 contained time-to-error recorded and calculated by the contractor for the hourly equipment use time (Ref [7]:20). This data was split into seven separate subfiles, each containing time-to-error data for a specific functional group. A few records within four of these functions had the hourly data recorded incorrectly by the contractor. When the time between consecutive errors was computed it was not always a positive value. This could not

be possible. To correct these few records a linear interpolation estimate between the SPR dates was used to estimate the number of hours between errors.

Since File2 is the operational data for the same project as that used in the creation of File1, it seemed quite reasonable to track the entire project and to create a file which contained data from File1 (test data) and File2 (operational data). File2 was concatenated with File1 and the resulting file was then sorted by SPR date, thus creating error data for a continuing project.

Twenty time-to-error files resulted from the above procedures and are listed, along with a brief description, in Table IV. No mention was made of File6, because it did not contain data on when errors were found. File EF5C is given in its entirety in Appendix B as an example of the data created.

Table IV

Time-To-Error Input File Table*

RADC File	Extracted Time-To-Error File	Description
File1	EF1	Contains 152 records from File1
File2	EF2	Contains 86 records from File2
File1/2	EF1/2	Contains 238 records from the concatenation of File2 onto File1
File3	EF3A	Contains 54 records from MODA of File3
	EF3B	Contains 30 records from MODB of File3
File4	EF4	Contains 435 records from File4
File5	EF5A	Contains 149 records from build "A" of File5 as computed in days
	EF5B	Contains 146 records from build "B" of File5 as computed in days
	EF5C	Contains 51 records from build "C" of File5 as computed in days
	EF5D	Contains 83 records from build "D" of File5 as computed in days
	EF5E	Contains 92 records from build "E" of File5 as computed in days
	EF5F	Contains 102 records from build "F" of File5 as computed in days
	EF5G	Contains 80 records from build "G" of File5 as computed in days

* All time-to-error files do not represent records with error categories "L", "T", "U", "V", and "X" or those that are explanatory in nature.

Table IV (Continued)
Time-To-Error Input File Table*

RADC File	Extracted Time-To-Error File	Description
File5	EF5AH	Contains 131 records from build "A" of File5 as computed in hours
	EF5BH	Contains 130 records from build "B" of File5 as computed in hours
	EF5CH	Contains 49 records from build "C" of File5 as computed in hours
	EF5DH	Contains 75 records from build "D" of File5 as computed in hours
	EF5EH	Contains 109 records from build "E" of File5 as computed in hours
	EF5FH	Contains 90 records from build "F" of File5 as computed in hours
	EF5GH	Contains 76 records from build "G" of File5 as computed in hours

* All time-to-error files do not represent records with error categories "L", "T", "U", "V", and "X" or those that are explanatory in nature.

IV.3 Time-To-Fix Data

Splitting the files, as was done for time-to-error data files, was not necessary for the creation of time-to-fix data files. Time-to-fix is unlike time-to-error in that all of the information required for determining time-to-fix is recorded in one record and requires no knowledge of the previous record. Each contractor had his own software development techniques. For a particular contractor the techniques remained the same from one "build", "modification", or "functional grouping" to the next. Since the same techniques were used within any particular software project it appeared unnecessary to split the files by "modifications", "builds", or "functional groups" for creation of the time-to-fix data files. Combining all of the files into one large time-to-fix file would be quite unreasonable because each contractor developed his software using different techniques. Besides, one of the reasons for modelling time-to-fix was to see if there existed distributions which were the same from one project to the next.

A separate time-to-fix file for File1, File2, File3, File4, and File5 was created, and each record within the new files consisted of the number of days it took to fix the error and the error category from Appendix F. Any error found and fixed on the same day was considered to have a fix time of 0.5 days.

File4 also had time-to-fix computed using a work calendar. The difference between when the error was found and when the error was fixed did not include weekends or holidays. This data was extracted and used to create file FF4B.

In File5 additional data was available, so that a file (FF5H) could be created where time-to-fix was represented by the hundredths of hours to fix (Ref [7]:20).

Six files resulted from the above procedures and are listed along with a brief description in Table V.

Table V

Time-To-Fix Input File Table*

RADC File	Extracted Time-To-Fix File	Description
File1	FF1	Contains 2172 records from File1 using a 365 day calendar
File3	FF3	Contains 219 records from File3 using a 365 day calendar
File4	FF4A	Contains 1254 records from File4 using a 365 day calendar
	FF4B	Contains 1254 records from File4 using a workday calendar
File5	FF5	Contains 1603 records from File5 using a 365 day calendar
	FF5H	Contains 1603 records from File5 as recorded in hundredths of the hour

* All time-to-fix files do not represent records with error categories "L", "T", "U", "V", and "X" or those that are explanatory in nature.

IV.4 Computation of Elapsed Time

Except for the error and fix files that have a suffix of "H", the calendar day is the basic unit of time. Each SPR is represented by a record and within each record all dates were recorded by month/day/year.

A seven year calendar was established to convert each month/day/year entry in a record into the number of days since 1 January 1970. The date 1 January 1970 was selected because it preceeded the earliest SPR and computations could be easily made using it. Once each time-to-error or time-to-fix date was transformed into the number of days since 1 January 1970, the number of days between the time-to-error or time-to-fix occurrences were easily calculated by subtracting the latest date from the earliest date.

A possible legitimate criticism of using the day as the unit of measure would be if the software package sat idle a lot or consisted of one program which was used sporadically. But on a large system, which is in continuous use, the day may be an appropriate unit of measure. CPU time has been suggested as an alternative. In the actual operation of a large computer system or a real-time system, it is sometimes difficult to track CPU utilization accurately. Data were not recorded in CPU time for the RADC files, but in future error collection efforts CPU time will undoubtedly be collected.

Four different contractors built the files and each file was recorded in a different format. Therefore, numerous computer programs were written to delete, extract, sort, and merge the records into identical formats which were usable by a time-to-error program and time-to-fix program.

IV.5 File Limitations

The creation of the files in Table IV was not a "cut and dry" decision. For instance, File4 was collected from a project which consisted of several independent software "builds". Each "build" was developed on a unique time schedule and ideally each "build" should have its own time-to-error subfile, but adequate information was not available in the file or from the contractor to permit segregation by "build".

The start date had to be determined for each time-to-error file. For files EF1, EF2, EF1/2, and a few of the subfiles in File5 the actual date at which formal testing began was specified in the contractor's reports. On other files an official date at which formal testing began ("day one") could not be obtained. Only the detection date of the first error was given in these cases. A date had to be established and three alternatives seemed most logical:

Alternative 1) Use the date of the first SPR as "day one"

Alternative 2) Use the date of the first SPR minus one day as "day one"

Alternative 3) Plot the data points and fit a straight line to the points. Project this line backwards to a date that would correspond to "day one".

Any of the three alternatives would be an approximation of the true "day one". If a software package is being tested thoroughly it is not unlikely that the first error will be found within the first day of testing. Therefore, "day one" was computed using Alternative 2 for those files requiring estimation of "day one". Declaring the date of the first SPR as "day one" as done in Alternative 1 would lose valuable data on the first SPR. Actual usage of Alternative 3 projected "day one" back to a date which was unacceptable. It was felt that the data from the first SPR was too important, especially for those models requiring the total number of errors per interval, to be lost by using Alternative 1 and Alternative 3 gave unreasonable predictions of "day one". Alternative 1 was tried for several files resulting in estimated parameters that were only slightly different by no more than a magnitude of one and distributions that were the same as for those using Alternative 2.

Chapter V

Parameter Estimation and Test of Hypothesis

V.1 Introduction

A means of parameter estimation and discrimination between the nine selected software reliability models had to be established. These functions were incorporated into a computer program called Program SRMOD (See Apppendix A).

V.2 Parameter Estimation

Each software reliability model has certain parameters that must be estimated (See Table II). The models in Group I of Table II required only the time-to-error (t_i) data from the data base, while those in Group II required time-to-error and the total number of errors per interval from the data base. Both of these inputs were available from the RADC files. From the extracted raw data, which consisted only of the time between successive SPRs and the number of errors within the interval, numerous calculations had to be made before parameter estimation could proceed. The cumulative number of errors up through each interval, the cumulative time up through each interval, the total number of time intervals, and the total number of errors had to be computed. Various other summations were

necessary before parameter estimation could be completed. These summations can be found in the equations that follow.

To estimate N and phi for the JM, SW, LJM, LSW, and MLSW models the following formulas (Ref [12]:2-3) were used:

$$\frac{K}{\hat{N}+1-(B/A)} = \sum_{i=1}^{j} \frac{M_i}{\hat{N}-n_{i-1}} \tag{9}$$

and

$$\hat{\phi} = \frac{(K/A)}{\hat{N}+1-(B/A)} \tag{10}$$

where

$$A = \begin{cases} \sum_{i=1}^{j} t_i & \text{for JM and LJM} \\ \sum_{i=1}^{j} (t_i^2/2) & \text{for SW and LSW} \\ \sum_{i=1}^{j} (t_i (T_{i-1} + t_i/2) & \text{for MLSW} \end{cases}$$

$$B = \begin{cases} \sum_{i=1}^{j} i t_i & \text{for JM} \\ \sum_{i=1}^{j} (n_{i-1} + 1) t_i & \text{for LJM} \\ \sum_{i=1}^{j} (i t_i^2/2) & \text{for SW} \\ \sum_{i=1}^{j} (n_{i-1} + 1)(t_i^2/2) & \text{for LSW} \\ \sum_{i=1}^{j} (n_{i-1} + 1) t_i (T_{i-1} + t_i/2) & \text{for MLSW} \end{cases}$$

t_i = length of time interval in which "M_i" errors were observed

T_{i-1} = cumulative time through (i-1)st interval

$$T_{i-1} = \sum_{r=1}^{i-1} t_r$$

$$T_0 = 0$$

n_{i-1} = cumulative number of errors observed up through the (i-1)st time interval

$$n_{i-1} = \sum_{r=1}^{i-1} M_r$$

$$n_0 = 0$$

M_i = number of errors in interval "i"

j = total number of time intervals

K = total number of errors observed

$$K = \sum_{i=1}^{j} M_i$$

The formulas are a consolidated version of several equations, each used to estimate N and phi for a particular model, utilizing the maximum likelihood function. They are basically the same as stated by Lipow (Ref [11]:2-3) with the exception that equations 9 and 10 also consider estimation of parameters for models JM, SW, and LSW. In all cases the maximum likelihood estimate for N and phi were used as explained in (Ref [8],[11],[14] ,[15],[23]).

Estimation of alpha and beta for WM1 and WM2 used formulas taken from Program Model (Ref [18] and [20]).

$$\hat{\alpha} = \frac{1}{\sqrt{\left[\left[\frac{S}{J-1}\right] - \left[\frac{R^2}{J(J-1)}\right]\right]}} * .6079271 \qquad (11)$$

$$\hat{\beta} = EXP(R/J + .5772/\hat{\alpha}) \qquad (12)$$

where

j = total number of time intervals

$$R = \sum_{i=1}^{j} \log(t_i)$$

$$S = \sum_{i=1}^{j} \log(t_i^2)$$

t_i = length of time interval

The two geometric models, JMG and MG, required "D" and "k" to be estimated. Given that the pdf for the JMG is

$$f(t_i) = Dk^{i-1} EXP(-Dk^{i-1} t_i) \qquad (13)$$

where

i = interval

t_i = length of time interval

D = initial detection rate

k = ratio constant

the maximum likelihood estimates for "D" and "k" can be determined (Ref [14]:330) and (Ref [11]:1). If given a sample of time-to-errors $(t_1, t_2, \ldots . t_j)$ the likelihood function, denoted by "L", would be

$$L(t_1, t_2, \ldots\ldots, t_j) = \prod_{i=1}^{j} Dk^{i-1} EXP(-Dk^{i-1} t_i) \qquad (14)$$

where "j" equals the number of intervals. Taking the natural logarithm of both sides of equation (14) produces

$$\log L = [j \log D] + [\sum_{i=1}^{j} (i-1)\log k] - [\sum_{i=1}^{j} Dk^{i-1} t_i] \qquad (15)$$

Maximizing equation (15) with respect to "D" yields

$$\frac{\delta \log L}{\delta D} = \frac{j}{D} - \sum_{i=1}^{j} k^{i-1} t_i = 0 \qquad (16)$$

and maximizing with respect to "k" yields

$$\frac{\delta \log L}{\delta k} = \left[\frac{1}{k}\right] \sum_{i=1}^{j} (i-1) - D \sum_{i=1}^{j} (i-1) k^{i-2} t_i = 0 \qquad (17)$$

The solution of equation (16) for "D" can then be substituted into equation (17) to arrive at

$$\frac{\sum_{i=1}^{j} i k^i t_i}{\sum_{i=1}^{j} k^i t_i} = \frac{j+1}{2} \qquad (18)$$

Since "j" and the t_i's are known quantities, equation (18) can be solved for "k". The solution will be called "$\hat{k}$" and will be a maximum likelihood estimate for "k". Likewise, the "$\hat{k}$" can be substituted into equation (16) to form

$$\hat{D} = \frac{j}{\sum_{i=1}^{j} \hat{k}^{i-1} t_i} \tag{19}$$

from which a maximum likelihood estimate for "D" can be derived.

Similiarly the maximum likelihood estimates of "k" and "D" for MG were derived to yield

$$\frac{\sum_{i=1}^{j} n_{i-1} \sum_{i=1}^{j} k^{n_{i-1}} t_i}{j} = \sum_{i=1}^{j} (n_{i-1}) k^{n_{i-1}} t_i \tag{20}$$

$$\hat{D} = \frac{j}{\sum_{i=1}^{j} \hat{k}^{n_{i-1}} t_i} \tag{21}$$

where

i = interval

j = total number of time intervals

n_{i-1} = cumulative number of errors up through the (i-1)st time interval

V.3 Model Restrictions

One problem, as recognized by Lapadula (Ref [9]) and Sukert (Ref [23]), did arise in all models requiring an estimation of "N". To solve equation (9) for "N" (estimate for the initial error content), incremental values were substituted for "N" until a value was found that would make the left side of equation (9) equal to its right side.

For several data sets Program SRMOD aborted because the computer time spent trying to find such an equality exceeded the allowable execution time permitted on the processing computer. In all cases the estimate for "N" at the time of abortion was far greater than would be expected. If it is assumed the models are valid, then the software projects from which the data came from would have had serious problems. No information could be found that would indicate this fact. No positive reason could be found that would explain why some data files had this problem and others not.

Sukert (Ref [23]:41) suggested that it could be caused by short time intervals consisting of many errors at the end of a file. This pattern was prevalent in those files which had the problem in this study.

The simplest and most convenient means found to alleviate this problem, so that testing could proceed without being aborted, was to allow "N" (the estimate for the initial error content of the software package) to reach

n (the actual number of errors found thus far in the software package) plus 10,000. At this point it was deemed highly unlikely that a reasonable solution for "N" could be determined. When this situation occurred, "N" was set to this arbitrarily large sum. After all restrictions were resolved and the parameters were estimated for the models, a test of hypothesis was selected to discriminate between the models.

V.4 Likelihood Ratio Test

The Likelihood Ratio Test (LRT) was chosen as the easiest means of discriminating between the nine different distributions. The LRT is defined by

$$L(T) = \frac{f_1(t|H_1)}{f_0(t|H_0)} \tag{22}$$

where

$$f_r(T|H_r) = \prod_{i=1}^{j} f_r(t_i|H_r)$$

$$H_0: G(t) = F_0(t|\theta_0)$$

$$H_1: G(t) = F_1(t|\theta_1)$$

The null hypothesis, H_0, is that the sample of t_i's taken from a population with an unknown cumulative distribution function, G(t), is equal to a theoretical distribution, $F_0(t|\theta_0)$, where θ_0 is the known parameter(s) of "F". The

null hypothesis can be tested against the alternative hypothesis, which states that G(t) equals another theoretical distribution, $F_1(t|\theta_1)$. Given that the t_i's are statiscally independent then the quantity $f_r(T|H_r)$ equals the joint pdf of the t_i's. The criterion against which L(T) is measured for acceptance or rejection of a given hypothesis is called the threshold of the test and is denoted by "η". The threshold is determined by the cost of making a correct or incorrect decision based upon apriori probabilities of the hypotheses in question. For this application it was assumed that the distributions under test had equal likelihood of occurrence and that the cost for an incorrect decision, regardless of the decision, was the same. Also, it was assumed there was no penalty for a correct decision. Given these assumptions "η" equals one and the LRT can be written

$$L(T) \underset{<H_0}{\overset{>H_1}{\gtrless}} \eta \quad \text{or further} \quad \ln[L(T)] \underset{<H_0}{\overset{>H_1}{\gtrless}} \ln(1.) \qquad (23)$$

which states that if the log of L(T) is less than the log of one, choose H_0, if it is greater than the log of one, choose H_1. The computational algorithm utilized in Program SRMOD was based on

$$\sum_{i=1}^{j}\{\ln[f_1(t_i|H_1)]\} - \sum_{i=1}^{j}\{\ln[f_0(t_i|H_0)]\} \underset{<H_0}{\overset{>H_1}{\gtrless}} 0 \qquad (24)$$

This approach can be repetitively applied to test several hypothesis against one another. Eventually, one hypothesis will be selected over all others. As an example, two models from Group II are tested against each other using EF1 as the sample population. The hypotheses are

H_0 : Distribution is the Modified Geometric Model

H_1 : Distribution is the Lipow-Schick Wolverton Model

Using the log of the maximum likelihood function as given in Appendix **C** for EF1, then

$$\sum_{i=1}^{j} \ln[f_0(t_i|H_0)] = -189.87$$

and

$$\sum_{i=1}^{j} \ln[f_1(t_i|H_1)] = -3937.13$$

which yields

$$(-3937.13) - (-189.87) \begin{matrix} >H_1 \\ \\ <H_0 \end{matrix} 0 \qquad (25)$$

or

$$-3747.26 \begin{matrix} >H_1 \\ \\ <H_0 \end{matrix} 0 \qquad (26)$$

Since -3747.26 is less than 0.0, H_0 is selected over H_1. This process is continued with the other models until one emerges over all others.

However, a decision had to be made on whether or not all of the models could be tested against one another. It was quite obvious that the JM model, which required only time-to-error as input, could not be tested against the LJM, which required time-to-error and the number of errors per interval as input. The only reasonable answer to be found was that hypothesis testing would be dependent on input. Thus, JM, SW, JMG,and WM1, which required only time-to-error as input, were tested using the Likelihood Ratio Test. In a separate and independent test the LJM, LSW, MLSW, MG and WM2 models were tested using the Likelihood Ratio Test. For ease of computation the natural logarithm was used in computing the maximum likelihood function for each distribution. On all Program SRMOD's outputs the natural log of the maximum likelihood functions (LLR) is given. In several cases the maximum likelihood function became so small it was considered equal to zero by the computer. The logarithm of zero cannot be taken and Program SRMOD aborted. To resolve this problem the log of the maximum likelihood function was set arbitrarily to -99999.0 whenever the computer considered the maximum likelihood function equal to zero. On all of Program SRMOD's outputs an LLR equal to -99999.0 should be treated as an extremely large negative number greater than the natural logarithm of (EXP(-675.))). A complete listing of Program SRMOD is given in Appendix A.

Chapter VI

Execution And Results

VI.1 Execution of Program SRMOD

Program SRMOD was designed and developed to estimate the parameters for each of the software reliability models in Table II and then discriminate between the models by hypothesis testing. For each of the time-to-error files listed in Table IV, parameters were estimated by Program SRMOD for all of the software reliability models being considered. Next Program SRMOD did a hypothesis test using the Likelihood Ratio Test as described in Section V.4. Upon completion, a listing was printed containing the vital statistics. A complete self-documented copy of Program SRMOD is contained in Appendix A.

Data input requirements for Program SRMOD were fairly simple. Each input record represented the time between two successive SPRs and the number of errors that occurred within this time frame. From this data all calculations for parameter estimation and hypothesis testing could be made for both Group I and Group II models. Calculations for Group I models ignored the data field which contained the number of errors per time interval. EF5C is listed in its entirety in Appendix B as a sample of the input required.

A seperate computer output listing was created for each group for a particular time-to-error file. In all there were 20 Group I output listings and 20 Group II output listings. Each listing was basically of the same format and contained information such as the total number of errors to date, total number of intervals, estimated parameters for each model, the log of the maximum likelihood function for each function, and finally the model selected by hypothesis testing. Appendix C contains all 40 output listings.

A summary of the results from Program SRMOD are presented in Tables VI and VII. The following section will discuss these results.

VI.2 Results from Program SRMOD

Table VI lists the time-to-error files, along with the software reliability models that were selected for each. Most obvious in Table VI is the number of files which had a geometric model selected for their time-to-error data. In Table VII it can be seen that the majority of files had a geometric model selected. EF1 was one of the largest time-to-error files and was extracted from File1. It contained data from 152 SPRs. The Schick-Wolverton Model was selected as the distribution which best modelled its time-to-error data points. However, when EF2 the operational data for the same software project, was joined with EF1, the Jelinski-Moranda Geometric Model was selected. The data for both

"modifications" (EF3A and EF3B) of File3 had the Jelinski-Moranda Geometric Model selected. EF4, which consisted of 149 data points from File4, had the Jelinski-Moranda De-Eutrophication Model selected for its distribution. File5 was divided into two groups of seven files. One group depicted data collected by the day, the other depicted data collected by the hour. Each of the seven files contained data from an independent functional group of software. The files for functional groups "A" and "D" (EF5A,EF5AH,EF5D,EF5DH) were the only files which had a different model selected for data collected by the day than for data collected by the hour. The Jelinski-Moranda Geometric model was selected the majority of the time for the subfiles of File5. In total three different models were chosen from Group I for the twenty files.

Group II was more consistent in its model selection. EF5GH was the only file that did not have the Modified Geometric Model selected for its time-to- error data. The Weibull2 model was selected for EF5GH. The data within EF5GH was data recorded by the hour for "functional group" "G" of File5. No obvious differences between this data and data in other files could be found.

Table VI

Model Selection By Group*

File	Distribution Selected From Group I	Distribution Selected From Group II
EF1	SW	MG
EF2	JMG	MG
EF1/2	JMG	MG
EF3A	JMG	MG
EF3B	JMG	MG
EF4	JM	MG
EF5A	JM	MG
EF5B	JMG	MG
EF5C	JMG	MG
EF5D	JM	MG
EF5E	JMG	MG
EF5F	JMG	MG
EF5G	JMG	MG
EF5AH	SW	MG
EF5BH	JMG	MG
EF5CH	JMG	MG
EF5DH	JMG	MG
EF5EH	JMG	MG
EF5FH	JMG	MG
EF5GH	JMG	WM2

* As specified in Table V:
Group I consisted of the JM, SW, WM1, and JMG models and required only the diference in time between "stop dates" as input.

Group II consisted of the LJM, LSW, MLSW, WM2, and MG models and required time-to-error and the number of errors within each interval

Table VII

Selection Summary

Group I

Model	Number of files selecting model
JM	3
SW	2
WM1	0
JMG	15

Group II

Model	Number of files selecting model
LJM	0
LSW	0
MLSW	0
WM2	1
MG	19

VI.3 Execution of Program Time-To-Fix

The data for Program Time-To-Fix was used in three different variations. First, each file in Table V was used as a continuous file to see if a distribution could be found for the random variable time-to-fix of an entire file. Next, an attempt to model time-to-fix within the major error categories (e.g. "AA", "BB", etc.) for a given file was performed. Finally, time-to-fix within subcategories (e.g. "AA010", "DD010", etc.) was modelled, trying to determine a distribution which best suited the data points in each subcategory of each file. There were 20 categories and over 140 subcategories whose time-to-fix data points were considered for modelling on each file in Table V. This means there were 160 output listings for each file or in total 960 listings from Program Time-To-Fix. This was too voluminous to include in this report. It was necessary to condense the output and include the results in table format, as can be seen in Tables VIII through XVII. Sample outputs from Program Time-To-Fix are given in Appendix D.

Each output listing consisted of four parts. First the estimates for the parameters were listed for the eleven distributions listed in Table III. Next the results from the Kolmogorov-Smirnoff test were printed. The Likelihood Ratio Test results followed and then finally the decision of which model, if any, was printed.

Input for Program Time-To-Fix consisted only of the

time-to-fix and the error category for that particular problem. As an example of the input required for Program Time-To-Fix a portion of FF1 is listed in Appendix E.

VI.4 Results from Program Time-To-Fix

Time-to-fix, without regard to an error category, could not be modelled at the 0.20 signifinance level for the given files. The Kolmogorov-Smirnoff test rejected all distributions when the files were used as a whole. No other signifinance levels were tried, primarily because of the time constraints of this study.

However, when the major error categories were extracted from the files and the time-to-fix data points for each category were modelled, distributions were selected. For example, there were 109 category "AA" errors in file FF5B and the Program Time-To-Fix selected the log-normal distribution as the distribution that best modelled time-to-fix for this error category. However, there were several categories for which time-to-fix could not be modelled. There were 192 category "AA" errors within file FF1, but no distribution was selected.

Table VIII reflects the percentage of categories which had distributions selected for their time-to-fix data. As can be seen, there was a wide difference between files. Categories "HH", "II", "KK", "PP", "RR", and "SS" had distributions selected for all six files. Time-to-fix for

some categories could not be modelled because there was either zero or only one data point. Time-to-fix for any particular category was always modelled in at least one file, but never was the distribution the same for all six files. The number of data points per category for each file varied from zero to 572. Table IX shows that the number of data points per category also varied greatly. A large percentage of categories had more than twenty data points, which means that the findings could be statistically significant. As would be expected, those categories with very few data points passed the Kolmogorov-Smirnoff Test more often than not. However, much more confidence could be placed in selected distributions which had a large number of data points and passed the Kolmogorov-Smirnoff test, than for those which only had a few data points. Table X shows that even when the categories had over twenty time-to-fix data points, time-to-fix could be modelled. Even the highest range of data points (50 or over) had distributions selected for time-to-fix for 45 percent of the categories.

A few examples of distributions selected are given in Table XI. Certain distributions appearred to be more prevalent. The log-normal distribution was chosen 42 percent of the time, as indicated in Table XII. The log-normal, the exponential, and the gamma distributions were usually selected on the basis of a large number of data points. On the other hand, the uniform and the pareto distributions were usually selected on the basis of less than ten data points.

As can be seen by the number of subcategories in Appendix F, categories such as "AA", "BB", "CC", and "DD" are quite inclusive in the type of errors that they cover. Trying to model time-to-fix within categories which are so broad in nature may not be the best approach. Possibly using subcategories, instead of major error categories, could produce more revealing information on the attributes of time-to-fix.

The problem with this approach is the loss of a large portion of the data, as is pointed out in Tables XIII and XIV. There was a significant loss of data points in all six files. Time-to-fix could not be modelled for fifty percent of the 864 possible subcategory files, because the subcategories had one or less time-to-fix data points and, therefore, distributions could not be selected. When there were enough data points for modelling, the percentage of subcategories having distributions selected decreased drastically as the number of points increased (See Table XV). Again, a variety of distributions was selected. There were a significant number of cases where distributions were selected for all six files for a particular subcategory. However, in no case was this distribution the same for all six files. A sample of the distributions selected for the subcategories is given in Table XVI. The entries in Table XVII were selected on the basis of their large number of data points. Again, the log-normal, the

exponential, and the gamma distributions were generally accepted on the basis of a large number of data points.

Table XVII points out that the uniform and pareto distributions were selected more than any of the other distributions. A closer look into this fact showed that the uniform was usually selected on the basis of fewer than ten data points. The pareto was selected approximately half the time on the basis of ten or fewer data points and the other half of the time on the basis of more than ten time-to-fix data points.

No significant difference was seen between those files measured in work days versus calendar days or calendar days versus hundredths of hours to fix. In a few cases the distributions differed, but generally the distributions stayed the same.

Table VIII

Modelling Time-To-Fix Within Categories

File	Percentage of categories in which a distribution was selected for time-to-fix
FF1	47
FF3	82
FF4A	53
FF4B	47
FF5	82
FF5H	88
Overall	67

Table IX

Data Point Spread Within Categories

Number of data points	Percentage of categories
0-1	6
2-9	20
10-19	27
20-49	21
50-over	26

Table X

Modelling of Time-To-Fix Within Categories by the Number of Data Points

Number of data points	Percentage of categories in which a distribution was selected for time-to-fix
0-1	0
2-9	95
10-19	79
20-49	95
50 or over	42

Table XI

A Sample of Distributions Selected For Time-To-Fix Within Categories

File	Category	Data points	Distribution selected
F1	JJ	152	Gamma
F3	BB	51	Lognormal
F4A	CC	18	Gamma
F4B	KK	31	Lognormal
F5	AA	109	Lognormal
F5	PP	148	Gamma
F5	RR	144	Exponential
F5H	MM	67	Lognormal

Table XII

Distribution Selection For Time-To-Fix Within Major Error Categories

Distribution	Percentage of time selected
Lognormal	42
Exponential	18
Uniform	15
Pareto	11
Gamma	9
Extreme-Value Largest	3
Extreme-Value Smallest	1
Weibull	1

Table XIII

Modelling Time-To-Fix Within Subcategories

File	Percentage of subcategories in which a distribution was selected for time-to-fix
FF1	67
FF3	30
FF4A	34
FF4B	33
FF5	50
FF5H	46
Overall	43

Table XIV

Data Point Spread Within Subcategories

Number of data points	Percentage of subcategories
0-1	50
2-9	29
10-19	10
20-49	7
50-over	4

Table XV

Modelling Time-To-Fix Within Subcategories by the Number of Data Points

Number of data points	Percentage of subcategories in which a distribution was selected for time-to-fix
0-1	0
2-99	96
10-19	89
20-49	79
50 or over	27

Table XVI

A Sample of Distributions Selected For Time-To-Fix Within Error Subcategories

File	Subcategory	Data points	Distribution selected
F1	BB100	54	Log-normal
F1	CC050	60	Log-normal
F1	GG020	36	Log-normal
F4A	BB020	46	Gamma
F4A	DD140	33	Gamma
F4B	DD010	27	Log-normal
F5	AA040	40	Log-normal
F5	BB060	173	Gamma
F5	DD050	67	Exponential
F5	PP020	116	Exponential
F5H	AA040	40	Exponential
F5H	BB020	46	Gamma
F5H	PP020	116	Log-normal

Table XVII

Distribution Selection For Time-To-Fix Within Error Subcategories

Distribution	Percentage of time selected
Uniform	36
Pareto	20
Log-normal	19
Exponential	13
Gamma	5
Extreme-Value-Largest	3
Extreme-Value Smallest	1
Laplace	1
Weibull	1

Chapter VII

Conclusions and Recommendations

VII.1 Conclusions

Before beginning this thesis it was not known if the existing software reliability models could be tested against one another. It was found that certain models, as given in Table II, could be tested and the criteria for which models could or could not be tested was primarily based on the available data and the basic underlying principles of each model. Utilizing maximum likelihood estimators for parameter estimations and the Likelihood Ratio Test for hypothesis testing, a computer program was wriiten which could discriminate between the software reliability models for a given data set. One of the most important findings was the dire need for correct and usable input data before testing can begin.

The Jelinski-Moranda Geometric Model was chosen a greater number of times than any other model in Group I of Table II. The Modified Geometric Model showed even more of a predominance over Group II models. Why did this occur? Both geometric models allow an infinite number of errors to occur over the life-cycle of a computer software system. Some of the other models assume a fixed number of errors. Allowing for an infinite number of errors is a more

realistic view of the data files under study. The data files came from projects which were large and had numerous changes and updates. New errors were introduced throughout the life-cycle and new requirements placed the projects in a dynamic environment. This was noticed when File2 was joined with File1. The addition of the operational data from File2 onto the testing data from File1 altered the selection of models from the Schick-Wolverton Model for File1 to the Jelinski-Moranda Geometric Model for File1/2. The detection rate of the geometric models varies geometrically and is not fixed as in some of the other models. This feature could account for the variation in the severity of testing at any particular time and also compensating for errors that do not necessarily have an equal probability of being detected. The size of the files had no impact on whether or not the geometric models were selected. This would enable them to be applied to any size software project.

Every data base should be be given a chance to be modelled and not to be prejudged regarding its distribution. Saying that all time-to-error data should be modelled by the geometric models could not be stochastically supported. A computer program, such as Program SRMOD, may be required to discriminate between models as a software project proceeds. The addition of EF2 onto EF1 to form EF1/2 is a good example of this. One model was chosen for the testing phase, while another model was chosen for the combined data from the testing and operational phases. It may be necessary to use

several different models during the life of the software package.

Modelling time-to-fix was not quite as conclusive as modelling time-to-error. It appeared that distributions for time-to-fix were project dependent. This could very well have been caused by the error categorization methodology. A certain error category could mean one thing to one contractor, but quite another to another contractor. Herein lies the problem of error categorization.

Disregarding error categories and trying to model software time-to-fix at large was unsuccessful at the 0.20 significance level. Modelling time-to-fix within major categories looked quite promising, in that time-to-fix for numerous categories had distributions selected. Trying to model time-to-fix within subcategories, did not improve the results. Rather, many data points were lost and could not be used in testing.

For many data sets the log-normal distribution was chosen. It is quite probable that with further modelling studies the log-normal distribution may play a key role in modelling time-to-fix for software projects. Other distributions that were selected quite frequently are the exponential, gamma, pareto, and uniform. Again, further testing and software error collection endeavors must be performed to validate or reject any of the findings of this study.

At best with the data that is presently available it

appears that modelling time-to-fix is software project dependent. No distribution emerged as the best for any particular file, category, or subcategory. Possibly with the use of the newer error categories better results would have been obtained.

VII.2 Recommendations

A disease can not be cured without the knowledge of the cause. So is the case with unreliable software. Herein lies the importance of collecting software error data. How can the dilemma of unreliable software ever be solved, if its characteristics are not identified and understood? More and better error collection tools and techniques must be developed and used. Errors should be collected on all types of software projects involving higher order languages, assembly languages, batch mode, time-sharing, structured design, and the countless other software procedures. Collection of software error data may seem costly, but considering that from software error data such items as personnel training requirements, pinpointing problem programs, time-to-fix analysis, and software reliability, to mention a few, may ultimately be determined by its collection and use.

The ultimate test for software reliability models is their actual use during software development and implementation. Only then can their validity be proven. If

software reliability models are used as a project is being developed, there is more control over what data should be collected and the data's correctness. Again, the importance of continued error collection is stressed. Further analysis is required to validate the findings of this study. More and better data must be collected, so that other software reliability models can be included in a program such as Program SRMOD. A software manager could then use this program, or one similar to it, to select a software reliability model for a software package. From the model selected the software manager could determine the reliability of the software and its mean-time-to-error. On the basis of these calculations, decisions, such as whether or not to accept the software, whether or not to implement the software, or whether or not the software package will be completed on schedule, could be made. Of the two groups of software reliability models listed in Table II, how does the software manager know which group of models to use? One suggestion is that if the manager is concerned only in the occurrence of the software downtime, regardless of how many errors occurred in the interval, the model selected from Group I should be used, otherwise the model from Group II should be used.

In this study entire projects or subfunctions were examined for their reliability. A possible different approach would be to determine the reliability of each computer program and utilizing the software system

flowcharts a topological network of programs could be formed. Each program could use a software reliability model for determining its reliability, but the entire network's reliability would be based upon the principles of reliability decomposition. Another recommendation is to investigate and include into Program SRMOD, if possible, the Geometric Poisson Model referenced in Section III.1.4.

Modelling time-to-fix appears to be dependent on the standarization of error categorization. The exact definition of error categories must be standardized before any meaningful results can be obtained. From the findings of this study it appears very unlikely that a distribution will ever be found which will universally model software time-to-fix. It does seem feasible to model time-to-fix data within error categories for a specified software project. As mentioned earlier, this study used a significance level of 0.20. An investigation at perhaps a significant level of 0.10 or 0.05 may be worthwhile. Another interesting area that should be examined is the correlation between the criticality of an error and the amount of time it took to fix the error. Both items are available in the RADC files.

If one recommendation had to be chosen over all others, it would be the importance of the continuation of software error collection. A list of items that would have aided this investigation and that should be considered in future software collection endeavors is as follows:

1) Instruction mix

2) Processing rate

3) CPU time for each test of a computer program

4) Number of machine instructions

5) Number of source instructions

6) Number of tests for each computer program

7) The number of successful and unsuccessful tests and the order in which they occurred

8) Apriori probability of correcting a specific error

9) Date error occurred

10) Number of errors occuring during each test

11) Date error was fixed

12) Controls allowed over program inputs

13) Was seeding and tagging performed? If so, how many indigenous and induced errors were found per test

14) Exact dates of when the development, design, coding, testing, integration, and operational phases began

15) Error categories with definitions that are unambiguous

16) Phase in which error occurred

17) Program in which error occurred

18) Criticality of error

19) The length of time in actual CPU seconds and the number of days it took to fix an error

20) Intensity of testing

21) Manpower allotted for the software project

22) Experience of manpower allotted

Of course, it is realized that many of these items

are quite subjective in nature, and the cost of collecting all of these characteristics could be prohibitive. The economics and feasibility must be weighed for any given software development project. Only with further error data can there be any hope of quantifying the problems associated with the development and operation of software.

Appendix A

Program SRMOD

Contained within this appendix is Program SRMOD, which was used for parameter estimation and testing of software reliability models. The program is self-documented and was written in FORTRAN for execution on a CDC6600. ANSI standards were adhered to as much as possible to aid in portability. The read and write statements may require modifications before the program can be used on another computer system. Also, subroutine ZBRENT from the IMSL subroutine library was used to compute the root of "k" for the key equation in subroutines JMG and MG. If the IMSL subroutine library is not available on the executing computer system a subroutine that can compute the root of a polynomial will have to be obtained and substituted in for ZBRENT. The memory requirement of 56K is for a large data base consisting of 2200 error records. If the data base to be tested is not that large, many of the matrices utilized in the program can be reduced, thereby reducing memory requirements. Central processing (CP) time varies considerably and is not dependent on the number of input records, but rather on the convergence rate of the key equations used in parameter estimations. For the data bases utilized in this study the CP time varied from less than 1 second to 50 seconds.

Given below is the jobstream required to execute Program SRMOD from source code on a CDC6600 computer. It is assumed that the software error input file is stored on disk as Tape1.

```
CAS,CM56000. T770293,CASTLE,BOX 4534
ATTACH,IMSL,ID=X654321
LIBARY,IMSL.
ATTACH,TAPE1.
FTN.
LGO.
7/8/9 CARD
SOURCE DECK OF PROGRAM SRMOD
7/8/9 CARD
6/7/8/9 CARD (END OF JOB)
```

```
************************************************************************000100
************************************************************************000110
****     PROGRAM SRMOD     ***************************************000120
****     AUTHOR: CAPT STEVE G. CASTLE     ******************************000130
****     DATE: 21 JAN 78     ***************************************000140
************************************************************************000150
************************************************************************000160
****                                                                  000170
****     THIS PROGRAM COMPUTES PARAMETERS FOR NINE   ******************000180
****     SOFTWARE RELIABILTY MODELS. THE MODELS ARE  ******************000190
****     DIVIDED INTO TWO GROUPS. EACH GROUP THEN    ******************000200
****     HAS ONE OF ITS MODELS SELECTED  BY THE      ******************000210
****     LIKELIHOOD RATIO TEST AS THE MODEL THAT     ******************000220
****     BEST DESCRIBES THE ERROR DATA.  FURTHER     ******************000230
****     DETAILS ON THE MODELS CAN BE OBTAINED       ******************000240
****     FROM THE THESIS, 'SOFTWARE RELIABILTY:      ******************000250
****     MODELLING TIME-TO-ERROR AND TIME-TO-FIX.'   ******************000260
****                                                                  000270
************************************************************************000280
****                                                                  000290
****     GROUP I MODELS                                               000300
****     REQUIRED ONLY THE TIME BETWEEN CONSECUTIVE                   000310
****     SOFTWARE PROBLEM REPORT DATES                                000320
****    1) JELINSKI-MORANDA DE-EUTROPHICATION                         000330
****    2) SCHICK-WOLVERTON                                           000340
****    3) JELINSKI-MORANDA GEOMETRIC DE-EUTROPHICATION               000350
****    4) WEIBUL1                                                    000360
****                                                                  000370
************************************************************************000380
****                                                                  000390
****     GROUP II MODELS                                              000400
****     REQUIRE TIME BETWEEN CONSECUTIVE SOFTWARE                    000410
****     PROBLEM REPORT DATES AND THE NUMBER OF                       000420
****     ERRORS OCCURRING DURING THE INTERVAL                         000430
****     1) LIPOW-JELINSKI-MORANDA DE-EUTROPHICATION                  000440
****     2) LIPOW-SCHICK-WOLVERTON                                    000450
```

```
****     3) MODIFIED-LIPOW-SCHICK-WOLVERTON                                 000460
****     4) MODIFIED GEOMETRIC MODEL                                        000470
****     5) WEIBULL2                                                        000480
****                                                                        000490
************************************************************************000500
************************************************************************000510
****                                                                        000520
****      INPUT REQUIREMENTS                                                000530
****     DATA IS FORMATTED AS FOLLOWS:                                      000540
****     RECORD POSITION  !!  FIELD                                         000550
****     ---------------------------------------------------------------000560
****         1-6          !!  TIME BETWEEN SUCCESSIVE ERROR OR            000570
****                      !!    SOFTWARE PROBLEM REPORT DATES              000580
****     ---------------------------------------------------------------000590
****         7-12         !!  NUMBER OF ERRORS OCCURRING IN              000600
****                      !!    INTERVAL                                   000610
****     ---------------------------------------------------------------000620
****                                                                        000630
****     EACH RECORD IS READ IN WITH A 2F6.2 FORMAT     ****************000640
****      SPECIFICATION FROM TAPE1                    ********************000650
************************************************************************000660
************************************************************************000670
****                                                                        000680
****      BASIC DESIGN     ************************************************000690
****      1) ALL INPUT RECORDS ARE READ IN AND COMPUTATIONS                 000700
****         ARE MADE THAT WILL BE REQUIRED FOR PARAMETER                   000710
****         ESTIMATIONS.                                                   000720
****      2) EACH MODEL HAS ITS PARAMETERS DETERMINED IN A                  000730
****         SEPERATE SUBROUTINE.                                           000740
****      3) BASIC HEADERS ARE PRINTED FOR GROUP I                          000750
****      4) HYPOTHESIS TEST IS PERFORMED FOR GROUP I                       000760
****      5) RESULTS FOR GROUP I ARE PRINTED                                000770
****      6) BASIC HEADERS ARE PRINTED FOR GROUP II                         000780
****      7) HYPOTHESIS TEST IS PERFORMED FOR GROUP II                      000790
****      8) RESULTS FOR GROUP II ARE PRINTED                               000800
************************************************************************000810
```

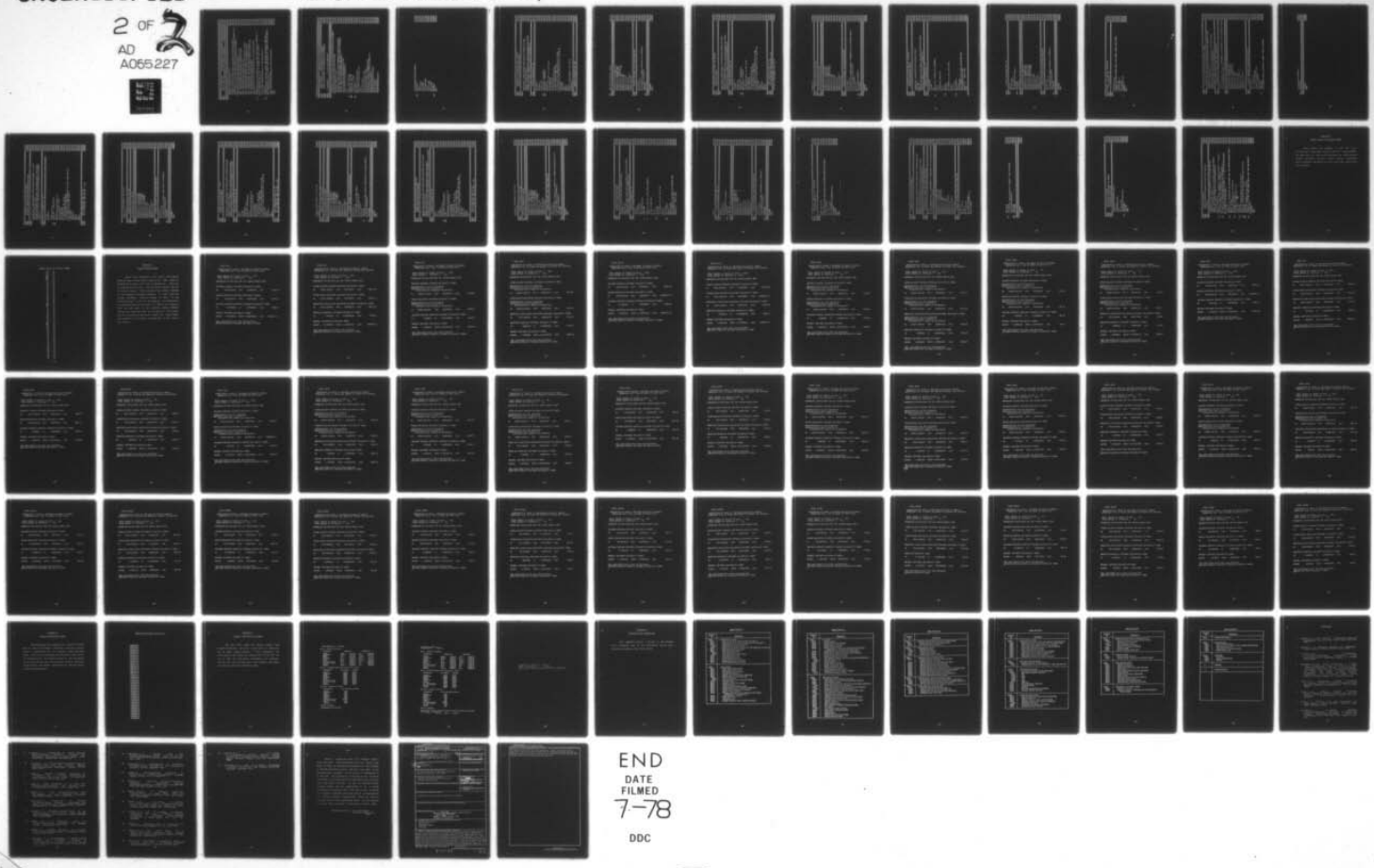
```
************************************************************************000820
****     SUBROUTINE AND VARIABLE DICTIONARY     ************************000840
****----------------------------------------------************************000850
****     BELOW IS A LIST AND DEFINITION OF VARIABLES                     000860
****     USED IN  THIS PROGRAM                                          000870
****-----------------------------------------------------------------    000880
****                                                                     000890
****     BJM - VALUE OF "B" FOR JMMOD                                    000900
****     BLJM - VALUE OF "B" FOR LJMMOD                                  000910
****     BLSW - VALUE OF "B" FOR LSWMOD                                  000920
****     BMSLW - VALUE OF "B" FOR MLSWM                                  000930
****     COEF - COEFFICIENTS TO KEY EQUATION FOR JMG                     000940
****     COEF1 - COEFFICIENTS TO KEY EQUATION FOR MG                     000950
****     CUT - CUMULATIVE TIME THROUGH (I-1) INTERVAL                    000960
****     DIFF - TNE MINUS ELN                                            000970
****     DLSN - DENOMINATOR OF LEFT SIDE OF KEY EQUATION                 000980
****     ELN - RSN TIMES DLSN                                            000990
****     FLAG - USED BY PRINT ROUTINE TO DETREMINE IF                    001000
****            "N" IS GREATER THAN 10,000 + TNI                         001010
****     FT - PROBABILITY DENSITY FUNCTION FOR MODEL                     001020
****     HEADER - HEADERS FOR EACH MODEL'S OUTPUT                        001030
****     HYPO - SUBROUTINE FOR HYPOTHESIS TESTING                        001040
****     JMG - SUBROUTINE FOR JELINSKI-MORANDA                           001050
****           GEOMETRIC MODEL                                           001060
****     JMMOD - SUBROUTINE FOR JELINSKI-MORANDA                         001070
****             DE-EUTROPHICATION MODEL                                 001080
****     J9 - A VALUE SIGNIFYING WHICH MODEL WAS CHOSEN                  001090
****     LJMMOD - SUBROUTINE FOR LIPOW-JELINSKI-                         001100
****              MORANDA MODEL                                          001110
****     LR - A MATRIX CONTAINING THE VALUE FOR EACH                     001120
****     LSWMOD - SUBROUTINE FOR LIPOW-SCHICK-WOLVERTON MODEL            001130
****     M - A MATRIX CONTAINING THE NUMBER OF ERRORS                    001140
****         PER INTERVAL                                                001150
****     MG - SUBROUTINE FOR MODIFIED GEOMETRIC MODEL                    001155
****     MLSW - SUBROUTINE FOR MODIFIED-LIPOW-SCHICK-                    001160
```

```
****          WOLVERTON MODEL                                          001170
****     N - ESTIMATE FOR INTITIAL ERROR CONTENT                       001180
****     NI1 - CUMULATIVE NUMBER OF ERRORS OBSERVED UP                 001190
****           THROUGH THE (I-1) TIME INTERVAL                         001200
****     NST - COUNTER                                                 001210
****     PARAM - CONTAINS VALUES FOR EACH ESTIMATED PARAMETER          001220
****     PHI - ESTIMATE FOR PHI                                        001230
****     PRINT - SUBROUTINE FOR PRINTING OUTPUT                        001240
****     RSN - RIGHT SIDE OF KEY EQUATION                              001250
****     SSTE - VALUE OF "A" FOR SWMOD                                 001290
****     STE - VALUE OF "A" FOR JMMOD AND LJMMOD                       001300
****     SUM3A - PRODUCT OF TIME-TO-ERRORS FOR WM1                     001310
****     SUM3B - PRODUCT OF THE SQUARES OF THE                         001320
****             TIME-TO-ERRRORS FOR WM1                               001330
****     SUM4A - PRODUCT OF TIME-TO-ERRORS                             001340
****     SUM4B - PRODUCT OF THE SQUARES OF THE                         001350
****             TIME-TO-ERRORS FOR WM2                                001360
****     SWITCH - USED TO PROCESS FIRST DATA POINT IN                  001370
****              KEY EQUATION                                         001380
****     SWMA - VALUE OF "A" FOR MLSWM                                 001390
****     SWMOD - SUBROUTINE FOR SCHICK-WOLVERTON MODEL                 001400
****     TAPE1 - INPUT DATA                                            001410
****     TE - TIME-TO-ERROR                                            001420
****     TEM - MATRIX CONTAINING ALL TIME-TO-ERROR INPUTS              001430
****     TEMP - TEMPORARY LOCATION FOR DIFF                            001440
****     TNE - TOTAL NUMBER OF ERRORS                                  001450
****     TNEI - TOTAL NUMBER OF EROS IN INTERVAL                       001460
****     TNI - TOTAL NUMBER OF INTERVALS                               001470
****     WEIBUL1 - SUBROUTINE FOR WEIBUL1(WM1)                         001480
****     WEIBUL2 - SUBROUTINE FOR WEIBUL2(WM2)                         001490
****                                                                   001500
****                                                                   001510
```

AD-A055 227 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
SOFTWARE RELIABILITY: MODELLING TIME-TO-ERROR AND TIME-TO-FIX.(U)
MAR 78 S G CASTLE

UNCLASSIFIED AFIT/GCS/EE/78-2 NL

2 OF
AD
A055227



END
DATE
FILMED
7-78
DDC

```
**********************************************************************001530
**********************************************************************001540
****     DRIVER ROUTINE     ******************************************001550
****     CALLS EACH SUBROUTINE AND PASSES VARIABLES     **************001560
**********************************************************************001570
**********************************************************************001580
      PROGRAM SRMOD(OUTPUT,TAPE1)                                       001590
      IMPLICIT REAL(A-I,K-Z)                                            001600
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)             001610
      DIMENSION FLAG(9),CUT(2200),PARAM(9,4),LR(9),M(2200),HEADER(9,6)  001620
      CALL READ(TE,TNEI,CUT,TNE,STE,SSTE,SWMA,BJM,BLJM,BSW,             001630
     1BLSW,BMLSW,M,SUM3A,SUM3B,SUM4A,SUM4B)                             001640
      CALL JMMOD(HEADER,TNE,BJM,STE,PARAM,LR,FLAG)                      001650
      CALL SWMOD(HEADER,TNE,BSW,SSTE,PARAM,LR,FLAG)                     001660
      CALL JMG(HEADER,TNE,PARAM,LR,FLAG)                                001670
      CALL WEIBUL1(HEADER,SUM3A,SUM4A,TNE,PARAM,LR,FLAG)                001680
      CALL LJMMOD(HEADER,TNE,BLJM,STE,M,PARAM,LR,FLAG)                  001690
      CALL LSWMOD(HEADER,TNE,BLSW,SSTE,M,PARAM,LR,FLAG)                 001700
      CALL MLSWM(HEADER,TNE,BMLSW,SWMA,M,PARAM,LR,CUT,FLAG)             001710
      CALL MG(HEADER,TNE,PARAM,LR,FLAG)                                 001720
      CALL WEIBUL2(HEADER,SUM3B,SUM4B,TNE,M,PARAM,LR,FLAG)              001730
      WRITE 10                                                          001740
10    FORMAT(*1*,5X,"EVALUATION OF SOFTWARE RELIABILITY MODELS",/,      001750
     $5X,"REQUIRING ONLY TIME TO ERROR AS INPUT DATA")                  001760
      CALL HYPO(LR,1,4,J9)                                              001770
      CALL PRINT(PARAM,HEADER,1,4,J9,FLAG,TNE,LR)                       001780
      WRITE 20                                                          001790
20    FORMAT(*1*,5X,"EVALUATION OF SOFTWARE RELIABILTY MODELS",/,       001800
     $2X,"REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL")   001810
      CALL HYPO(LR,5,9,J9)                                              001820
      CALL PRINT(PARAM,HEADER,5,9,J9,FLAG,TNE,LR)                       001830
      STOP                                                              001840
      END                                                               001850
```

```
**********************************************************************001860
**********************************************************************001870
****     READ ROUTINE     ************************************************001880
****     READS INPUT DATA AND COMPUTES PRELIMINARY                       001890
****     FORMULAS REQUIRED FOR PARAMETER ESTIMATION                      001900
****                                                                     001910
****     VARIABLES     I,J,J1 ARE COUNTERS                               001920
**********************************************************************001930
**********************************************************************001940
      SUBROUTINE READ(TE,TNEI,CUT,TNE,STE,SSTE,SWMA,BJM,BLJM,BSW,       001950
     1BLSW,BMLSW,M,SUM3A,SUM3B,SUM4A,SUM4B)                             001960
      IMPLICIT REAL(A-I,K-Z)                                            001970
      DIMENSION FLAG(9),CUT(2200),PARAM(9,4),LR(9),M(2200)              001980
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)             001990
      TNI=TE=TNEI=CT=TNE=STE=SSTE=SWMA=I=0.                             002000
      BJM=BLJM=BSW=BLSW=BMLSW=NI1(1)=0.                                 002010
      SUM3A=SUM3B=SUM4A=SUM4B=0.0                                       002020
      J=1                                                               002030
      NI1(1)=CUT(1)=0.0                                                 002040
10    READ(1,20) TE,TNEI                                                002050
20    FORMAT(F6.2,F6.2)                                                 002060
      IF(EOF(1)) 50,30                                                  002070
30    STE=STE+TE                                                        002080
      SSTE=SSTE+(TE**2.)*.5                                             002090
      SWMA=SWMA+(TE*(CT+TE*.5))                                         002100
      I=I+1.                                                            002110
      BJM=BJM+(I*TE)                                                    002120
      BLJM=BLJM+(TNE+1.)*TE                                             002130
      BSW=BSW+(I*(TE**2.))*.5                                           002140
      BLSW=BLSW+((TNE+1.)*(TE**2.)*.5)                                  002150
      BMLSW=BMLSW+((TNE+1.)*TE*(CT+(TE*.5)))                            002160
      SUM3A=SUM3A+ALOG(TE)                                              002170
      SUM4A=SUM4A+(ALOG(TE)**2.)                                        002180
      J1=IFIX(TNEI)                                                     002190
      DO 40 J2=1,J1                                                     002200
      SUM3B=SUM3B+ALOG(TE)                                              002210
```

```
      SUM4B=SUM4B+(ALOG(TE)**2.)                                    002220
40    CONTINUE                                                      002230
      TNI=J                                                         002240
      TEM(J)=TE                                                     002250
      M(J)=TNEI                                                     002260
      CT=CT+TE                                                      002270
      TNE=TNE+TNEI                                                  002280
      J=J+1                                                         002290
      NI1(J)=TNE                                                    002300
      CUT(J)=CT                                                     002310
      GO TO 10                                                      002320
50    RETURN                                                        002330
      END                                                           002340
```

```
C*****************************************************************002350
C****     PARAMETER ESTIMATION FOR JELINSKI-                          002360
C****     MORANDA DE-EUTROPHICATION MODEL                             002370
C*****************************************************************002380
C****                                                                 002390
      SUBROUTINE JMMOD(HEADER,TNE,BJM,STE,PARAM,LR,FLAG)             002400
      IMPLICIT REAL(A-I,K-Z)                                         002410
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)          002420
      DIMENSION FLAG(9),PARAM(9,4),LR(9),HEADER(9,6)                 002430
C*****************************************************************002440
C****     COMPUTE ESTIMATE OF "N"                                     002450
C*****************************************************************002460
      RSN=NST=0.                                                     002470
      J1=TNI                                                         002480
      SWITCH=0.                                                      002490
      N=TNI-.999                                                     002500
10    DLSN=(N+1.-(BJM/STE))                                          002510
20    RSN=RSN+(1./(N-NST))                                           002520
      NST=NST+1.                                                     002530
      IF(NST.LE.(TNI-1.)) GO TO 20                                   002540
      ELN=RSN*DLSN                                                   002550
      DIFF=TNI-ELN                                                   002560
      IF(DIFF.EQ.0.)GO TO 50                                         002570
      IF(SWITCH.EQ.0.)TEMP=DIFF                                      002580
      IF(((DIFF.LT.0.).AND.(TEMP.GT.0.)).OR.                         002590
     1((DIFF.GT.0.).AND.(TEMP.LT.0.))) GO TO 50                      002600
      TEMP=DIFF                                                      002610
      SWITCH=1.                                                      002620
      N=N+1.                                                         002630
      RSN=0.                                                         002640
      NST=0.                                                         002650
      IF(N.LT.(TNI+10000.)) GO TO 10                                 002660
      FLAG(1)=1.                                                     002670
C*****************************************************************002680
C****     COMPUTE ESTIMATE FOR PHI                                    002690
C*****************************************************************002700
```

```
50    PHI=TNI/(STE*(N+1.)-BJM)                                                  002710
**********************************************************************************002720
**********************************************************************************002730
      HEADER(1,1)="JELINSKI-M"                                                  002740
      HEADER(1,2)="ORANDA SOF"                                                  002750
      HEADER(1,3)="TWARE RELI"                                                  002760
      HEADER(1,4)="ABILITY MO"                                                  002770
      HEADER(1,5)="DEL       "                                                  002780
      HEADER(1,6)=10H                                                           002790
      PARAM(1,1)="N="                                                           002800
      PARAM(1,2)=N                                                              002810
      PARAM(1,3)="PHI="                                                         002820
      PARAM(1,4)=PHI                                                            002830
**********************************************************************************002840
****      COMPUTE MAXIMUM LIKELIHOOD FUNCTION                                   002850
**********************************************************************************002860
      LR(1)=0.0                                                                 002870
      DO 75 J=1,J1                                                              002880
      FT=PHI*(N-(J-1))*EXP(-PHI*(N-(J-1))*TEM(J))                               002890
      LR(1)=LR(1)+ALOG(FT)                                                      002900
75    CONTINUE                                                                  002910
**********************************************************************************002920
**********************************************************************************002930
      RETURN                                                                    002940
      END                                                                       002950
```

```
*****************************************************************************002960
****     PARAMETER ESTIMATION FOR SCHICK-                                    002970
****     WOLVERTON MODEL                                                     002980
*****************************************************************************002990
****                                                                         003000
      SUBROUTINE SWMOD(HEADER,TNE,BSW,SSTE,PARAM,LR,FLAG)                   003010
      IMPLICIT REAL(A-I,K-Z)                                                003020
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)                 003030
      DIMENSION FLAG(9),PARAM(9,4),LR(9),HEADER(9,6)                        003040
*****************************************************************************003050
****     COMPUTE ESTIMATE FOR "N"                                            003060
*****************************************************************************003070
      RSN=NST=0.                                                            003080
      J1=TNI                                                                003090
      SWITCH=0.                                                             003100
      N=TNI-.999                                                            003110
10    DLSN=(N+1.-(BSW/SSTE))                                                003120
20    RSN=RSN+(1./(N-NST))                                                  003130
      NST=NST+1.                                                            003140
      IF(NST.LE.(TNI-1.)) GO TO 20                                          003150
      ELN=RSN*DLSN                                                          003160
      DIFF=TNI-ELN                                                          003170
      IF(DIFF.EQ.0.)GO TO 50                                                003180
      IF(SWITCH.EQ.0.)TEMP=DIFF                                             003190
      IF(((DIFF.LT.0.).AND.(TEMP.GT.0.)).OR.                                003200
     1((DIFF.GT.0.).AND.(TEMP.LT.0.))) GO TO 50                             003210
      TEMP=DIFF                                                             003220
      N=N+1.                                                                003230
      SWITCH=1.                                                             003240
      RSN=0.                                                                003250
      NST=0.                                                                003260
*****************************************************************************003270
****     STOP COMPUTING FOR "N" IF                                           003280
****     "N" IS GREATER THAN 10,000 + TNI                                    003290
*****************************************************************************003300
      IF(N.LT.(TNI+10000.))GO TO 10                                         003310
```

```
      FLAG(2)=1.                                                          003320
*****************************************************************************003330
****      COMPUTE ESTIMATE FOR PHI                                         003340
*****************************************************************************003350
50    PHI=TNI/(SSTE*(N+1)-BSW)                                            003360
*****************************************************************************003370
*****************************************************************************003380
      HEADER(2,1)="SCHICK-WOL"                                            003390
      HEADER(2,2)="VERTON SOF"                                            003400
      HEADER(2,3)="TWARE RELI"                                            003410
      HEADER(2,4)="ABILITY MO"                                            003420
      HEADER(2,5)="DEL       "                                            003430
      HEADER(2,6)=10H                                                     003440
      PARAM(2,1)="N="                                                     003450
      PARAM(2,2)=N                                                        003460
      PARAM(2,3)="PHI="                                                   003470
      PARAM(2,4)=PHI                                                      003480
*****************************************************************************003490
****      COMPUTE MAXIMUM LIKELIHOOD FUNCTION                              003500
*****************************************************************************003510
      LR(2)=0.0                                                           003520
      DO 75 J=1,J1                                                        003530
      FT=PHI*(N-(J-1))*TEM(J)*EXP(-.5*PHI*(N-(J-1))*(TEM(J)**2.))         003540
      IF(FT.GT.1.0E-20)GO TO 70                                           003550
      LR(2)=-999999.                                                      003560
      GO TO 75                                                            003570
70    LR(2)=LR(2)+ALOG(FT)                                                003580
75    CONTINUE                                                            003590
*****************************************************************************003600
*****************************************************************************003610
      RETURN                                                              003620
      END                                                                 003630
```

```
**********************************************************************003640
****     PARAMETER ESTIMATION FOR JELINSKI-                            003650
****     MORANDA GEOMETRIC MODEL                                       003660
**********************************************************************003670
****                                                                   003680
      SUBROUTINE JMG(HEADER,TNE,PARAM,LR,FLAG)                        003690
      EXTERNAL F                                                      003700
      REAL LR                                                         003710
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)           003720
      DIMENSION PARAM(9,4),LR(9),HEADER(9,6)                          003730
**********************************************************************003740
****     COMPUTE ESTIMATE OF "K"                                       003750
**********************************************************************003760
      I=IFIX(TNI)                                                     003770
      J1=TNI                                                          003780
      J=1                                                             003790
5     COEF(J)=(I-((TNI+1)/2.))*TEM(I)                                 003800
      I=I-1                                                           003810
      J=J+1                                                           003820
      IF(I.NE.0) GO TO 5                                              003830
      A=.9                                                            003840
      B=1.1                                                           003850
10    FA=F(A)                                                         003860
      FB=F(B)                                                         003870
      IF((FA*FB).L1.0.0) GO TO 50                                     003880
      A=A-.1                                                          003890
      B=B+.1                                                          003900
      GO TO 10                                                        003910
50    EPS=.000001                                                     003920
      NSIG=6                                                          003930
      MAXFN=10000                                                     003940
      CALL ZBRENT(F,EPS,NSIG,A,B,MAXFN,IER)                           003950
      IF(IER.EQ.0)GO TO 100                                           003960
      PRINT *,'   IER=',IER                                           003970
      STOP                                                            003980
**********************************************************************003990
```

```
****     COMPUTE ESTIMATE FOR "D"                                          004000
**************************************************************************004010
100  DD=0.0                                                               004020
     N=IFIX(TNI)                                                          004030
     DO 200 I=1,N                                                         004040
     DD=DD+B**(I-1)*TEM(I)                                                004050
200  CONTINUE                                                             004060
     D=TNI/DD                                                             004070
**************************************************************************004080
**************************************************************************004090
     HEADER(3,1)="JELINSKI-M"                                             004100
     HEADER(3,2)="ORANDA GEO"                                             004110
     HEADER(3,3)="METRIC SOF"                                             004120
     HEADER(3,4)="WARE RELIA"                                             004130
     HEADER(3,5)="BILITY MOD"                                             004140
     HEADER(3,6)="EL        "                                             004150
     PARAM(3,1)="D="                                                      004160
     PARAM(3,2)=D                                                         004170
     PARAM(3,3)="K="                                                      004180
     PARAM(3,4)=B                                                         004190
**************************************************************************004200
****     COMPUTE MAXIMUM LIKELIHOOD FUNCTION                               004210
**************************************************************************004220
     LR(3)=0.0                                                            004230
     DO 75  J=1,J1                                                        004240
     FT=D*(B**(J-1))*EXP(-D*(B**(J-1))*TEM(J))                            004250
     LR(3)=LR(3)+ALOG(FT)                                                 004260
75   CONTINUE                                                             004270
**************************************************************************004280
**************************************************************************004290
     RETURN                                                               004300
     END                                                                  004310
```

```
**********************************************************************004320
**** COMPUTE  f(t) FOR KEY EQUATION                                    004330
****      TO ESTIMATE "K"                                              004340
**********************************************************************004350
****                                                                   004360
      FUNCTION F(X)                                                    004370
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)            004380
      F=COEF(1)                                                        004390
      N=IFIX(TNI)-1                                                    004400
      DO 1 I=1,N                                                       004410
1     F=X*F+COEF(I+1)                                                  004420
      RETURN                                                           004430
      END                                                              004440
```

```
*****************************************************************************004450
****     PARAMETER ESTIMATION FOR WEIBUL1 MODEL                                004460
*****************************************************************************004470
      SUBROUTINE WEIBUL1(HEADER,SUM3A,SUM4A,TNE,PARAM,LR,FLAG)                 004480
      IMPLICIT REAL(A-I,K-Z)                                                   004490
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)                    004500
      DIMENSION PARAM(9,4),LR(9),HEADER(9,6)                                   004510
*****************************************************************************004520
****     ESTIMATION OF SHAPE AND SCALE PARAMETERS                              004530
*****************************************************************************004540
      J1=TNI                                                                   004550
      A=1./(((SUM4A/(TNI-1.)-(SUM3A**2.)/(TNI*(TNI-1)))*(.6079271))**.5)       004560
      B=EXP(SUM3A/TNI+.5772/A)                                                 004570
*****************************************************************************004580
*****************************************************************************004590
      HEADER(4,1)="WEIBULL SO"                                                 004600
      HEADER(4,2)="FTWARE REL"                                                 004610
      HEADER(4,3)="IABILITY M"                                                 004620
      HEADER(4,4)="ODEL      "                                                 004630
      HEADER(4,5)=10H                                                          004640
      HEADER(4,6)=10H                                                          004650
      PARAM(4,1)="ALPHA="                                                      004660
      PARAM(4,2)=A                                                             004670
      PARAM(4,3)="BETA="                                                       004680
      PARAM(4,4)=B                                                             004690
*****************************************************************************004700
****     COMPUTE MAXIMUM LIKELIHOOD FUNCTION                                   004710
*****************************************************************************004720
      LR(4)=0.0                                                                004730
      DO  75 J=1,J1                                                            004740
      X=-((TEM(J)/B)**A)                                                       004750
      IF(X.GT.-675)GO TO 50                                                    004760
      LR(4)=-999999.                                                           004770
      GO TO 75                                                                 004780
50    CONTINUE                                                                 004790
      FT=(A/B)*((TEM(J)/B)**(A-1))*EXP(-(TEM(J)/B)**A)                         004800
```

```
      LR(4)=LR(4)+ALOG(FT)                                                    004810
75    CONTINUE                                                                004820
*************************************************************************004830
*************************************************************************004840
      RETURN                                                                  004850
      END                                                                     004860
```

```
**********************************************************************004870
****     PARAMETER ESTIMATION FOR LIPOW-                                004880
****     JELINSKI-MORANDA DE-EUTROPHICATION MODEL                       004890
**********************************************************************004900
****                                                                    004910
      SUBROUTINE LJMMOD(HEADER,TNE,BLJM,STE,M,PARAM,LR,FLAG)            004920
      IMPLICIT REAL(A-I,K-Z)                                            004930
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)             004940
      DIMENSION FLAG(9),PARAM(9,4),LR(9),HEADER(9,6),M(2200)            004950
      RSN=0.                                                            004960
**********************************************************************004970
**** COMPUTE ESTIMATE OF "N"                                           004980
**********************************************************************004990
      J1=TNI                                                            005000
      SWITCH=0.                                                         005010
      J=0                                                               005020
      N=TNE-M(J1)+.1                                                    005030
10    DLSN=(N+1.-(BLJM/STE))                                            005040
20    J=J+1                                                             005050
      RSN=RSN+(M(J)/(N-NI1(J)))                                         005060
      IF(J.LT.TNI)GO TO 20                                              005070
      ELN=RSN*DLSN                                                      005080
      DIFF=TNE-ELN                                                      005090
      IF(DIFF.EQ.0.)GO TO 50                                            005100
      IF(SWITCH.EQ.0.)TEMP=DIFF                                         005110
      IF(((DIFF.LT.0.).AND.(TEMP.GT.0)).OR.((DIFF.GT.0.)                005120
     1.AND.(TEMP.LT.0)))GO TO 50                                        005130
      TEMP=DIFF                                                         005140
      SWITCH=1.                                                         005150
      N=N+1.                                                            005160
      RSN=0.                                                            005170
      J=0                                                               005180
      NST=0.                                                            005190
**********************************************************************005200
****     STOP COMPUTING FOR "N" IF                                      005210
****     "N" IS GREATER THAN 10,000 + TNI                               005220
```

```
**********************************************************************005230
      IF(N.LT.(TNE+10000)) GO TO 10                                  005240
      FLAG(5)=1.                                                     005250
**********************************************************************005260
****      COMPUTE ESTIMATE FOR PHI                                    005270
**********************************************************************005280
50    PHI=TNE/(STE*(N+1.)-BLJM)                                      005290
**********************************************************************005300
**********************************************************************005310
      HEADER(5,1)="LIPOW-JELI"                                       005320
      HEADER(5,2)="NSKI-MORAN"                                       005330
      HEADER(5,3)="DA SOFTWAR"                                       005340
      HEADER(5,4)="E RELIABIL"                                       005350
      HEADER(5,5)="ITY MODEL "                                       005360
      HEADER(5,6)=10H                                                005370
      PARAM(5,1)="N="                                                005380
      PARAM(5,2)=N                                                   005390
      PARAM(5,3)="PHI ="                                             005400
      PARAM(5,4)=PHI                                                 005410
**********************************************************************005420
****      COMPUTE MAXIMUM LIKELIHOOD FUNCTION                        005430
**********************************************************************005440
      LR(5)=0.0                                                      005450
      DO 75 J=1,J1                                                   005460
      FT=PHI*(N-NI1(J))*EXP(-PHI*(N-NI1(J))*TEM(J))                  005470
      LR(5)=LR(5)+ALOG(FT)                                           005480
75    CONTINUE                                                       005490
**********************************************************************005500
**********************************************************************005510
      RETURN                                                         005520
      END                                                            005530
```

```
************************************************************************005540
****     PARAMETER ESTIMATION OF LIPOW-                                 005550
****     SCHICK-WOLVERTON MODEL                                         005560
************************************************************************005570
****                                                                    005580
      SUBROUTINE LSWMOD(HEADER,TNE,BLSW,SSTE,M,PARAM,LR,FLAG)           005590
      IMPLICIT REAL(A-I,K-Z)                                            005600
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)             005610
      DIMENSION FLAG(9),PARAM(9,4),LR(9),HEADER(9,6),M(2200)            005620
************************************************************************005630
****     COMPUTE ESTIMATE OF "N"                                        005640
************************************************************************005650
      RSN=0.                                                            005660
      J=0                                                               005670
      J1=TNI                                                            005680
      SWITCH=0.                                                         005690
      N=TNE-M(J1)+.1                                                    005700
10    DLSN=(N+1-(BLSW/SSTE))                                            005710
20    J=J+1                                                             005720
      RSN=RSN+(M(J)/(N-NI1(J)))                                         005730
      IF(J.LT.TNI) GO TO 20                                             005740
      ELN=RSN*DLSN                                                      005750
      DIFF=TNE-ELN                                                      005760
      IF(DIFF.EQ.0.) GO TO 50                                           005770
      IF(SWITCH.EQ.0.)TEMP=DIFF                                         005780
      IF(((DIFF.LT.0.).AND.(TEMP.GT.0.)).OR.                            005790
     1((DIFF.GT.0.).AND.(TEMP.LT.0.))) GO TO 50                         005800
      TEMP=DIFF                                                         005810
      N=N+1.                                                            005820
      SWITCH=1.                                                         005830
      J=0                                                               005840
      RSN=0.                                                            005850
************************************************************************005860
****     STOP COMPUTING FOR "N" IF                                      005870
****     "N" IS GREATER THAN 10,000 + TNI                               005880
************************************************************************005890
```

```
      IF(N.LT.(TNE+10000.)) GO TO 10                                    005900
      FLAG(6)=1.                                                        005910
*****************************************************************************005920
****      COMPUTE ESTIMATE FOR PHI                                       005930
*****************************************************************************005940
50    PHI=TNE/(SSTE*(N+1.)-BLSW)                                        005950
*****************************************************************************005960
*****************************************************************************005970
      HEADER(6,1)="LIPOW-SCHI"                                          005980
      HEADER(6,2)="CK-WOLVERT"                                          005990
      HEADER(6,3)="ON SOFTWAR"                                          006000
      HEADER(6,4)="E RELIABIL"                                          006010
      HEADER(6,5)="ITY MODEL "                                          006020
      HEADER(6,6)=10H                                                   006030
      PARAM(6,1)="N="                                                   006040
      PARAM(6,2)=N                                                      006050
      PARAM(6,3)="PHI="                                                 006060
      PARAM(6,4)=PHI                                                    006070
*****************************************************************************006080
****      COMPUTE MAXIMUM LIKELIHOOD FUNCTIONN                           006090
*****************************************************************************006100
      LR(6)=0.0                                                         006110
      DO 75 J=1,J1                                                      006120
      X=-PHI*(N-NI1(J))*(TEM(J)**2.)                                    006130
      IF(X.GT.-675.)GO TO 60                                            006140
      LR(6)=-999999.                                                    006150
      GO TO 75                                                          006160
60    CONTINUE                                                          006170
      FT=PHI*(N-NI1(J))*TEM(J)*EXP(-PHI*(N-NI1(J))*(TEM(J)**2.))        006180
      LR(6)=LR(6)+ALOG(FT)                                              006190
75    CONTINUE                                                          006200
*****************************************************************************006210
*****************************************************************************006220
      RETURN                                                            006230
      END                                                               006240
      ----
```

```
**********************************************************************006250
****     PARAMETER ESTIMATION FOR MODIFIED-                            006260
****     LIPOW-SCHICK-WOLVERTON MODEL                                  006270
**********************************************************************006280
****                                                                   006290
      SUBROUTINE MLSWM(HEADER,TNE,BMLSW,SWMA,M,PARAM,LR,CUT,FLAG)      006300
      IMPLICIT REAL(A-I,K-Z)                                           006310
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)            006320
      DIMENSION FLAG(9),CUT(2200),PARAM(9,4),LR(9),HEADER(9,6),M(2200) 006330
**********************************************************************006340
****     COMPUTE ESTIMATE OF "N"                                       006350
**********************************************************************006360
      RSN=0.                                                           006370
      J=0                                                              006380
      J1=TNI                                                           006390
      SWITCH=0.                                                        006400
      N=TNE-M(J1)+.1                                                   006410
10    DLSN=(N+1-(BMLSW/SWMA))                                          006420
20    J=J+1                                                            006430
      RSN=RSN+(M(J)/(N-NI1(J)))                                        006440
      IF(J.LT.TNI) GO TO 20                                            006450
      ELN=RSN*DLSN                                                     006460
      DIFF=TNE-ELN                                                     006470
      IF(DIFF.EQ.0.)GO TO 50                                           006480
      IF(SWITCH.EQ.0.)TEMP=DIFF                                        006490
      IF(((DIFF.LT.0.).AND.(TEMP.GT.0.)).OR.                           006500
     1((DIFF.GT.0.).AND.(TEMP.LT.0.)))GO TO 50                         006510
      TEMP=DIFF                                                        006520
      SWITCH=1.                                                        006530
      N=N+1.                                                           006540
      RSN=0.                                                           006550
      J=0                                                              006560
**********************************************************************006570
****     STOP COMPUTING FOR "N" IF                                     006580
****     "N" IS GREATER THAN 10,000 + TNI                              006590
**********************************************************************006600
```

```
      IF(N.LT.(TNE+10000.))GO TO 10                                          006610
      FLAG(7)=1.                                                             006620
*************************************************************************006630
****      COMPUTE ESTIMATE FOR PHI                                          006640
*************************************************************************006650
50    PHI=TNE/(SWMA*(N+1)-BMLSW)                                             006660
*************************************************************************006670
*************************************************************************006680
      HEADER(7,1)="MODIFIED-L"                                               006690
      HEADER(7,2)="IPOW-SCHIC"                                               006700
      HEADER(7,3)="K-WOLVERTO"                                               006710
      HEADER(7,4)="N SOFTWARE"                                               006720
      HEADER(7,5)=" RELIABILI"                                               006730
      HEADER(7,6)="TY MODEL  "                                               006740
      PARAM(7,1)="N="                                                        006750
      PARAM(7,2)=N                                                           006760
      PARAM(7,3)="PHI="                                                      006770
      PARAM(7,4)=PHI                                                         006780
*************************************************************************006790
****      COMPUTE MAXIMUM LIKELIHOOD FUNCTION                               006800
*************************************************************************006810
      LR(7)=0.0                                                              006820
      DO 75 J=1,J1                                                           006830
      FT=PHI*(N-NI1(J))*(CUT(J)+(TEM(J)/2.))*EXP(-PHI*(N-NI1(J))*            006840
     $ (CUT(J)+(TEM(J)/2.))*TEM(J))                                          006850
      LR(7)=LR(7)+ALOG(FT)                                                   006860
75    CONTINUE                                                               006870
*************************************************************************006880
*************************************************************************006890
      RETURN                                                                 006900
      END                                                                    006910
```

```
**********************************************************************006920
****     PARAMETER ESTIMATION FOR                                      006930
****     MODIFIED GEOMETRIC MODEL                                      006940
**********************************************************************006950
      SUBROUTINE MG(HEADER,TNE,PARAM,LR,FLAG)                          006960
      EXTERNAL F1                                                      006970
      REAL NI1,LR                                                      006980
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)            006990
      DIMENSION PARAM(9,4),LR(9),HEADER(9,6)                           007000
**********************************************************************007010
****     COMPUTE ESTIMATE OF "K"                                       007020
**********************************************************************007030
      J1=TNI                                                           007040
      TNI1=0.0                                                         007050
      I=IFIX(TNI)                                                      007060
      DO 1 N=1,I                                                       007070
      TNI1=TNI1+NI1(N)                                                 007080
1     CONTINUE                                                         007090
      J=1                                                              007100
5     COEF1(J)=(NI1(J)-(TNI1/TNI))*TEM(I)                              007110
      J=J+1                                                            007120
      IF(J.LE.I)GO TO 5                                                007130
      A=.9                                                             007140
      B=1.1                                                            007150
10    FA=F1(A)                                                         007160
      FB=F1(B)                                                         007170
      IF((FA*FB).LT.0.0) GO TO 50                                      007180
      A=A-.1                                                           007190
      B=B+.1                                                           007200
      GO TO 10                                                         007210
50    EPS=.000001                                                      007220
      NSIG=6                                                           007230
      MAXFN=10000                                                      007240
      CALL ZBRENT(F1,EPS,NSIG,A,B,MAXFN,IER)                           007250
      IF(IER.EQ.0)GO TO 100                                            007260
      PRINT *,'   IER=',IER                                            007270
```

```
      STOP                                                                  007280
*************************************************************************007290
****      COMPUTE ESTIMATE FOR "D"                                         007300
*************************************************************************007310
100   DD=0.0                                                               007320
      N=IFIX(TNI)                                                          007330
      DO 200 I=1,N                                                         007340
      DD=DD+B**(NI1(I))*TEM(I)                                             007350
200   CONTINUE                                                             007360
      D=TNI/DD                                                             007370
*************************************************************************007380
*************************************************************************007390
      HEADER(8,1)="MODIFIED G"                                             007400
      HEADER(8,2)="EOMETRIC M"                                             007410
      HEADER(8,3)="ODEL      "                                             007420
      HEADER(8,4)="          "                                             007430
      HEADER(8,5)="          "                                             007440
      HEADER(8,6)="          "                                             007450
      PARAM(8,1)="D="                                                      007460
      PARAM(8,2)=D                                                         007470
      PARAM(8,3)="K="                                                      007480
      PARAM(8,4)=B                                                         007490
*************************************************************************007500
****      COMPUTE MAXIMUM LIKELIHOOD FUNCTION                              007510
*************************************************************************007520
      LR(8)=0.0                                                            007530
      DO 75 J=1,J1                                                         007540
      FT=D*(B**NI1(J))*EXP(-D*(B**NI1(J))*TEM(J))                          007550
      LR(8)=LR(8)+ALOG(FT)                                                 007560
75    CONTINUE                                                             007570
*************************************************************************007580
*************************************************************************007590
      RETURN                                                               007600
      END                                                                  007610
      ----
```

```
**********************************************************************007620
****     COMPUTES f(t) FOR KEY EQUATION                                007630
****     TO ESTIMATE "K"                                               007640
**********************************************************************007650
*****                                                                  007660
      FUNCTION F1(X)                                                   007670
      REAL NI1                                                         007680
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)            007690
      ITNI=IFIX(TNI)                                                   007700
      F1=0.0                                                           007710
      DO 1 J=1,ITNI                                                    007720
      F1=F1+(COEF1(J)*(X**NI1(J)))                                     007730
1     CONTINUE                                                         007740
      RETURN                                                           007750
      END                                                              007760
      STOP
```

```
**********************************************************************007770
****     PARAMTER ESTIMATION FOR WEIBUL2                                007780
**********************************************************************007790
****                                                                    007800
      SUBROUTINE WEIBUL2(HEADER,SUM3B,SUM4B,TNE,M,PARAM,LR,FLAG)        007810
      IMPLICIT REAL(A-I,K-Z)                                            007820
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)             007830
      DIMENSION M(2200),PARAM(9,4),LR(9),HEADER(9,6)                    007840
**********************************************************************007850
****     ESTIMATION OF SHAPE AND SCALE PARAMETERS                       007860
**********************************************************************007870
      J1=TNI                                                            007880
      A=1./(((SUM4B/(TNE-1.)-(SUM3B**2.)/(TNE*(TNE-1)))*(.6079271))**.5)007890
      B=EXP(SUM3B/TNE+.5772/A)                                          007900
**********************************************************************007910
**********************************************************************007920
      HEADER(9,1)="WEIBULL SO"                                          007930
      HEADER(9,2)="FTWARE REL"                                          007940
      HEADER(9,3)="IABILITY M"                                          007950
      HEADER(9,4)="ODEL      "                                          007960
      HEADER(9,5)=10H                                                   007970
      HEADER(9,6)=10H                                                   007980
      PARAM(9,1)="ALPHA="                                               007990
      PARAM(9,2)=A                                                      008000
      PARAM(9,3)="BETA="                                                008010
      PARAM(9,4)=B                                                      008020
**********************************************************************008030
****     COMPUTE MAXIMUM LIKELIHOOD FUNCTION                            008040
**********************************************************************008050
      LR(9)=0.0                                                         008060
      DO 75  J=1,J1                                                     008070
      J2=M(J)                                                           008080
      DO 70 J3=1,J2                                                     008090
      X=-((TEM(J)/B)**A)                                                008100
      IF(X.GT.-675.)GO TO 50                                            008110
      LR(9)=-99999.                                                     008120
```

```
      GO TO 70                                                          008130
50    CONTINUE                                                          008140
      FT=(A/B)*((TEM(J)/B)**(A-1))*EXP(-(TEM(J)/B)**A)                  008150
      LR(9)=LR(9)+ALOG(FT)                                              008160
70    CONTINUE                                                          008170
75    CONTINUE                                                          008180
*******************************************************************************008190
*******************************************************************************008200
      RETURN                                                            008210
      END                                                               008220
```

```
************************************************************************008230
****     HYPOTHESIS TESTING                                             008240
****     SELECTS BEST DISTRIBUTION USING                                008250
****     LIKELIHOOD RATIO TEST                                          008260
************************************************************************008270
      SUBROUTINE HYPO(LR,J,K,L)                                         008280
      REAL LR                                                           008290
      DIMENSION LR(9)                                                   008300
      L=J                                                               008310
      DO 10 I=J,K                                                       008320
      IF(LR(L).LT.LR(I))L=I                                             008330
10    CONTINUE                                                          008340
      RETURN                                                            008350
      END                                                               008360
```

```
************************************************************************008370
****     PRINT SUBROUTINE                                                008380
****     PRINTS HEADERS,PARAMETERS,AND DISTRIBUTION                      008390
****     SELECTED                                                        008400
************************************************************************008410
      SUBROUTINE PRINT(PARAM,HEADER,J,J1,J9,FLAG,TNE,LR)                008420
      IMPLICIT REAL(A-I,K-Z)                                            008430
      COMMON TEM(2200),COEF(2200),TNI,COEF1(2200),NI1(2200)             008440
      DIMENSION FLAG(9),PARAM(9,4),LR(9),HEADER(9,6)                    008450
      WRITE 10,TNE,TNI                                                  008460
10    FORMAT(//,1X,' TOTAL NUMBER OF ERRORS TO DATE = ',F6.0,/,         008470
     $2X,'TOTAL NUMBER OF INTERVALS = ',F6.0)                           008480
20    FORMAT(/,1X,'**CONVERGENCE WAS NOT REACHED**',/,                  008490
     $1X,'**CALCULATIONS WERE STOPPED AT ',/,5X,'(N=10000+TOTAL'        008500
     $' NUMBER OF #STOPS#)**')                                          008510
      WRITE 25                                                          008520
25    FORMAT(/,1X,'PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:')   008530
      DO 50 J4=J,J1                                                     008540
      WRITE 40,(HEADER(J4,J2),J2=1,6)                                   008550
40    FORMAT(//,1X,6A10)                                                008560
      IF(FLAG(J4).EQ.0.0)GO TO 30                                       008570
      WRITE 20                                                          008580
30    CONTINUE                                                          008590
      WRITE 45,(PARAM(J4,J2),J2=1,4),LR(J4)                             008600
45    FORMAT(/,1X,A6,F13.7,3X,A5,F12.9,3X,'LLR=',F14.2)                 008610
50    CONTINUE                                                          008620
      WRITE 60,(HEADER(J9,J2),J2=1,6)                                   008630
60    FORMAT(//,1X,'THE LIKELIHOOD RATIO TEST HAS SELECTED:',           008640
     $/,1X,6A10)                                                        008650
      RETURN                                                            008660
      END                                                               008670
```

Appendix B

Sample Input for Program SRMOD

Given within this appendix is file F5C in its entirety and it represents typical input for Program SRMOD. The first entry of each record indicates the time interval between successive software problem reports. The second entry represents the number of errors occurring within this time interval.

Sample Input for Program SRMOD

```
  1.    1.
  9.    1.
 76.    1.
  8.    1.
 33.    1.
 14.    1.
  1.    5.
  1.    1.
  5.    2.
  6.    2.
  6.    1.
  7.    2.
  1.    1.
  3.    2.
  4.    2.
  2.    3.
  5.    1.
  2.    5.
  4.    2.
 10.    1.
 10.    1.
  1.    1.
  1.    1.
  1.    3.
  5.    1.
  2.    3.
  2.    1.
 11.    1.
  1.    1.
  8.    1.
 29.    1.
 20.    3.
  4.    1.
 11.    3.
  4.    2.
  2.    2.
  7.    1.
  8.    2.
  4.    1.
  7.    2.
  1.    2.
  1.    1.
  1.    1.
  4.    1.
  1.    1.
  5.    1.
  2.    1.
  7.    1.
  2.    1.
  4.    1.
  2.    1.
```

Appendix C

Program SRMOD Output

Within this appendix is the output from Program SRMOD for each of the twenty input files. Each output is labelled as to which error file it pertains to. There are two outputs for each file. One for Group I models and one for Group II models. "LLR" represents the natural logrithm of the maximum likelihood function. In some cases the maximum likelihood function became so small it was considered equal to zero by the computer and subsequently the logarithm could not be taken. To resolve this situation "LLR" was set equal to the arbitrary number -999999.0 whenever the logarithm could not be taken for this reason. The next 40 pages are the actual outputs from Progam SRMOD. All parameters are the same as those used in the text of this thesis.

```
FILE: EF1

 EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
   REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =  2172.
 TOTAL NUMBER OF INTERVALS =   152.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=       248.0010000    PHI=    .004852014    LLR=        -184.28


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       152.0010000    PHI=    .018812043    LLR=        -127.13


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=         1.1727845    K=      .994996064    LLR=        -185.34


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     3.5433276    BETA= 1.288208006    LLR=    -1000227.17


THE LIKELIHOOD RATIO TEST HAS SELECTED:
SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL
**
```

```
FILE: EF1

 EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =  2172.
 TOTAL NUMBER OF INTERVALS =   152.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      2546.1000000    PHI =  .009734999    LLR=       -1811.59


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      2172.1000000    PHI=   .024776264    LLR=       -3937.13


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      2176.1000000    PHI=   .000306270    LLR=       -1816.24


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=           .7792413   K=    1.000000225    LLR=        -189.87


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     8.1579818    BETA= 1.104085712    LLR=      -99999.00


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
**
```

```
    FILE: EF2

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   198.
 TOTAL NUMBER OF INTERVALS =    86.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10086.0010000   PHI=    .000034624   LLR=        -176.84


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10086.0010000   PHI=    .000005879   LLR=        -278.70


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .1843821   K=    1.017790064   LLR=        -166.95


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.6343261   BETA= 2.564747416   LLR=        -274.50


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
   FILE: EF2

    EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
   REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


   TOTAL NUMBER OF ERRORS TO DATE =   198.
   TOTAL NUMBER OF INTERVALS =     86.

  PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


  LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

  **CONVERGENCE WAS NOT REACHED**
  **CALCULATIONS WERE STOPPED AT
      (N=10000+TOTAL NUMBER OF #STOPS#)**

  N=     10198.1000000    PHI =  .000079119    LLR=        -217.30


  LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  **CONVERGENCE WAS NOT REACHED**
  **CALCULATIONS WERE STOPPED AT
      (N=10000+TOTAL NUMBER OF #STOPS#)**

  N=     10198.1000000    PHI=   .000013402    LLR=        -517.36


  MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       997.1000000    PHI=   .000007243    LLR=        -212.46


  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

  D=           .3481729   K=     1.000000225   LLR=        -176.73


  WEIBULL SOFTWARE RELIABILITY MODEL

  ALPHA=     1.7505832    BETA= 2.317399574    LLR=        -633.13


  THE LIKELIHOOD RATIO TEST HAS SELECTED:
  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
  **
```

```
FILE: EF1/2

 EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
   REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =  2370.
 TOTAL NUMBER OF INTERVALS =   238.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      362.0010000   PHI=   .002594701   LLR=       -357.52


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      263.0010000   PHI=   .001593007   LLR=   -1000157.19


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=        1.1039887   K=     .994998539   LLR=       -355.86


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    2.2821522   BETA= 1.658591296   LLR=   -1000139.45


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
**
```

```
FILE: EF1/2

 EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =  2370.
 TOTAL NUMBER OF INTERVALS =   238.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      2401.1000000   PHI =  .010385982   LLR=       -2012.38


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      2370.1000000   PHI=   .006958987   LLR=   -1000229.81


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      2370.1000000   PHI=   .000131312   LLR=       -1979.38


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=          .5774299   K=    1.000000225   LLR=        -368.62


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    4.5476435   BETA= 1.213738644   LLR=     -104017.74


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

```
   FILE: EF3A

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   144.
 TOTAL NUMBER OF INTERVALS =    54.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10054.0010000   PHI=    .000033415   LLR=        -113.04


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10054.0010000   PHI=    .000004598   LLR=        -190.19


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .1882861   K=     1.027401418   LLR=        -105.48


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.5693922   BETA= 2.515452292   LLR=        -183.00


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
    FILE: EF3A

     EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
    REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


    TOTAL NUMBER OF ERRORS TO DATE =   144.
    TOTAL NUMBER OF INTERVALS =    54.

   PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


   LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

   **CONVERGENCE WAS NOT REACHED**
   **CALCULATIONS WERE STOPPED AT
       (N=10000+TOTAL NUMBER OF #STOPS#)**

   N=     10144.1000000    PHI =  .000088541    LLR=        -150.17


   LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

   **CONVERGENCE WAS NOT REACHED**
   **CALCULATIONS WERE STOPPED AT
       (N=10000+TOTAL NUMBER OF #STOPS#)**

   N=     10144.1000000    PHI=   .000012158    LLR=        -371.43


   MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

   **CONVERGENCE WAS NOT REACHED**
   **CALCULATIONS WERE STOPPED AT
       (N=10000+TOTAL NUMBER OF #STOPS#)**

   N=     10144.1000000    PHI=   .000001103    LLR=        -136.65


   MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

   D=           .3354005   K=    1.000000225    LLR=        -112.99


   WEIBULL SOFTWARE RELIABILITY MODEL

   ALPHA=     1.8409500    BETA= 1.995204713    LLR=        -598.27


   THE LIKELIHOOD RATIO TEST HAS SELECTED:
   MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
  **
```

```
    FILE: EF3B

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =     75.
 TOTAL NUMBER OF INTERVALS =     30.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=     10030.0010000    PHI=    .000034818    LLR=         -61.60


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=     10030.0010000    PHI=    .000009208    LLR=         -82.15


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=           .2170610    K=     1.037888981    LLR=         -59.65


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.5983478    BETA= 2.735939264    LLR=         -74.64


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
   FILE: EF3B

    EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
   REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


   TOTAL NUMBER OF ERRORS TO DATE =     75.
   TOTAL NUMBER OF INTERVALS =     30.

  PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


  LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

  **CONVERGENCE WAS NOT REACHED**
  **CALCULATIONS WERE STOPPED AT
      (N=10000+TOTAL NUMBER OF #STOPS#)**

  N=     10075.1000000    PHI =  .000086801    LLR=         -79.13


  LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  **CONVERGENCE WAS NOT REACHED**
  **CALCULATIONS WERE STOPPED AT
      (N=10000+TOTAL NUMBER OF #STOPS#)**

  N=     10075.1000000    PHI=   .000022932    LLR=        -174.71


  MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       115.1000000    PHI=   .000277906    LLR=         -78.93


  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

  D=           .3488373   K=      .999999994   LLR=         -61.59


  WEIBULL SOFTWARE RELIABILITY MODEL

  ALPHA=     1.3858821    BETA= 3.162500962   LLR=        -200.62


  THE LIKELIHOOD RATIO TEST HAS SELECTED:
  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
 **
```

```
FILE: EF4

 EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
   REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =  1254.
 TOTAL NUMBER OF INTERVALS =   435.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      604.0010000   PHI=   .001086245   LLR=        -837.44


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      436.0010000   PHI=   .001028449   LLR=   -1000688.93


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .6017035   K=     .997989480   LLR=        -845.95


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.6803426   BETA= 2.538534110   LLR=       -1246.52


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL
..
```

FILE: EF4

EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL

TOTAL NUMBER OF ERRORS TO DATE = 1254.
TOTAL NUMBER OF INTERVALS = 435.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 1571.1000000 PHI = .001362447 LLR= -1204.93

LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 1256.1000000 PHI= .001109199 LLR= -2600.13

MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 1259.1000000 PHI= .000007872 LLR= -1179.66

MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D= .3714118 K= 1.000000225 LLR= -865.78

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= 1.8039613 BETA= 2.408449833 LLR= -3503.76

THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..

FILE: EF5A

EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
REQUIRING ONLY TIME TO ERROR AS INPUT DATA

TOTAL NUMBER OF ERRORS TO DATE = 432.
TOTAL NUMBER OF INTERVALS = 149.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 931.0010000 PHI= .000476236 LLR= -282.72

SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 694.0010000 PHI= .000300735 LLR= -303.79

JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D= .4395602 K= .998978879 LLR= -282.74

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= 1.9040927 BETA= 2.576061061 LLR= -310.03

THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL
..

```
FILE: EF5A

 EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =   432.
 TOTAL NUMBER OF INTERVALS =   149.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      1080.1000000    PHI =  .001391441    LLR=        -408.04


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       987.1000000    PHI=   .000714475    LLR=        -861.16


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       454.1000000    PHI=   .000044175    LLR=        -420.13


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=           .4070826   K=     1.000000225   LLR=        -282.90


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.9463397    BETA= 2.729025073    LLR=        -832.35


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

```
    FILE: EF5B

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   293.
 TOTAL NUMBER OF INTERVALS =   146.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10146.0010000   PHI=    .000019503   LLR=        -383.58


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10146.0010000   PHI=    .000001812   LLR=   -1000333.21


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .1121143   K=    1.008732806   LLR=        -373.45


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.3167573   BETA= 4.282208435   LLR=        -442.44


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
    FILE: EF5B

  EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
 REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =   293.
 TOTAL NUMBER OF INTERVALS =   146.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10293.1000000   PHI =  .000038787   LLR=       -429.08


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10293.1000000   PHI=   .000003599   LLR=       -920.82


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      915.1000000   PHI=   .000001399   LLR=       -436.78


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=          .1967605   K=    1.000000225   LLR=       -383.36


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.4295102   BETA= 3.463712479   LLR=       -832.92


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

```
    FILE: EF5C

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =     80.
 TOTAL NUMBER OF INTERVALS =     51.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=     10051.0010000    PHI=    .000013889    LLR=        -151.54


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=     10051.0010000    PHI=    .000001064    LLR=        -213.77


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=           .0756303   K=     1.028589567    LLR=        -146.74


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.2148178    BETA= 6.199801215    LLR=        -161.05


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
**
```

```
..
     FILE: EF5C

   EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
  REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


  TOTAL NUMBER OF ERRORS TO DATE =     80.
  TOTAL NUMBER OF INTERVALS =     51.

 PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


 LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

 **CONVERGENCE WAS NOT REACHED**
 **CALCULATIONS WERE STOPPED AT
     (N=10000+TOTAL NUMBER OF #STOPS#)**

 N=     10080.1000000    PHI =  .000021749    LLR=        -157.61


 LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

 **CONVERGENCE WAS NOT REACHED**
 **CALCULATIONS WERE STOPPED AT
     (N=10000+TOTAL NUMBER OF #STOPS#)**

 N=     10080.1000000    PHI=   .000001664    LLR=        -299.87


 MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

 N=       140.1000000    PHI=   .000012506    LLR=        -157.50


 MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

 D=           .1393443   K=      .999999998   LLR=        -151.51


 WEIBULL SOFTWARE RELIABILITY MODEL

 ALPHA=     1.2678031    BETA= 5.535613459   LLR=        -243.87


 THE LIKELIHOOD RATIO TEST HAS SELECTED:
 MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

FILE: EF5D

EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
REQUIRING ONLY TIME TO ERROR AS INPUT DATA

TOTAL NUMBER OF ERRORS TO DATE = 170.
TOTAL NUMBER OF INTERVALS = 83.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 112.0010000 PHI= .004527282 LLR= -182.51

SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 89.0010000 PHI= .002273557 LLR= -208.23

JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D= .5583315 K= .985014918 LLR= -182.75

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= 1.5293342 BETA= 3.509133908 LLR= -204.49

THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

```
   FILE: EF5D

    EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
   REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


   TOTAL NUMBER OF ERRORS TO DATE =   170.
   TOTAL NUMBER OF INTERVALS =    83.

  PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


  LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

  N=       206.1000000   PHI =  .005781998   LLR=       -210.52


  LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       175.1000000   PHI=   .002899614   LLR=       -403.16


  MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       171.1000000   PHI=   .000108849   LLR=       -226.90


  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

  D=           .2803987   K=    1.000000225   LLR=       -188.54


  WEIBULL SOFTWARE RELIABILITY MODEL

  ALPHA=     1.5812815   BETA= 3.512632233   LLR=       -394.68


  THE LIKELIHOOD RATIO TEST HAS SELECTED:
  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
 **
```

```
    FILE: EF5E

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   223.
 TOTAL NUMBER OF INTERVALS =   109.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10109.0010000   PHI=   .000024622   LLR=       -261.17


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10109.0010000   PHI=   .000002944   LLR=       -401.67


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .1935082   K=    1.004856851   LLR=       -259.51


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.3986271   BETA= 3.482394209   LLR=       -317.98


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
    FILE: EF5E

     EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
    REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


     TOTAL NUMBER OF ERRORS TO DATE =   223.
     TOTAL NUMBER OF INTERVALS =   109.

    PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


    LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

    **CONVERGENCE WAS NOT REACHED**
    **CALCULATIONS WERE STOPPED AT
        (N=10000+TOTAL NUMBER OF #STOPS#)**

    N=    10223.1000000   PHI =  .000050097   LLR=       -297.24


    LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

    **CONVERGENCE WAS NOT REACHED**
    **CALCULATIONS WERE STOPPED AT
        (N=10000+TOTAL NUMBER OF #STOPS#)**

    N=    10223.1000000   PHI=   .000005970   LLR=       -661.09


    MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

    N=      248.1000000   PHI=   .000023390   LLR=       -296.07


    MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

    D=          .2477213   K=    1.000000225   LLR=       -261.10


    WEIBULL SOFTWARE RELIABILITY MODEL

    ALPHA=    1.4959293   BETA= 3.489720447   LLR=       -641.54


     THE LIKELIHOOD RATIO TEST HAS SELECTED:
     MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
    *EOR
    **
```

```
    FILE: EF5F

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   207.
 TOTAL NUMBER OF INTERVALS =   102.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10102.0010000   PHI=   .000021590   LLR=       -257.84


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10102.0010000   PHI=   .000003736   LLR=       -327.59


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .2116517   K=    1.000498749   LLR=       -257.82


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.3914725   BETA= 4.415219075   LLR=       -274.46


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
   FILE: EF5F

  EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
 REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =   207.
 TOTAL NUMBER OF INTERVALS =   102.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=       425.1000000   PHI =  .001412641   LLR=       -293.35


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       925.1000000   PHI=   .000093097   LLR=       -568.81


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       214.1000000   PHI=   .000030102   LLR=       -298.58


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=           .2170157   K=    1.000000225   LLR=       -257.83


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.5045255   BETA= 4.179165058   LLR=       -525.59


 THE LIKELIHOOD RATIO TEST HAS SELECTED:
..
 MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

```
    FILE: EF5G

  EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
    REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   198.
 TOTAL NUMBER OF INTERVALS =    80.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10080.0010000   PHI=    .000010108   LLR=        -263.02


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

**CONVERGENCE WAS NOT REACHED**
**CALCULATIONS WERE STOPPED AT
    (N=10000+TOTAL NUMBER OF #STOPS#)**

N=    10080.0010000   PHI=    .000000691   LLR=        -356.08


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=          .0808674   K=    1.006034619   LLR=        -262.18


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    1.0130925   BETA= 8.081652568   LLR=        -265.35


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
   FILE: EF5G

    EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
   REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


   TOTAL NUMBER OF ERRORS TO DATE =    198.
   TOTAL NUMBER OF INTERVALS =     80.

  PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


  LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

  N=       538.1000000    PHI =  .000579303    LLR=        -310.98


  LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  **CONVERGENCE WAS NOT REACHED**
  **CALCULATIONS WERE STOPPED AT
      (N=10000+TOTAL NUMBER OF #STOPS#)**

  N=     10198.1000000    PHI=    .000001698    LLR=        -599.78


  MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       208.1000000    PHI=    .000009344    LLR=        -317.00


  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

  D=           .1015205    K=     1.000000225    LLR=        -263.00


  WEIBULL SOFTWARE RELIABILITY MODEL

  ALPHA=       .9783441    BETA= 7.429661570    LLR=        -643.07


  THE LIKELIHOOD RATIO TEST HAS SELECTED:
  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
  ..
```

```
FILE: EF5AH

 EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
   REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   432.
 TOTAL NUMBER OF INTERVALS =   131.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=     1020.0010000   PHI=   .001696781   LLR=        -67.87


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      476.0010000   PHI=   .009180697   LLR=        -43.09


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=        1.7285712   K=     .998994010   LLR=        -67.87


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=    2.1878506   BETA=  .677890384   LLR=        -52.64


THE LIKELIHOOD RATIO TEST HAS SELECTED:
SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL
..
```

```
FILE: EF5AH

 EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =   432.
 TOTAL NUMBER OF INTERVALS =   131.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      1283.1000000   PHI =  .005056635   LLR=        -212.86


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       933.1000000   PHI=   .017684491   LLR=        -619.92


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       463.1000000   PHI=   .000807953   LLR=        -233.33


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=         1.6178002   K=     1.000000225   LLR=         -67.97


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     2.0099102   BETA=  .718852616   LLR=        -205.50


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

```
FILE: EF5BH

 EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
   REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =   292.
 TOTAL NUMBER OF INTERVALS =   130.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      146.0010000   PHI=    .026925030   LLR=        -45.56


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      141.0010000   PHI=    .025678361   LLR=       -258.64


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=       13.1518735   K=      .972742772   LLR=        -26.77


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=      .8020556   BETA=  .457405985   LLR=        -54.28


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
FILE: EF5BH

 EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


 TOTAL NUMBER OF ERRORS TO DATE =   292.
 TOTAL NUMBER OF INTERVALS =   130.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=      409.2000000   PHI =  .015238470   LLR=       -110.42


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      392.2000000   PHI=   .013734957   LLR=       -620.39


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=      307.2000000   PHI=   .000874744   LLR=       -261.64


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=        1.6021067   K=    1.000000225   LLR=        -68.72


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     .8699890   BETA=  .441868617   LLR=        -86.38


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

FILE: EF5CH

EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
REQUIRING ONLY TIME TO ERROR AS INPUT DATA

TOTAL NUMBER OF ERRORS TO DATE = 80.
TOTAL NUMBER OF INTERVALS = 49.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 133.0010000 PHI= .008114052 LLR= -55.44

SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 103.0010000 PHI= .010094756 LLR= -77.63

JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D= 1.2835062 K= .984494710 LLR= -55.15

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= 1.1348635 BETA= 1.181769416 LLR= -55.76

THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..

```
FILE: EF5CH

 EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


TOTAL NUMBER OF ERRORS TO DATE =     80.
TOTAL NUMBER OF INTERVALS =     49.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=       127.1000000    PHI =  .017387951    LLR=          -62.77


LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=       111.1000000    PHI=   .019708765    LLR=          -164.79


MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=        84.1000000    PHI=   .001817107    LLR=          -93.20


MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D=           .8692568    K=      .999999998    LLR=          -55.87


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.1905967    BETA= 1.101664197    LLR=          -91.51


THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
..
```

```
FILE: EF5DH

 EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
   REQUIRING ONLY TIME TO ERROR AS INPUT DATA


 TOTAL NUMBER OF ERRORS TO DATE =    170.
 TOTAL NUMBER OF INTERVALS =     75.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N=       104.0010000    PHI=    .018365972    LLR=          -63.79


SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N=        80.0010000    PHI=    .043824873    LLR=          -71.40


JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D=         2.1701876    K=      .983265420    LLR=          -63.72


WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA=     1.6347346    BETA=   .945031468    LLR=          -77.41


THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..
```

```
   FILE: EF5DH

    EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
   REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL


   TOTAL NUMBER OF ERRORS TO DATE =   170.
   TOTAL NUMBER OF INTERVALS =    75.

  PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:


  LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

  N=       205.1000000    PHI =  .025091666    LLR=          -98.18


  LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       175.1000000    PHI=   .054954685    LLR=          -274.08


  MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

  N=       171.1000000    PHI=   .001963488    LLR=          -114.08


  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

  D=         1.0872455    K=     1.000000225   LLR=          -68.72


  WEIBULL SOFTWARE RELIABILITY MODEL

  ALPHA=     1.7219281    BETA=  .938994089    LLR=          -161.96


  THE LIKELIHOOD RATIO TEST HAS SELECTED:
  MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL
 **
```

FILE: EF5EH

EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
REQUIRING ONLY TIME TO ERROR AS INPUT DATA

TOTAL NUMBER OF ERRORS TO DATE = 223.
TOTAL NUMBER OF INTERVALS = 92.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 136.0010000 PHI= .014357752 LLR= -72.22

SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 118.0010000 PHI= .024763425 LLR= -99.55

JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D= 2.5625911 K= .984498442 LLR= -70.82

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= 1.2124251 BETA= .892276525 LLR= -75.56

THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..

FILE: EF5EH

EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL

TOTAL NUMBER OF ERRORS TO DATE = 223.
TOTAL NUMBER OF INTERVALS = 92.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 273.1000000 PHI = .021394947 LLR= -122.05

LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 252.1000000 PHI= .034917187 LLR= -371.89

MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 227.1000000 PHI= .001293842 LLR= -170.28

MODIFIED GEOMETRIC MODEL

D= 1.1776384 K= 1.000000225 LLR= -76.95

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= 1.2576628 BETA= .867753088 LLR= -174.28

THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC MODEL
..

FILE: EF5FH

EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
REQUIRING ONLY TIME TO ERROR AS INPUT DATA

TOTAL NUMBER OF ERRORS TO DATE = 207.
TOTAL NUMBER OF INTERVALS = 90.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 106.0010000 PHI= .023368186 LLR= -67.17

SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 103.0010000 PHI= .022910804 LLR= -157.92

JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D= 6.9393877 K= .965056505 LLR= -58.10

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= .8551609 BETA= .779779826 LLR= -76.41

THE LIKELIHOOD RATIO TEST HAS SELECTED:
JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..

FILE: EF5FH

EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL

TOTAL NUMBER OF ERRORS TO DATE = 207.
TOTAL NUMBER OF INTERVALS = 90.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 218.1000000 PHI = .036308260 LLR= -109.78

LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 215.1000000 PHI= .035881426 LLR= -406.03

MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 207.1000000 PHI= .001809664 LLR= -198.62

MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D= 1.1246817 K= 1.000000225 LLR= -79.42

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= .8218078 BETA= .657791428 LLR= -144.34

THE LIKELIHOOD RATIO TEST HAS SELECTED:
MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

FILE: EF5GH

EVALUATION OF GROUP I SOFTWARE RELIABILITY MODELS
REQUIRING ONLY TIME TO ERROR AS INPUT DATA

TOTAL NUMBER OF ERRORS TO DATE = 198.
TOTAL NUMBER OF INTERVALS = 76.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 78.0010000 PHI= .044198370 LLR= -48.82

SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 76.0010000 PHI= .028922954 LLR= -196.67

JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL

D= 14.4139897 K= .945673555 LLR= -32.41

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= .6378079 BETA= .599887132 LLR= -54.62

THE LIKELIHOOD RATIO TEST HAS SELECTED:
.JELINSKI-MORANDA GEOMETRIC SOFWARE RELIABILITY MODEL
..

FILE: EF5GH

EVALUATION OF GROUP II SOFTWARE RELIABILITY MODELS
REQUIRING TIME TO ERROR AND NUMBER OF ERRORS PER INTERVAL

TOTAL NUMBER OF ERRORS TO DATE = 198.
TOTAL NUMBER OF INTERVALS = 76.

PARAMETER ESTIMATIONS FOR THE TESTED MODELS ARE:

LIPOW-JELINSKI-MORANDA SOFTWARE RELIABILITY MODEL

N= 198.1000000 PHI = .068513675 LLR= -91.83

LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 198.1000000 PHI= .044056958 LLR= -431.00

MODIFIED-LIPOW-SCHICK-WOLVERTON SOFTWARE RELIABILITY MODEL

N= 198.1000000 PHI= .002860647 LLR= -185.42

MODIFIED GEOMETRIC SOFTWARE RELIABILITY MODEL

D= .9447691 K= 1.000000225 LLR= -80.32

WEIBULL SOFTWARE RELIABILITY MODEL

ALPHA= .6623245 BETA= .391920693 LLR= -62.85

THE LIKELIHOOD RATIO TEST HAS SELECTED:
WEIBULL SOFTWARE RELIABILITY MODEL

Appendix D

Program Time-To-Fix Input

Contained within this appendix is a portion of File5 used as input into Program Time-To-Fix, which was used to select a distribution for the software time-to-fix data files. The first six positions are the time it took to fix the software error. The seventh postion is the criticality of the error and the last five positions indicate the error category which was deemed appropriate for this particular software problem report.

Input for Program Time-To-Fix

```
.84DBB060
.84DBB060
.84CBB070
.84DBB080
.84CBB060
.85CBB020
.86DBB130
.86CBB070
.87DBB070
.87CBB070
.88DBB140
.88DBB120
.88DBB060
.88UBB070
.88CBB060
.88CBB140
.88CBB070
.90CBB070
.90CBB060
.90CBB070
.91DBB080
.92CBB080
.92CBB070
.92CBB070
.92CBB070
.93DBB060
.93DBB060
.93CBB070
.93DBB060
.93CBB140
.94DBB060
.94DBB080
.95CBB080
.95CBB140
.95CBB070
.95DBB140
.95CBB060
.95DBB070
.95CBB070
.97CBB070
.97DBB070
.99DBB140
.99DBB140
```

Appendix E

Program Time-To-Fix's Output

The next three pages are typical outputs from Program Time-To-Fix. The first is the result of using file FF3 as input without regards to error categories. The second page is the output for category "KK" of file FF5. As can be seen the log-normal distribution was selected. Finally, the third listing shows what happened when there was only one data point entered as data.

INPUT CONSISTED OF 219 POINTS

PART I. PARAMETER ESTIMATION

DISTRIBUTION	PARAMETER 1		PARAMETER 2	
UNIFORM	ALPHA =	.5000E+00	BETA =	.4600E+02
EXPONENTIAL	THETA =	.5000E+01	MU =	.5244E+01
WEIBULL	BETA =	.1080E+01	ALPHA =	.5244E+01
NORMAL	MU =	.5744E+01	SIGMA =	.7139E+01
LOG-NORMAL	MU =	.1123E+01	SIGMA =	.1133E+01
GAMMA	ALPHA =	.9407E+00	BETA =	.6106E+01
LOGISTIC	MU =	.5744E+01	SIGMA =	.7139E+01
EXT VALUE-SMALLEST	BETA=	.5566E+01	MU=	.8957E+01
EXT VALUE-LARGEST	BETA =	.5566E+01	MU =	.2532E+01
PARETO	K =	.5000E+00	A =	.5508E+00
LAPLACE	THETA =	.4000E+01	PHI =	.1744E+01

PART II. KOLMOGOROV-SMIRNOV TEST (MAX DVALUE = .37230)

DISTRIBUTION	MAX KSVALUE	DECISION
UNIFORM	.65922	REJECT
EXPONENTIAL	.21500	REJECT
WEIBULL	.15207	REJECT
NORMAL	.22671	REJECT
LOG-NORMAL	.13762	REJECT
GAMMA	.13327	REJECT
LOGISTIC	.20418	REJECT
EXT VALUE-SMALLEST	.25738	REJECT
EXT VALUE-LARGEST	.23224	REJECT
PARETO	.23485	REJECT
LAPLACE	.22927	REJECT

PART III. LIKLIHOOD RATIO TEST.

DISTRIBUTION	LN OF LIKELIHOOD FUNCTION
UNIFORM	-845.
EXPONENTIAL	-582.
WEIBULL	-603.
NORMAL	-741.
LOG-NORMAL	-594.
GAMMA	-601.
LOGISTIC	-738.
EXT VALUE-SMALLEST	999999.
EXT. VALUE-LARGEST	-681.
PARETO	-595.
LAPLACE	-810.

PART IV. DECISION.

NO DISTRIBUTION WAS SELECTED.

ANALYSIS OF CATEGORY 'KK'
INPUT CONSISTED OF 31 POINTS

PART I. PARAMETER ESTIMATION

DISTRIBUTION	PARAMETER 1		PARAMETER 2	
UNIFORM	ALPHA =	.1000E+01	BETA =	.8900E+02
EXPONENTIAL	THETA =	.1000E+01	MU =	.1029E+02
WEIBULL	BETA =	.9425E+00	ALPHA =	.7997E+01
NORMAL	MU =	.1129E+02	SIGMA =	.1837E+02
LOG-NORMAL	MU =	.1467E+01	SIGMA =	.1361E+01
GAMMA	ALPHA =	.6552E+00	BETA =	.1723E+02
LOGISTIC	MU =	.1129E+02	SIGMA =	.1337E+02
EXT VALUE-SMALLEST	BETA=	.1432E+02	MU=	.1956E+02
EXT VALUE-LARGEST	BETA =	.1432E+02	MU =	.3023E+01
PARETO	K =	.1000E+01	A =	.6818E+00
LAPLACE	THETA =	.3000E+01	PHI =	.8290E+01

PART II. KOLMOGOROV-SMIRNOV TEST (MAX DVALUE = .19218)

DISTRIBUTION	MAX KSVALUE	DECISION
UNIFORM	.65925	REJECT
EXPONENTIAL	.35901	REJECT
WEIBULL	.21435	REJECT
NORMAL	.29155	REJECT
LOG-NORMAL	.16674	ACCEPT
GAMMA	.19290	REJECT
LOGISTIC	.29823	REJECT
EXT VALUE-SMALLEST	.36165	REJECT
EXT VALUE-LARGEST	.28383	REJECT
PARETO	.29032	REJECT
LAPLACE	.36057	REJECT

PART III. LIKLIHOOD RATIO TEST.

DISTRIBUTION	LN OF LIKELIHOOD FUNCTION
UNIFORM	-138.
EXPONENTIAL	-103.
WEIBULL	-106.
NORMAL	-134.
LOG-NORMAL	-99.
GAMMA	-104.
LOGISTIC	-130.
EXT VALUE-SMALLEST	999999.
EXT VALUE-LARGEST	-125.
PARETO	-88.
LAPLACE	-124.

PART IV. DECISION.

THE LIKLIHOOD RATIO AND KOLMOGOROV-SMIRNOV TESTS HAVE SELECTED THE LOG-NORMAL DISTRIBUTION. PARAMETERS ARE
MU = .1467E+01 SIGMA = .1361E+01

ANALYSIS OF CATEGORY 'AA100'
INPUT CONSISTED OF 1 POINTS
NOT ENOUGH DATA POINTS FOR FURTHER ANALYSIS

Appendix F

Software Error Categories

This appendix contains a listing of the software error categories used by the contractors during error collection procedures (Ref [3017]:45-50).

ERROR CATEGORIES

CATEGORY ID	CATEGORIES
AA000	COMPUTATIONAL ERRORS
AA010	TOTAL NUMBER OF ENTRIES COMPUTED INCORRECTLY
AA020	PHYSICAL OR LOGICAL ENTRY NUMBER COMPUTED INCORRECTLY
AA040	WRONG EQUATION OR CONVENTION USED
AA041	MATHEMATICAL MODELING PROBLEM
AA050	RESULTS OF ARITHMETIC CALCULATION INACCURATE/NOT AS EXPECTED
AA060	MIXED MODE ARITHMETIC ERROR
AA070	TIME CALCULATION ERROR
AA071	TIME CONVERSION ERROR
AA072	TIME TRUNCATION/ROUNDING ERROR
AA080	SIGN CONVENTION ERROR
AA090	UNITS CONVERSION ERROR
AA100	VECTOR CALCULATION ERROR
AA110	CALCULATION FAILS TO CONVERGE
AA120	QUANTIZATION/TRUNCATION ERROR
BB000	LOGIC ERRORS
BB010	LIMIT DETERMINATION ERROR
BB020	WRONG LOGIC BRANCH TAKEN
BB030	LOOP EXITED ON WRONG CYCLE
BB040	INCOMPLETE PROCESSING
BB050	ENDLESS LOOP DURING ROUTINE OPERATION
BB060	MISSING LOGIC OR CONDITION TEST
BB061	INDEX NOT CHECKED
BB062	FLAG OR SPECIFIC DATA VALUE NOT TESTED
BB070	INCORRECT LOGIC
BB080	SEQUENCE OF ACTIVITIES WRONG
BB090	FILTERING ERROR
BB100	STATUS CHECK/PROPAGATION ERROR
BB110	ITERATION STEP SIZE INCORRECTLY DETERMINED
BB120	LOGICAL CODE PRODUCED WRONG RESULTS
BB130	LOGIC ON WRONG ROUTINE
BB140	PHYSICAL CHARACTERISTICS OF PROBLEM TO BE SOLVED, OVERLOOKED, OR MISUNDERSTOOD
BB150	LOGIC NEEDLESSLY COMPLEX
BB160	INEFFICIENT LOGIC
BB170	EXCESSIVE LOGIC
BB180	STORAGE REFERENCE ERROR (SOFTWARE PROBLEM)

ERROR CATEGORIES

CATEGORY ID	CATEGORIES
CC000	I/O ERRORS
CC010	MISSING OUTPUT
CC020	OUTPUT MISSING DATA ENTRIES
CC030	ERROR MESSAGE NOT OUTPUT
CC040	ERROR MESSAGE GARBLED
CC050	OUTPUT OR ERROR MESSAGE NOT COMPATIBLE WITH DESIGN DOCUMENTATION (INCLUDING GARBLED OUTPUT)
CC060	MISLEADING OR INACCURATE ERROR MESSAGE TEXT
CC070	OUTPUT FORMAT ERROR (INCLUDING WRONG LOCATION)
CC080	DUPLICATE OR EXCESSIVE OUTPUT
CC090	OUTPUT FIELD SIZE INADEQUATE
CC100	DEBUG OUTPUT PROBLEM (RELATIVE TO DESIGN DOCUMENTATION)
CC101	LACK OF DEBUG OUTPUT
CC102	TOO MUCH DEBUG
CC110	HEADER OUTPUT PROBLEM
CC120	OUTPUT TAPE FORMAT ERROR
CC130	OUTPUT CARD FORMAT ERROR
CC140	ERROR IN PRINTER CONTROL
CC150	LINE COUNT/PAGE EJECT ERROR
CC160	NEEDED OUTPUT NOT PROVIDED IN DESIGN
CC161	INSUFFICIENT OUTPUT OPTIONS
DD000	DATA HANDLING ERRORS
DD010	VALID INPUT DATA IMPROPERLY SET/USED
DD020	DATA WRITTEN IN OR READ FROM WRONG DISK LOCATION
DD030	DATA LOST/NOT STORED
DD040	DATA, INDEX OR FLAG NOT SET OR SET/INITIALIZED INCORRECTLY
DD041	NUMBER OF ENTRIES SET INCORRECTLY
DD050	DATA, INDEX OR FLAG MODIFIED OR UPDATED INCORRECTLY
DD051	NUMBER OF ENTRIES UPDATED INCORRECTLY
DD060	EXTRANEOUS ENTRIES GENERATED (TABLE, ARRAY, ETC.)
DD070	BIT MANIPULATION ERROR
DD071	ERROR USING BIT MODIFIER
DD080	FLOATING POINT/INTEGER CONVERSION ERROR
DD090	INTERNAL VARIABLE ERROR (DEFINITION OR SET/USE)
DD100	DATA PACKING/UNPACKING ERROR
DD110	ROUTINE LOOKING FOR DATA IN NON-EXISTENT RECORD
DD120	BOUNDS VIOLATION
DD130	DATA CHAINING ERROR
DD140	DATA OVERLOW OR OVERFLOW PROCESSING ERROR
DD150	READ ERROR
DD151	ALL AVAILABLE DATA NOT READ
DD160	LONG LITERAL PROCESSING ERROR
DD170	SORT ERROR
DD180	OVERLAY ERROR
DD190	SUBSCRIPTING CONVENTION ERROR
DD200	DOUBLE BUFFERING ERROR

ERROR CATEGORIES

CATEGORY ID	CATEGORIES
EE000	OPERATING SYSTEM/SYSTEM SUPPORT SOFTWARE ERRORS
EE010	JOVIAL PRODUCES ERRONEOUS MACHINE CODE
EE020	OS MISSING NEEDED CAPABILITY
FF000	CONFIGURATION ERRORS
FF010	COMPILATION ERROR
FF011	SEGMENTATION PROBLEM
FF020	ILLEGAL INSTRUCTION
FF030	UNEXPLAINABLE PROGRAM HALT
GG000	ROUTINE/ROUTINE INTERFACE ERRORS
GG010	ROUTINE PASSING INCORRECT AMOUNT OF DATA (INSUFFICIENT OR TOO MUCH)
GG020	ROUTINE PASSING WRONG PARAMETERS OR UNITS
GG030	ROUTINE EXPECTING WRONG PARAMETERS
GG040	ROUTINE FAILS TO USE AVAILABLE DATA
GG050	ROUTINE SENSITIVE TO INPUT DATA ORDER
GG060	CALLING SEQUENCE OR ROUTINE/ROUTINE INITIALIZATION ERROR
GG070	ROUTINES COMMUNICATING THROUGH WRONG DATA BLOCK
GG080	ROUTINE USED OUTSIDE DESIGN LIMITATION
GG090	ROUTINE WON'T LOAD (ROUTINE INCOMPATIBILITY)
GG100	ROUTINE OVERFLOWS CORE WHEN LOADED
HH000	ROUTINE/SYSTEM SOFTWARE INTERFACE ERRORS
HH010	OS INTERFACE ERROR (CALLING SEQUENCE OR INIALIZATION)
HH020	ROUTINE USES EXISTING SYSTEM SUPPORT SOFTWARE INCORRECTLY
HH030	ROUTINE USES SENSE/JUMP SWITCH IMPROPERLY
II000	TAPE PROCESSING INTERFACE ERROR
II010	TAPE UNIT EQUIPMENT CHECK NOT MADE
II020	ROUTINE FAILS TO READ CONTINUATION TAPE
II030	ROUTINE FAILS TO UNLOAD TAPE AFTER COMPLETION
II040	ERRONEOUS INPUT TAPE FORMAT

ERROR CATEGORIES

CATEGORY ID	CATEGORIES
JJ000	USER INTERFACE ERRORS
JJ010	OPERATIONS REQUEST OR DATA CARD/ROUTINE INCOMPATABILITY
JJ020	MULTIPLE PHYSICAL CARD/LOGICAL CARD PROCESSING ERROR
JJ030	INPUT DATA INTERPRETED INCORRECTLY BY ROUTINE
JJ040	VALID INPUT DATA REJECTED OR NOT USED BY ROUTINE
JJ050	INPUT DATA REJECTED BUT USED
JJ060	INPUT DATA READ BUT NOT USED
JJ070	ILLEGAL INPUT DATA ACCEPTED AND PROCESSED
JJ080	LEGAL INPUT DATA PROCESSED INCORRECTLY
JJ090	POOR DESIGN IN OPERATOR INTERFACE
JJ100	INADEQUATE INTERRUPT AND START CAPABILITY
KK000	DATA BASE INTERFACE ERRORS
KK010	ROUTINE/DATA BASE INCOMPATIBILITY
KK011	UNCOORDINATED USE OF DATA ELEMENTS BY MORE THAN ONE USER
LL000	USER REQUESTED CHANGES
LL010	SIMPLIFIED INTERFACE AND/OR CONVENIENCE
LL020	NEW AND/OR ENHANCED FUNCTIONS
LL021	CPU
LL022	DISK
LL023	TAPE
LL024	I/O
LL025	CORE
LL030	SECURITY
LL040	NEW HARDWARE/OS CAPABILITY
LL050	INSTRUMENTATION
LL060	CAPACITY
LL070	DATA BASE MANAGEMENT AND INTEGRITY
LL080	EXTERNAL PROGRAM INTERFACE
MM000	PRESET DATA BASE ERRORS
MM010	DATA OR OPERATIONS REQUEST CARD DESCRIPTIONS
MM020	ERROR MESSAGE TEXT
MM030	NOMINAL, DEFAULT, LEGAL, MAX/MIN VALUES
MM040	PHYSICAL CONSTANTS AND MODELING PARAMETERS
MM041	EPHEMERIS PARAMETERS
MM050	DICTIONARY (BIT STRING) PARAMETERS
MM060	MISSING DATA BASE SETTINGS

ERROR CATEGORIES

CATETORY ID	CATEGORIES
NN000	GLOBAL VARIABLE/COMPOOL DEFINITION ERRORS
NN010	ITEMS IN WRONG LOCATION (WRONG DATA BLOCK)
NN011	DEFINITION SEQUENCE ERROR
NN020	DATA DEFINITION ERROR
NN021	TABLE DEFINITION INCORRECT
NN030	LENGTH OF DEFINITION INCORRECT
NN040	COMMENTS ERROR
NN050	DELETE UNNEEDED DEFINITIONS
PP000	RECURRENT ERRORS
PP010	PROBLEM REPORT REOPENED
PP020	PROBLEM REPORT A DUPLICATE OF PREVIOUS REPORT
QQ000	DOCUMENTATION ERRORS
QQ010	ROUTINE LIMITATION
QQ020	OPERATING PROCEDURES
QQ030	DIFFERENCE BETWEEN FLOW CHART AND CODE
QQ040	TAPE FORMAT
QQ050	DATA CARD/OPERATION REQUEST CARD FORMAT
QQ060	ERROR MESSAGE
QQ070	ROUTINE'S FUNCTIONAL DESCRIPTION
QQ080	OUTPUT FORMAT
QQ090	DOCUMENTATION NOT CLEAR/NOT COMPLETE
QQ100	TEST CASE DOCUMENTATION
QQ110	OPERATING SYSTEM DOCUMENTATION
QQ120	TYPO/EDITORIAL ERROR/COSMETIC CHANGE
RR000	REQUIREMENTS COMPLIANCE ERRORS
RR010	EXCESSIVE RUN TIME
RR020	REQUIRED CAPABILITY OVERLOOKED OR NOT DELIVERED AT TIME OF REPORT

ERROR CATEGORIES

CATEGORY ID	CATEGORIES
SS000	UNIDENTIFIED ERRORS
TT000	OPERATOR ERROR
TT010	TEST EXECUTION ERROR
TT020	ROUTINE COMPILED AGAINST WRONG COMPOOL/MASTER COMMON
TT030	WRONG DATA BASE USED
TT040	WRONG MASTER CONFIGURATION USED
TT050	WRONG TAPE(S) USED
UU000	QUESTIONS
UU010	DATA BASE
UU020	MASTER CONFIGURATION
UU030	ROUTINE
V	HARDWARE
X	NON-REPRODUCIBLE

References

1. Barlow, R. H. and F. Porschan. Mathematical Theory of Reliability. New York: John Wiley & and Sons, Inc, 1965

2. Caplen, R. A Practical Approach to Reliability London, England: Business Books Limited, 1972.

3. Corcoran, W. J., H. Weingarten, P. W. Zehna. "Estimating Reliability After Corrective Action," Management Science, 10: 786-795 (July 1964).

4. Dumonceaux, R. (St. Johns University), C. E. Antle (The Pennsylvania State University), G. Haas (Northrop Corporation and West Coast University), "Likelihood Ratio Test for Discrimination Between Two Models with Unkown Location and Scale Parameters," Air Force Aerospace Research Laboratories, Air Force Systems Command, United States Air Force, Contract No. F33615-71-C-1169.

5. Dryer, A. R. "Hypothesis Testing Procedures for Seperate Families of Hypotheses," Journal of the American Statistical Association, 69: 140-145 (March 1974).

6. Farnan, D. N. (Defense Systems Management School). "Relaible Computer Software. What Is It and How to Get It," In House Report, November 1975. (AD-A026988).

7. Fries, M. J. "Software Error Data Acquisition," In House Report, Boeing Aerospace Company, April 1977. (RADC-TR-77-130).

8. Jelinski, Z., P. Moranda "Software Reliability Research," published in Statistical Computer Performance Evaluation, edited by W. Freiberger. New York, New York: Academic Press, 1972.

9. La Padula, L. J. "Engineering of Quality Software Systems (Software Relaibility Modeling and Measurement Techniques)," In House Report, Mitre Corporation, January 1975. (AD-A007773).

10. Lilliefors, H. W. "On the Kolmogorov-Smirnov Test for Normality With Mean and Variance Unkown," American Statistical Association Journal,: 399-402 (June 1967).

11. Lipow, M. "Maximum Likelihood Estimation of Parameters of a Software Time-To-Failure Distribution," TRW Systems Group, TRW Report No. 2260.1.9-73D-15 (RW1), June 1973.

12. Lipow, M. "Some Variations of a Model for Software Time-To-Failure, TRW Systems Group, Correspondence ML-74-2260.1.9-21, August 1974.

13. Massey, F. J. "The Kolmogorov-Smirnov Test For Goodness of Fit," American Statistical Association Journal,: 68-78 (March 1951).

14. Moranda, P. B. "Prediction of Software Reliability During Debugging," Proceedings 1975 Annual Reliability and Maintainability Symposium, Washington, D. C., January 1975.

15. Moranda, P. B. "Probability-Based Models for the Failures During Burn-In Phase," Joint National Meeting ORSA/TIMS, Las Vegas, New Mexico, November 1975. (AD-A020706).

16. Nelson E. C. "A Statistical Basis for Software Reliability Assessment," TRW Software Series, TRW-SS-73-03, March 1973.

17. Randell, B. "System Structure for Software Fault Tolerance," IEEE Software Engineering, SE1,2: 220-232 (June 1975).

18. Regulinski, T. L. Program Model - A computer program written for hypothesis testing of probability density functions. Given at the Air Force Institute of Technology Wright Patterson AFB, Ohio, 1977.

19. Regulinski, T. L. Lecture notes from Software Reliability, EE7.53. Given at the Air Force Institute of Technology, Wright Patterson AFB, Ohio, 1977.

20. Regulinski, T. L. "Optimisation of Reliability Processes" Thesis submitted to the University of Bradford, 191-195, September 1976.

21. Rudner, B. "Seeding/Tagging Estimation of Software Errors : Models and Estimates," In House Report, January 1977. (AD-A036655).

22. Shooman, M. L. "Software Reliability Research," published in Statistical Computer Performance Evaluation, edited by W. Freiberger. New York, New York: Academic Press, 1972.

23. Sukert, Allen N. "A Software Reliability Modeling Study," In House Report, Rome Air Development Center, Griffis AFB, New York, August 1976. (AD-A030798).

24. Thayer, T. A., G. R. Craig, L. E. Frey, W. L. Hetrick, and M. Lipow. "Software Reliability Study," In House Report, TRW Defense and Space Systems Group, 5-2, August 1976. (AD-A030798).

25. Trivedi, A. K. and M. L. Shooman (Polytechnic Institute of New York). "Computer Software Reliability : Many-State Markov Modeling Techniques," In House Report, July 1975. (AD-A014824).

26. Ulsamer, E. "Computers - Key to Tomorrow's Air Force," Air Force Magazine, 56: 52 (june 1973).

27. Wagoner, W. L. "The Final Report on a Software Reliability Measurement Study," In House Report, The Aerospace Corporation, Report No. TOR-0074(4112)-1, August 1973.

28. Weiss, H. K. "Estimation of Reliabiity Growth in a Complex System with a Poisson-Type Failure," Operations Research, 4: 532-545 (October 1956).

29. William, H. E., T. A. James, A. A. Beaureguard, P. Hilcoff, "Software Systems Reliability: A Raytheon Project History," In House Report, Raytheon Company, June 1977. (RADC-TR-77-188).

30. Wolverton, R. W. and G. J. Schick, "Assessment of Software Reliability," TRW Software Series, TRW-SS-73-04, September 1972.

Vita

Steve G. Castle was born on 2 December 1948 in Ilion, New York. After graduating from Ilion Central High School in 1966, he continued his education at Utica College of Syracuse University, Utica, New York, from which he was graduated with a Bachelor of Arts degree in Mathematics on 6 June 1970. Upon graduation, he enlisted in the Air Force and served as an Accounting and Finance Specialist at Myrtle Beach AFB, South Carolina. In 1972 he attended Officer Training School and was commissioned to be a Second Lieutenant on 13 October 1972. From 1972 to 1976 he served as a Computer Programmer and System Analyst at Headquarters Air Training Command, Randolph AFB, Texas for which he received the Air Force Commendation Medal. He was assigned to the Air Force Institute of Technology in October, 1976.

Permanent address: 147 Oliver Drive
New Smyrna Beach, Fla.
32069

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/GCS/EE/78-2	2. GOVT ACCESSION NO. (9)
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): (6) SOFTWARE RELIABILITY: MODELLING TIME-TO-ERROR AND TIME-TO -FIX.	5. TYPE OF REPORT & PERIOD COVERED: Masters thesis, MS Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): (19) Steve G. Castle	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Institute of Technology (AFIT-EN) Wright Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (RADC/ISIS) Griffiss AFB, New York	12. REPORT DATE: Mar[redacted]78 (11)
	13. NUMBER OF PAGES: ~~153~~ 182 P. (12)
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public releases; IAW AFR 190-17

JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software Reliability
Reliability
Computer Software
Modelling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Various competing software reliability models were tested utilizing time-to-error data extracted from a large data base provided by Rome Air Development Center (RADC), Griffiss Air Force Base, New York. Only those models for which appropriate information was available from the RADC data base were used. Each model that was tested had its paramters estimated and discrimination between the models was performed by the Likelihood Ratio Test. Code for the computer program used in parameter estimation and Likelihood Ratio Testing is included. The same large data base was also used in determination of the probability functions

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A122 25

hc

governing the random variable time-to-fix. Time-to-fix data was extracted from data base and was sorted by error categories. Eleven of the more common probability density functions, along with the Kolmogorov-Smirnov Test and the Likelihood Ratio Test, were used to isolate the distribution that best modelled software time-to-fix for each particular data set.