A METHODOLOGY FOR SOFTWARE RELIABILITY PREDICTION AND QUALITY CONTROL

Norman F. Schneidewind

Naval Postgraduate School
Monterey, California

November 1972

COMPUTER PRODUCTS GROUP - Pr cing Form

CPG -_____ GRA Accession # AD-754377 Date Received _____

One Time Costs			
Normal NTIS Processing $30.00	CPG Evaluatic.	Subtotal	Predicted No. of Sales (2)
Purchase Price (1)	CPG Announcement	Less Contrib. Subsidy	Unit Cost
Archival	CPG Marketing	Total Cost	Evaluator's Initials

Value of Information		Cost to Fill Each Order	
Tech Report @ 05¢/ g Documentation @ 2 g Data @ 1¢/Kbyte Source Deck @ 1¢/Stmnt Object Deck @ 10¢/card	Qty @ Value	Order Processing CPG Processing Shipping & Mailing	Time & Material (4) Total PEC CPG Cost Per Input
Multiplier (3)	Adjusted Value	Royalty	Sales Price
Approved by		Date	Fiche PRICE

(1) Explanation of Purchase See attached form

(2) Basis of Prediction

(3) Basis if not Equal to 1

(4) Computation:
Documentation @ 5¢/pg
Cards @ 1¢/card
Minireels @ $10/reel
Regular Tape @ $25/reel

Qty	
@	

NTIS-CPC-73-36 (TEST FORM)

NPS-55SS72111A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

A METHODOLOGY FOR SOFTWARE RELIABILITY PREDICTION AND QUALITY CONTROL

by

Norman F. Schneidewind

November 1972

Approved for public release; distribution unlimited.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

(Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b GROUP

3 REPORT TITLE

A Methodology for Software Reliability Prediction and Quality Control

4 DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Technical Report, 1972

5 AUTHOR(S) (First name, middle initial, last name)

Norman F. Schneidewind

6 REPORT DATE November 1972	7a. TOTAL NO OF PAGES 48	7b. NO OF REFS 0
8a CONTRACT OR GRANT NO b PROJECT NO P509001 c d.	9a. ORIGINATOR'S REPORT NUMBER(S) NPS-55SS72111A 9b OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY Computer Sciences Department Naval Electronics Laboratory Center

13 ABSTRACT

The increase in importance of software in command and control and other complex systems requires increased attention to the problems of software reliability and quality control. This paper reports on initial attempts to develop a methodology for Naval Tactical Data System software reliability and presents the results of several statistical analyses which were performed in order to obtain an appreciation for the statistical characteristics of software reliability data. An approach to analyzing software reliability problems is outlined and a methodology for reliability prediction and quality control is presented. Characteristics of software reliability statistical distributions are reported.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED

Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Software Reliability						
Software Quality Control						
Software Reliability Prediction						
Software Measurements						

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral Mason Freeman, USN
Superintendent

M. U. Clauser
Provost

ABSTRACT:

The increase in importance of software in command and control and other complex systems requires increased attention to the problems of software reliability and quality control. This paper reports on initial attempts to develop a methodology for Naval Tactical Data System software reliability and presents the results of several statistical analyses which were performed in order to obtain an appreciation for the statistical characteristics of software reliability data. An approach to analyzing software reliability problems is outlined and a methodology for reliability prediction and quality control is presented. Characteristics of software reliability statistical distributions are reported.

This task was supported by the Naval Electronics Laboratory Center under Job Order 55436.

Norman F. Schneidewind
Norman F. Schneidewind
Professor of Information Systems

Approved by:

J. R. Borsting, Chairman
Department of Operations Research and Administrative Sciences

Released by:

J. M. Wozencraft
Dean of Research

NPS-55SS72111A

November 1972

TABLE OF CONTENTS

I. Summary.

The increase in importance of software in command and control and other complex systems has not been accompanied by commensurate progress in the development of analytical techniques for the measurement of software quality and the prediction of software reliability. In recognition of the disparity, the Computer Sciences Department of the Naval Electronics Laboratory Center, San Diego, California is sponsoring this software quality control and reliability research project.

The objectives of the research are to develop procedures for controlling software quality and to develop a methodology for predicting software reliability. The data which were employed were Naval Tactical Data System (NTDS) Trouble Reports and supporting documentation.

In order to accomplish these objectives, it was necessary to perform many statistical analyses of NTDS test data. The major analyses are listed below.

- analysis of the number of software troubles per unit time[1] as a function of cumulative test time[2];
- analysis of the distribution of time between troubles[3] and the distribution of number of troubles per unit time;
- goodness of fit tests for identifying theoretical reliability functions which might be appropriate for reliability prediction;
- estimation of reliability function parameters;

[1]Number of troubles per unit time is the number of troubles occurring in a program test computer time interval divided by the time interval.

[2]Cumulative test time is the total computer time used to date in testing a single program.

[3]Time between troubles is the cumulative computer time between two consecutive troubles. The two consecutive troubles may occur in the same or different test runs.

- estimation of confidence limits for reliability function parameters and reliability functions;
- analysis of program reliability variability between and among programs; and
- development of equations for estimating reliability and test requirements.

With respect to NTDS programs, the research to date suggests these conclusions: (1) software reliability prediction is feasible but that much more analysis is required in order to validate the approaches which have been developed; (2) there is greater variability in program reliability between programs than there is within programs; (3) in general, the occurrence of software troubles is not a stationary process; and (4) there is no single probability distribution which typifies the occurrence of software troubles.

II. Objectives and Approach.

One objective was to determine the feasibility of predicting software reliability based on the use of program test results. A second objective was to identify quantitative measures of program quality which could be used in software quality control. A third objective was to investigate methods for estimating the amount of test time which is required in order to satisfy program reliability requirements. Test time estimates are needed at two stages: (1) prior to the commencement of program testing when, based on reliability requirements, it is necessary to make an initial

estimate of required test time and (2) during testing when, based on reliability requirements and the reliability achieved to date, it is necessary to make an estimate of the required remaining test time.

Two other areas of investigation involved the analysis of sources of program reliability variation and the identification of the appropriate program sampling unit to use for reliability analysis.

Software trouble reports were associated with scheduled test time in order to obtain distributions of time between troubles. The most important distribution, from the standpoint of reliability prediction, is time between troubles, since $Q(t) = \int_0^t f(t')dt'$ and $R(t) = 1 - Q(t)$, where $f(t)$ is the density function of time between troubles, t is program operating time, $Q(t)$ is unreliability and $R(t)$ is reliability. Thus, if a theoretical density function $f(t)$ can be fitted to the empirical relative frequency distribution, an estimate of the reliability function can be obtained. If no theoretical density function is suitable, the empirical relative frequency distribution of time between troubles can be summed to obtain an estimate of the unreliability function from which an estimate of the reliability function can be obtained. Either a theoretical or empirical reliability function can be used to predict the reliability of a program for various program operating times. However, if a theoretical reliability function can be used, confidence interval estimates can be obtained for the theoretical reliability function parameters.

This is important because it is then possible to estimate the reliability function parameters which are necessary to achieve stated reliability objectives. With parameter estimates available, it is also possible to estimate the amount of test time, as a function of number of troubles, which will be required in order to achieve reliability objectives.

It is possible to employ distribution-free methods and empirical reliability functions to estimate confidence limits for the reliability function. This approach provides the capability of comparing the desired reliability function (reliability objective) with the empirical reliability function and its confidence limits, but, with no theoretical reliability function available, there is no capability for making those parameter estimates which are of interest in reliability analysis.

In order to identify the major contributors of program reliability variability, an Analysis of Variance test was employed. Additionally, goodness of fit tests were conducted for various relative frequency distributions of time between troubles and program run time in order to identify the type of distribution which is applicable to software failures.

A problem of sampling arises due to the possible non-randomness of sample selection. In the case of program testing, randomness would mean that each part of a program has an equal probability of being tested. However, in practice, samples are not "drawn" in the usual sense; rather, inputs and program segments are selected for testing based

on the criticality of the segment to mission success, or for some less objective reason. Some program segments will be more intensively tested than others. Whatever the criteria employed for program testing, it is clear that the various program segments do not have equal probability of selection.

III. Data Analysis.

A. Trouble Rates and Program Run Time

The first analysis which was performed was to examine the pattern of trouble rates as a function of cumulative test time. This was done in order to ascertain whether trouble rates decrease and eventually stabilize or whether they continue to fluctuate as testing continues. The achievement of an approximately constant trouble rate, after a period of testing has elapsed, would indicate that the occurrence of troubles has stabilized and that the major troubles have been identified and corrected.

Two types of trouble rates were analyzed. One is a weekly trouble rate and the second is a cumulative trouble rate. The first rate is the number of troubles detected during a week divided by the amount of computer test time expended during the week. This rate provides an indication of short term fluctuations the rate of detecting software troubles. The second rate is the total number of troubles which have occurred since testing began divided by the total elapsed test time. This rate provides an indication of whether the long term trouble rate is decreasing, constant or increasing. A decreasing rate would indicate that program

reliability increases with increases in testing.

Another random variable which was analyzed is program run time. Program run time is the elapsed computer time from start of program to the occurrence of a trouble. Hence, the random variable program run time only applies when a trouble occurs during a test. Program run time was used in Analysis of Variance tests for estimating the relative contributions to variations in program reliability due to differences between and among programs.

An analysis of program run time can also be used to indicate whether program reliability increases as testing continues. As testing progresses, we would expect to observe a gradual increase in program run time and an eventual stabilization around a mean value.

All statistical estimates presented in this report are based on total number of troubles. In NTDS testing, troubles are classified according to High (software unuseable) Medium (major limitation) and Low (minor limitation) severity categories. The trouble reports were not segregated by category of trouble because the initial interest was to obtain an overall picture of trouble occurrence distributions; secondly, sample sizes are considerably reduced if troubles are analyzed by categories.

Data concerning the occurrence of troubles and mean program run time for several programs of Ship 1 are shown in Figure 1 and Figure 2. The trouble rate shown in Figure 1 is

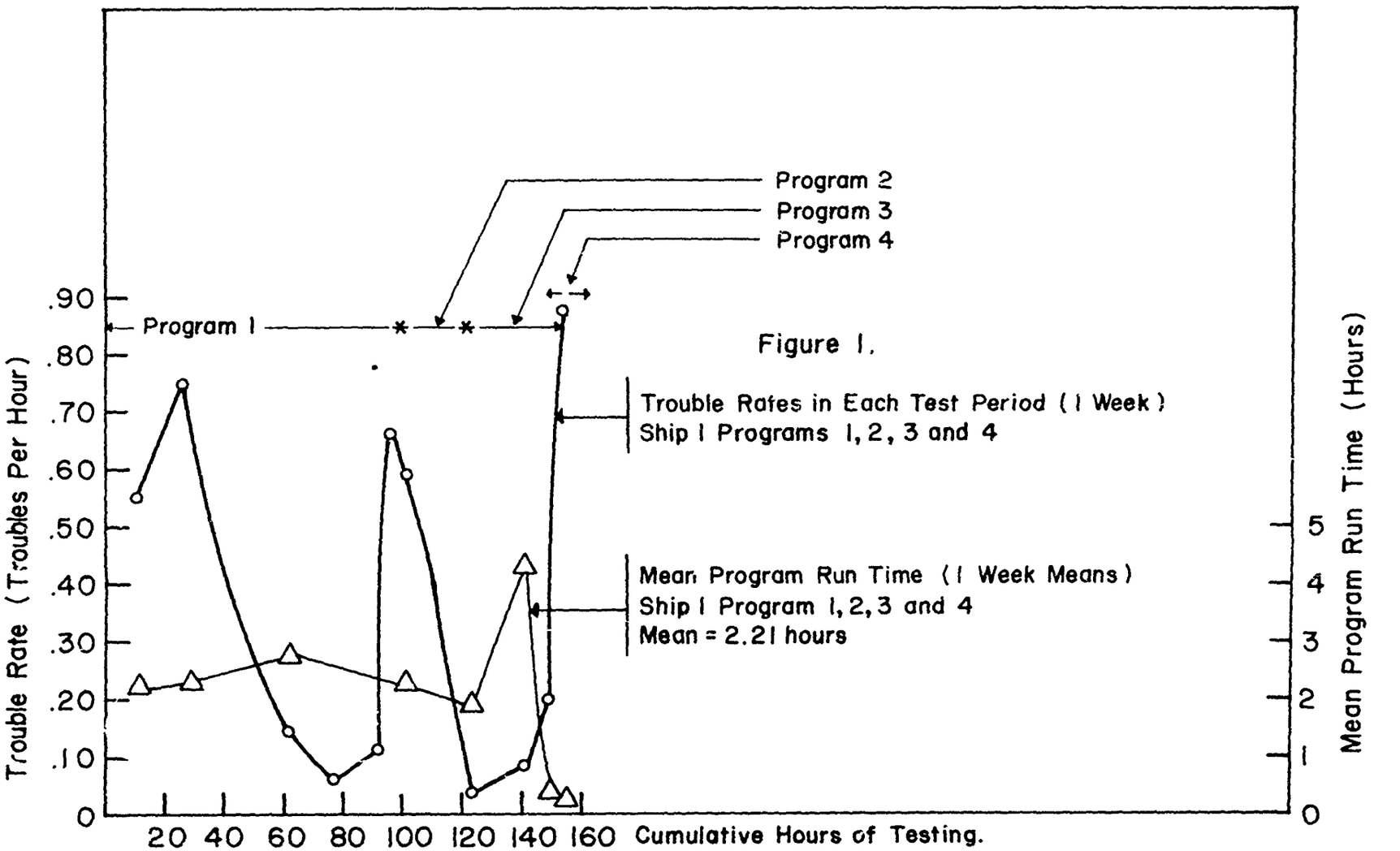

Program 2
Program 3
Program 4
Program 1
.90
.80
.70
.60
.50
.40
.30
.20
.10
0
Trouble Rate (Troubles Per Hour)
Trouble Rates in Each Test Period (1 Week)
Ship 1 Programs 1, 2, 3 and 4
Mean Program Run Time (1 Week Means)
Ship 1 Program 1, 2, 3 and 4
Mean = 2.21 hours
5
4
3
2
1
0
Mean Program Run Time (Hours)
20 40 60 80 100 120 140 160 Cumulative Hours of Testing.

Figure 1.

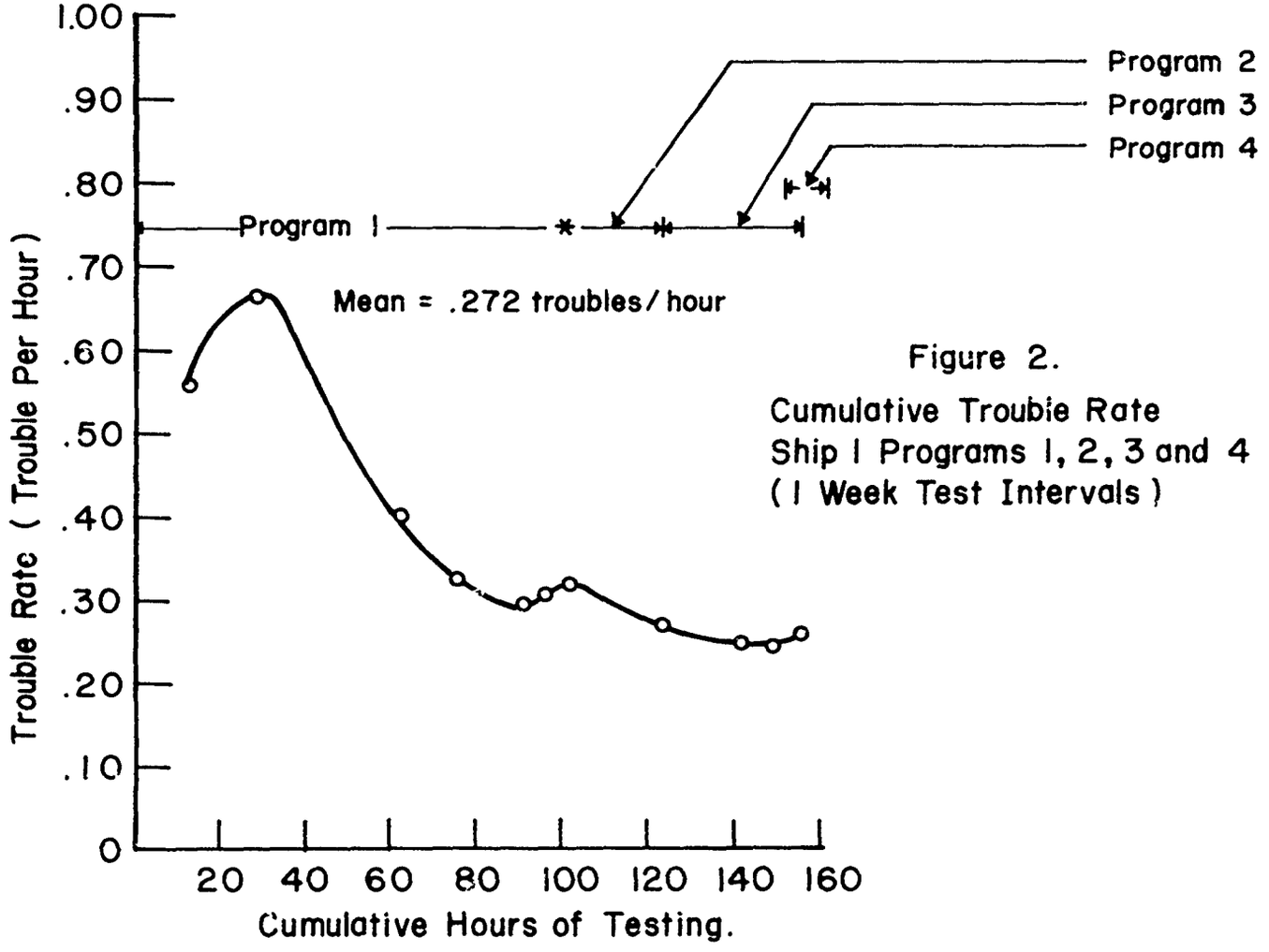


Figure 2.
Cumulative Trouble Rate
Ship I Programs 1, 2, 3 and 4
(I Week Test Intervals)

the weekly rate; the trouble rate shown in Figure 2 is the cumulative trouble rate. Also plotted in Figure 1 is mean program run time, computed on a weekly basis. These data suggest that trouble rates decrease with increased testing. It appears that fluctuations occur but with decreasing amplitude as testing continues. Subsequent analysis on many program modules have verified this decreasing oscillatory behavior. The data concerning program run time are inconclusive.

In summary, the data presented here and subsequent analysis[1] indicate that trouble rates decrease and stabilize and that time between troubles increase and stabilize with continued testing.

B. Distribution of Time Between Troubles and the Reliability Function

Integration with respect to time, of the probability density function of time between troubles yields the unreliability function from which the reliability function can be obtained. The reliability function is used to predict the reliability of a program for various operating times. In order to estimate the reliability function, it is first necessary to obtain the empirical distribution of time between troubles from a sample of trouble report data. Then, parameter estimates are obtained from the sample data, and a goodness of fit test is made in an attempt to identify an appropriate theoretical reliability function. In addition to its use as a reliability predictor, the reliability function can be employed to estimate additional testing requirements

[1]Based on analyses which have been performed subsequent to the period covered by this report.

whenever predicted reliability is less than specified reliability.

Goodness of fit tests were conducted for Program 1, Ship 1 to determine whether typical reliability functions, such as the normal or exponential, would be appropriate for predicting program reliability. The normal test was of interest to ascertain whether software has an increasing hazard rate, i.e., conditional trouble rate increases with operating time. The exponential test was of interest to ascertain whether software exhibits a constant hazard rate.

The results of the Chi Square test for normality are given in Table A-1 of the Appendix. The hypothesis that the distribution is normal is rejected at the .005 level of significance. Thus, the normal reliability function appears inappropriate for this program. A goodness of fit test against the exponential distribution which used the Kolmogorov-Smirnov (K-S) method is shown in Table A-4 of the Appendix. Individual time between trouble values were not available. It was necessary to estimate time between trouble by dividing the number of troubles occurring during a time interval by that interval. Thirty-three Trouble Reports provided 10 time between troubles values. For this small sample, the hypothesis of an exponential distribution was accepted at the .05 level of significance. The theoretical exponential reliability and empirical reliability functions are shown in Figure 3.

The fact that this particular program passed a

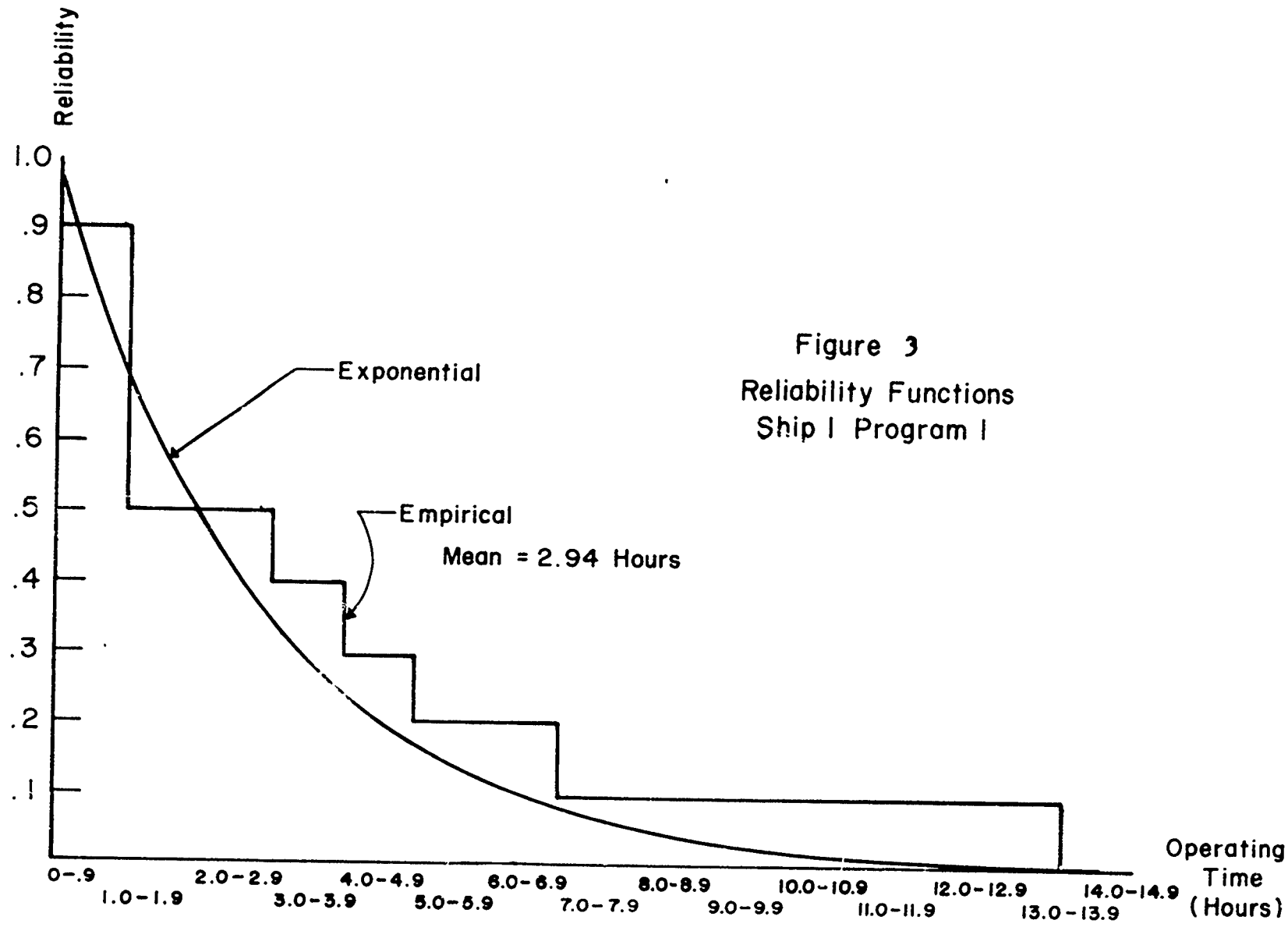


Figure 3
Reliability Functions
Ship I Program I

goodness of fit test for the exponential does not mean that programs, in general, have this distribution. Subsequent analysis indicates that over a sufficiently long operating period, the distribution of time between troubles and the distribution of number of troubles occurring in a given time interval are not stationary processes. The mean time between troubles decreases significantly with increased program testing, although the form of the distribution in later test periods may be the same as in earlier periods.

Another random variable which may be used to provide a limited form of reliability prediction is program run time. The complement of the distribution function of program run time is the conditional probability of a program operating successfully for t hours, given that trouble will occur during the run. This interpretation is used because program run time is the time to failure for programs which fail. Although this probability is not equivalent to reliability, it is a useful measure of reliability because the probability of surviving during the required operating time, for programs which fail, can be estimated. In addition, program run time can be readily obtained for NTDS programs, whereas only approximate values of the time between troubles variable can be obtained by laborious methods.

C. Analysis of Homogeneity of Reliability Distribution Among Programs

A major objective of this research is the determination of whether the various NTDS programs have the same or different distributions. If all or many of the programs

have the same type of distribution, it might be possible to develop a general model of software reliability which would be valid for a large number of programs. Conversely, if there is considerable variety in type of distribution, it would be necessary to identify the type of distribution which is applicable to a program in order to obtain its reliability function.

Since the time between troubles variable was difficult to obtain in large quantities from available data, whereas the program run time variable was available for many programs, the latter was utilized for this analysis. Although the reliability function cannot be derived from program run time, this variable is a measure of program reliability. Hence, an analysis of program run time distributions for various programs provides an indication of whether programs have similar or dissimilar reliability characteristics.

In order to estimate the consistency, if any, of type of distribution among programs, the following program-ship combinations were analyzed:

- Program 1, Ship 1
- Program 1, Ship 2
- Program 1, Ships 1-7
- Program 5, Ship 8
- Ten-Program-Ship Pairs Combination

The ship designation refers to a ship mock-up for on-shore program testing and does not refer to data collected from a ship at sea. The type of goodness of fit test that was made was based on the shape of the program's histogram (exponential

tests were made for programs with an exponentially shaped histogram). For some programs, goodness of fit tests were made against two different distributions, when it was convenient to do so. The results of the tests are listed in Table 1. The figures are located in the Appendix.

These results suggest that there is a lack of homogeneity of type of program run time distribution. Currently, NTDS modules, rather than programs, are undergoing analysis. Since a program consists of several modules, a program may be too large and complex a unit to use for reliability analysis due to the interactions among modules. Modules appear to be more suitable for analysis because each module performs a specific function and module coding has been somewhat standardized. Also, due to differences in software interface requirements among ships, ship operating requirements and computer configurations have an effect on software reliability. Since the various modules are used on many different ships, the effect of ship environment on software reliability would tend to be minimized when reliability is analyzed by module.

D. Analysis of Variance.

A second and more rigorous test for determining whether significant differences in reliability characteristics exist among programs was performed using Analysis of Variance (AOV). This test involves the hypothesis of equality of means among several populations. Program run time was used in this test. A single category test was used. The single category of classification was program/ship combination. It was of

TABLE 1

Program Run Time Goodness of Fit Tests

Appendix Location	Program	Ship	Test	Result
Figure A-1	1	1	Probability Paper - Normality	Appears Non-Normal
Figure A-1	1	2	Probability Paper - Normality	Appears Non-Normal
Figure A-1	1	1-7	Probability Paper - Normality	Appears Non-Normal
Figure A-2	1	2	K-S Goodness of Fit For Exponential	Accept ($\alpha = .05$)
Figure A-3	1	1-7	K-S Goodness of Fit For Exponential	Accept ($\alpha = .05$)
Figure A-4	5	8	K-S Goodness of Fit For Normal	Reject ($\alpha = .05$)
Table A-2	10 program-ship pairs		Chi-Square Goodness of Fit For Normality	Accept ($\alpha = .005$)

interest to learn whether mean program run times differ for various program-ship combinations. It would have also been possible to perform a two category analysis - program and ship; however, the primary interest at this stage of the analysis was to compare program run time means for program-ship combinations. Twenty-eight programs were used in one AOV test. Some departure from the assumptions of an AOV test are present, because program run time is not normally distributed for all 28 programs. There is also some departure from the assumption of equal variances.[1] The results of the test are given in Table A-3 in the Appendix. The hypothesis that all program run time means are equal is rejected at the .05 level of significance.

A second AOV test was conducted using only Program 1 for Ships 1-7. In this case, the test involves equality of program run time means for the same program used on seven different ships. In this case, the category of classification is Ship. Here, there is also some departure from the assumptions of the AOV test. The results of this test are summarized in Table A-5. The hypothesis of equality of program run time means is rejected at the .05 level of significance. Since this test was only conducted for one program, the result does not mean that, in general, the ship environment effect is significant. A two category (program and ship) AOV would provide better information about the effects of program and ship.

Thus, both tests, one involving 28 programs and many

[1] A logarithmic transformation might be appropriate in order to normalize the data for programs which have skewed distributions that are approximately normal. If the transformation did result in normalization, the assumptions of the AOV test would be better fulfilled.

ships and the other involving the same program for seven ships, indicate that the programs are heterogeneous with respect to reliability characteristics.

Although the AOV and the goodness of fit tests (described in the previous section) are not exhaustive, the results suggest that program reliability characteristics are heterogeneous and that program reliability and quality control may have to be dealt with on an individual program basis.

IV. Reliability Prediction

One approach to software reliability prediction is to identify a theoretical reliability function which represents a good fit to the empirical data. This approach would be accomplished by using the following sequence:

- Tentative selection of reliability function based on shape of frequency function of empirical data
- Estimation of reliability function parameters
- Identification of reliability function by using goodness of fit tests
- Estimation of reliability function parameters confidence limit
- Estimation of reliability function confidence limit
- Prediction of reliability and its confidence limit for various intended operating times
- Comparison of required reliability with predicted reliability

The implementation of the above sequence is complicated by the fact that the time between troubles or number of troubles per fixed time interval is not a stationary process with respect to test time. As a result of a reduction in the trouble rate as testing continues, the form of the distribution may remain the same over time but parameter values may change, or the actual form of the distribution may change. This means that a reliability function which is based on the total number of data points collected over the entire test time may not be an accurate predictor, because the data set

is non-representative of the current state of the error occurrence process. If the form of the distribution remains the same throughout the test period and parameters change, indicating an improvement in program quality as testing continues, a smoothing technique could be applied to the most recent data points in order to obtain parameter estimates that would apply to the next time increment. The parameter estimate would be updated as testing continues. If the form of the distribution changes with test time, the problem is much more complex and requires the identification of the distribution which is most appropriate for each stage of testing and operation. Unfortunately, sample size may be drastically reduced when the currency of data points is improved by eliminating out-of-date values.

The following indicates a procedure which would be employed for reliability prediction, once an appropriate reliability function is obtained. The fact that the specifics of this procedure are based on the exponential reliability function does not mean that the exponential distribution can be applied to all programs. In addition, although the specifics of the example are based on the exponential distribution, the general procedure would be applicable to other distributions.

It was shown earlier that for Program 1, Ship 1, an exponential reliability function could be used. Although the fit is not strong, it will be assumed that the exponential applies in order to illustrate the procedure. The calculation

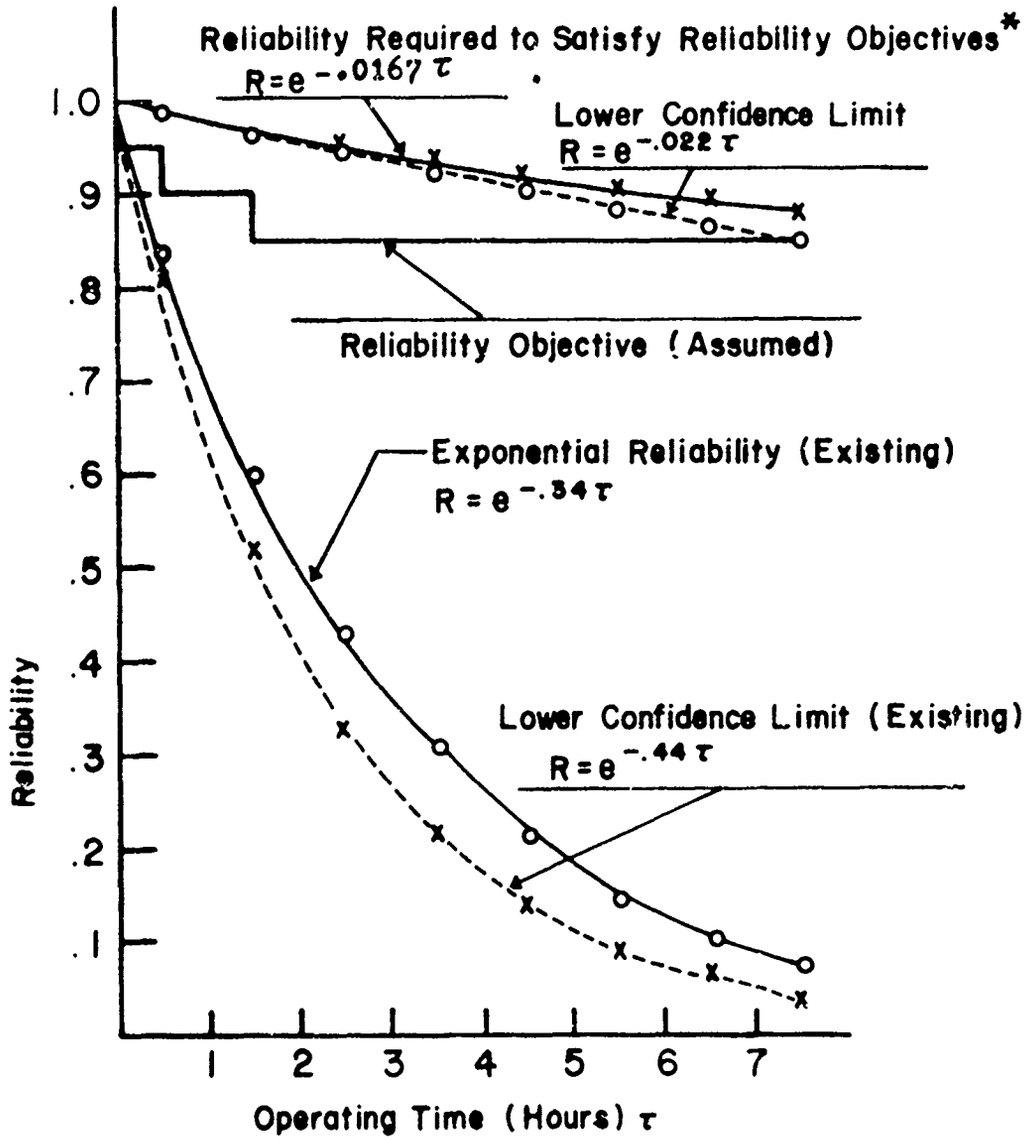
of the lower confidence limit for the MTBT and the reliability function is shown in Section B of the Appendix. The procedure consists of estimating the lower confidence limit of the MTBT and using this value in the exponential reliability function to obtain the reliability lower confidence limit. The exponential reliability function and its 95 per cent lower confidence limit are shown in Figure 4. The sample MTBT which was obtained is 2.94 hours; the 95 per cent lower confidence limit of MTBT is 2.27 hours. Exponential reliability is therefore $R = e^{-.34t}$ and the lower limit is $R = e^{-.44t}$.

The reliability function performs two functions: (1) it is the means of reliability prediction and (2) the lower confidence limit can be compared with the required reliability function for determining whether reliability requirements are satisfied. If this is not the case, the required reliability, MTBT, test time and allowable number of troubles can be estimated. Two examples of this procedure, using the assumed reliability objective shown in Figure 4, are given in Section C of the Appendix. Both examples pertain to a situation in which it is necessary to estimate remaining test requirements after testing is under way. One example pertains to incurring zero future troubles and the other pertains to incurring 10 future troubles. Additional test time, MTBT, reliability and lower reliability limit requirements are estimated for the two cases and are summarized in Section C. The reliability function which would

Figure 4

Reliability Function and Its Confidence Limit for Program I, Ship I Using Exponential Reliability Function.

α = .05 Level of Significance



*Assuming Zero Troubles During Remaining Tests.

$R = e^{-.0172\tau}$ For 10 Troubles During Remaining Tests.

be required in order to satisfy the reliability objective is shown in Figure 4. It is seen that the lower limit of this reliability function is greater than or equal to the reliability objective at all points during the operating time of the program. Thus, the original reliability function parameter estimate, pertaining to test results achieved to date, is used in conjunction with the reliability objective to estimate the remaining test performance requirements. A revised reliability function which will satisfy the reliability objective is also estimated. An interesting result of this analysis is that the 10 trouble situation requires more test time but lower MTBT and reliability than the zero trouble situation, for a given lower reliability limit. This is due to the narrower confidence band which is possible with a larger sample size (greater number of troubles). The examples illustrate that the reliability function can also be used for program quality control by providing a means for estimating the test performance which is necessary to satisfy reliability specifications.

In the example, if the program is tested for an additional 1883 hours and no troubles occur, the required reliability is demonstrated. If one trouble occurs before the expiration of 1883 hours, an amount of time in addition to 1883 hours will be required to demonstrate reliability.

The amount of additional test time (1980 hours), corresponding to 10 future troubles, would apply to the situation in which reliability cannot be demonstrated prior to

the occurrence of the tenth trouble. If an additional test time equal to 1980 hours has expired and no more than 10 troubles has occurred, reliability would be demonstrated.

The estimation of future test requirements is an iterative process. At the termination of each test stage, future test requirements are estimated on the basis of test experience to date and required reliability. Test requirements for each stage are specified in terms of paired values of number of troubles and amount of test time. The pair which will apply depends upon the software trouble experienced during the next stage. Once reliability has been demonstrated, testing is discontinued and the predicted reliability function of the final stage become the reliability function for operational use. Updating of the reliability function would be continued during the operational phase as additional data on software troubles is obtained.

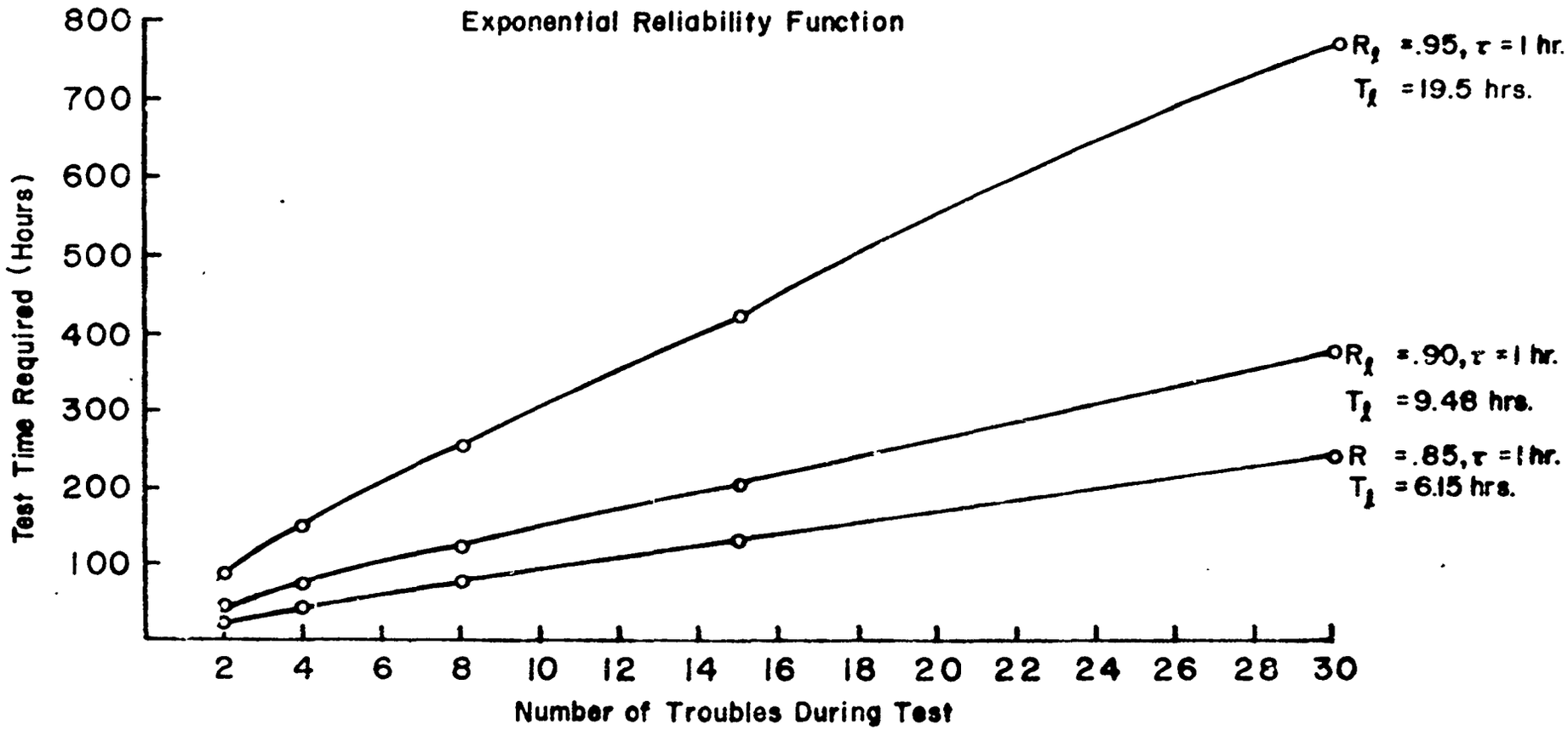
As indicated previously, the reliability function which applies during one test stage may not apply during a subsequent test stage. As testing proceeds and additional troubles occur, the type of reliability distribution or its parameters are revised. The revised function is used to obtain the reliability prediction for the next stage. At the conclusion of each test stage, it is assumed that the revised reliability function, with parameter estimate updated for the next stage, is applicable to the next test stage. Once additional data are obtained from the next stage, estimates of reliability and test requirements ar

revised as necessary. In the example, it was assumed that the exponential distribution was applicable to the next stage. However, the parameter estimate was revised in accordance with assumed values of number of troubles occuring in the next test stage.

In some cases, a change in the type of reliability function is made at the termination of a test stage, if a significant change occurs in the distribution of time between troubles.

Equations for estimating the amount of test time required in order to achieve a reliability objective are formulated in Section D of the Appendix. Required test time is a function of reliability lower limit and program operating time (these constitute the reliability specifications), χ^2 (value of Chi Square distribution) and number of troubles. Required test time as a function of number of troubles is shown in Figure 5 for required lower confidence limits of R(t) of .85, .90 and .95 for one hour of operating time. These curves can be used to estimate the amounts of test time required for achieving specified reliabilities. For a given reliability objective, test time increases approximately linearly with number of troubles. However, test time increases rapidly with increases in reliability objective. For example, if the reliability objective is increased from .90 to .95 for 18 troubles, the reliability requirement increases by 5.6 per cent and required test time increases from 240 to 500 hours, or an increase of 108 per cent. This set of curves is

Figure 5.
Amount of Test Time Required to Achieve
Indicated Lower Limit of Reliability $\alpha = .05$



applicable only to exponential reliability functions.

Required MTBT and reliability versus number of troubles for various values of reliability objective, are shown in Figure 6. The curves in Figure 6 can be used to estimate the MTBT and reliability that are required, for a given number of troubles, in order to satisfy reliability requirements. This set of curves is applicable only to exponential reliability functions.

V. Results and Conclusions.

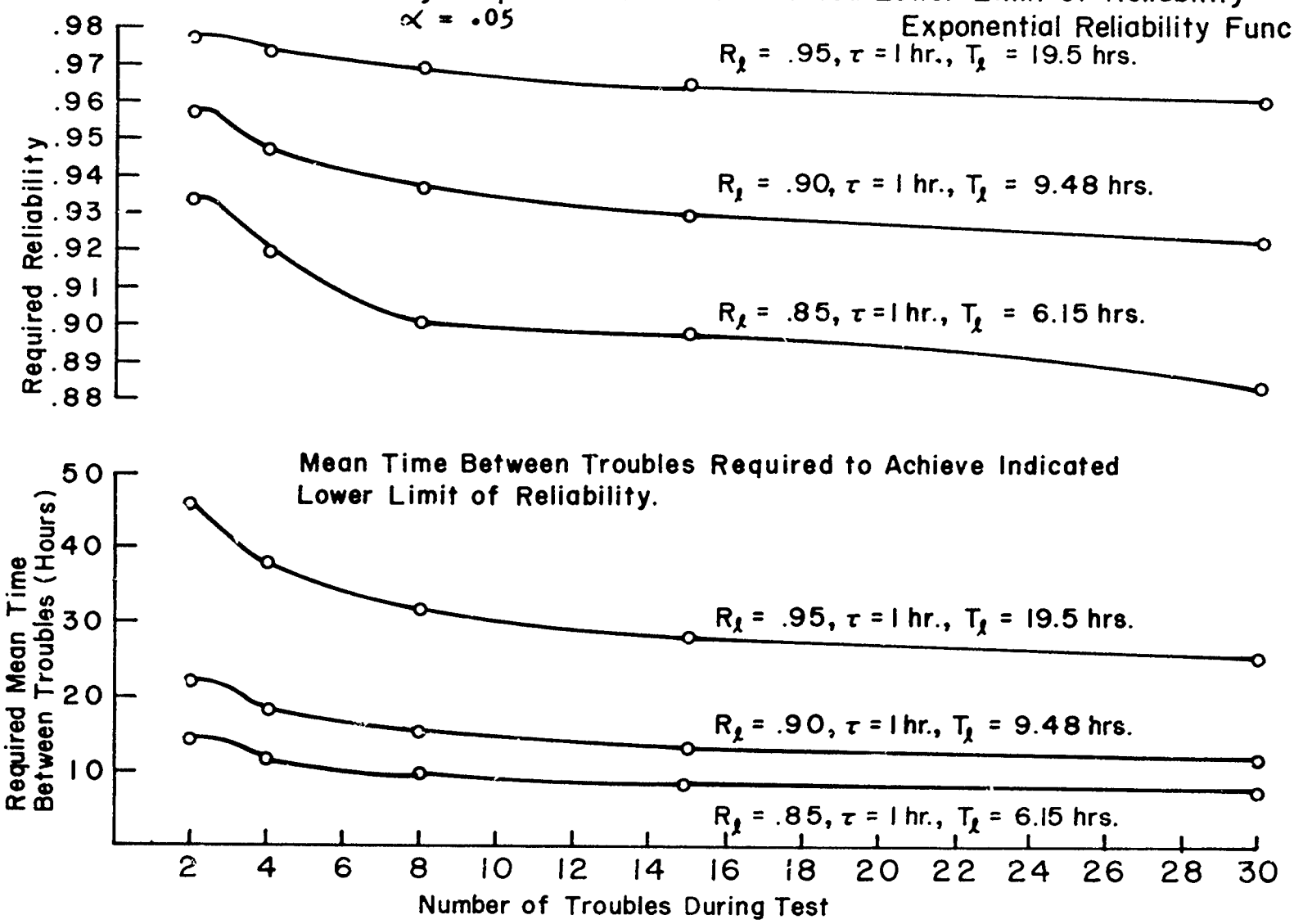
Major results and conclusions of the the first phase of the research are given below.

1. A methodology for software reliability prediction and quality control has been presented which could be implemented in an NTDS software production environment. The value of the methodology is that it provides a framework for software reliability analysis. The specifics of the approach will probably be supplanted by an improved model which is now under development.
2. Methods have been described for estimating the reliability and test performance requirements which are necessary in order to satisfy program reliability objectives.
3. Major factors which affect software reliability prediction accuracy are the heterogeneity of reliability characteristics among programs and the non-stationary nature of the error occurrence process.
4. NTDS programs appear to be heterogeneous with respect to reliability characteristics.

5. Reliability prediction and quality control measures should be applied on an individual program/ship combination basis, due to the significant variability in reliability characteristics among programs.

Figure 6.

Reliability Required to Achieve Indicated Lower Limit of Reliability



VI. Appendix

TABLE A-1

Chi Square Test for Normality

Ship: 1 Program: 1

Time Between Troubles Distribution

f_0	f_t	$f_0 - f_t$	$(f_0-f_t)^2 \times 10^{-4}$	$(f_0-f_t)^2/f_t \times 10^{-4}$
.10	.070	+.030	9.0	128
.40	.168	+.232	538.0	3,200
0	.266	-.266	707.0	2,650
.10	.259	-.159	253.0	975
.10	.161	-.061	37.2	231
.10	.063	+.037	13.7	218
0	.014	+.186	236.0	16,850
.10	0	-	-	-
.10	0	-	-	-
				24,252

$$\sum (N)\frac{(f_0-f_t)^2}{f_t} = (10)(2.43) = 24.3$$

n = 7, df = 4, $\chi^2_{4,99.5} = 14.9$ Reject Normality

Legend

f_0 observed frequency

f_t theoretical frequency (normal distribution)

df degrees of freedom

χ^2 Chi square

n = number of frequency groups

N = number of data points

TABLE A-2

Chi Square Test for Normality for 10 Program-Ship Pairs

for Program Run Time Distribution

f_0	f_t	$f_0 - f_t$	$(f_0-f_t)^2 \times 10^{-4}$	$(f_0-f_t)^2/f_t \times 10^{-4}$
.172	.138	+.034	11.55	83.8
.274	.240	+.034	11.55	48.1
.222	.274	-.052	27.00	98.6
.197	.205	-.008	.64	3.1
.089	.101	-.012	1.44	14.3
.015	.033	+.003	.09	2.1
.012	.007	-	-	-
.015	.002	-	-	-
.003	0	-	-	-
				250.0

$$\sum (N) \frac{(f_0-f_t)^2}{f_t} = (325)(.0250) = 8.13$$

$n = 6$, $df = 3$, $\chi^2_{3,99.5} = 12.84$ Accept Normality

TABLE A-3

Analysis of Variance Results for Program Run Time

28 Programs

	Sum of Squares	df	Mean Square ss/df
Between Programs	123	27	4.56
Within Programs	558	268	2.08
	681		

$F = 4.56/2.08 = 2.19$ Reject hypothesis of equal means.

$F_{.95}(24^*,120^{**}) = 1.61$

$F_{.95}(24^*,\infty) = 1.52$

---

* Nearest table value.

** Highest table value before infinity.

Legend

df degrees of freedom

ss sum of squares

F value of F distribution

TABLE A-4

K-S Test for Exponential

Ship 1 Program 1

Time Between Troubles Distribution

S(t)	F(t)	D(t)
.10	.16	.06
.50	.40	.10
.50	.57	.07
.60	.69	.09
.70	.78	.08
.80	.85	.05
.80	.89	.09
.90	.92	.02
1.00	.99	.01

N = 10 values of time between troubles, involving 33 Trouble Reports

$D(t)_{max}$ = .10

$D_{10,.05}$ = .409

Accept Exponential

Legend

S(t) Sample CDF

F(t) Theoretical CDF

$D(t) = |S(t) - F(t)|$

TABLE A-5

Analysis of Variance Results for Program Run Time

for Program 1, Various Ships[1]

	Sum of Squares	df	Mean Square ss/df
Between Ships	141	8**	17.6
Within Ships	129	86	1.50
	270		

$F = 17.6/1.50 = 11.7$ Reject hypothesis of equal means.

$F_{.95}(8,60^*) = 2.10$

---

[1] Ships: 1,2,3,4,5,6, and 7. (2 of these ships had 2 versions of program 1, thus constituting 9 versions of Program 1 in total)**

*Closest table value.

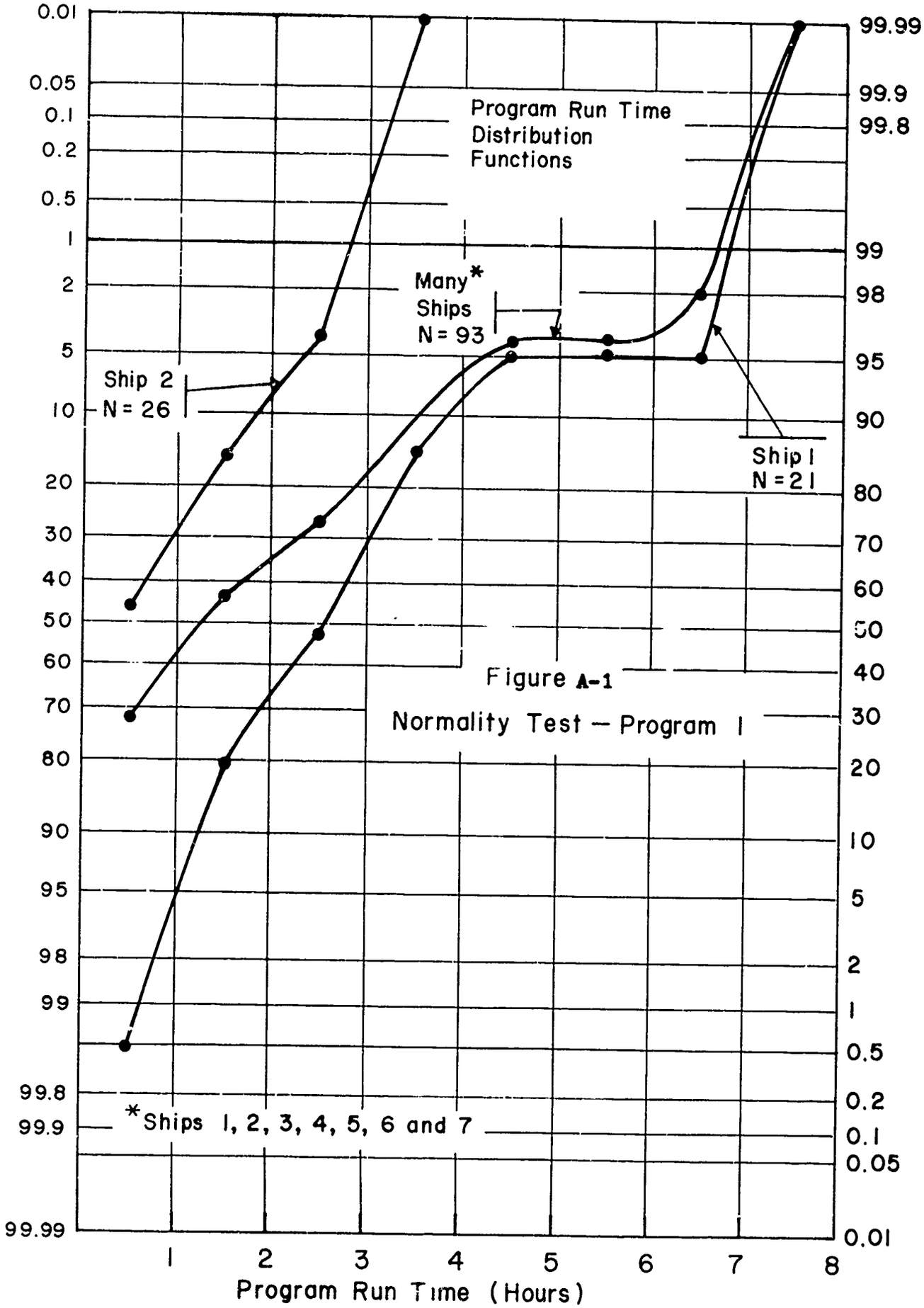


Figure A-1

Normality Test — Program 1

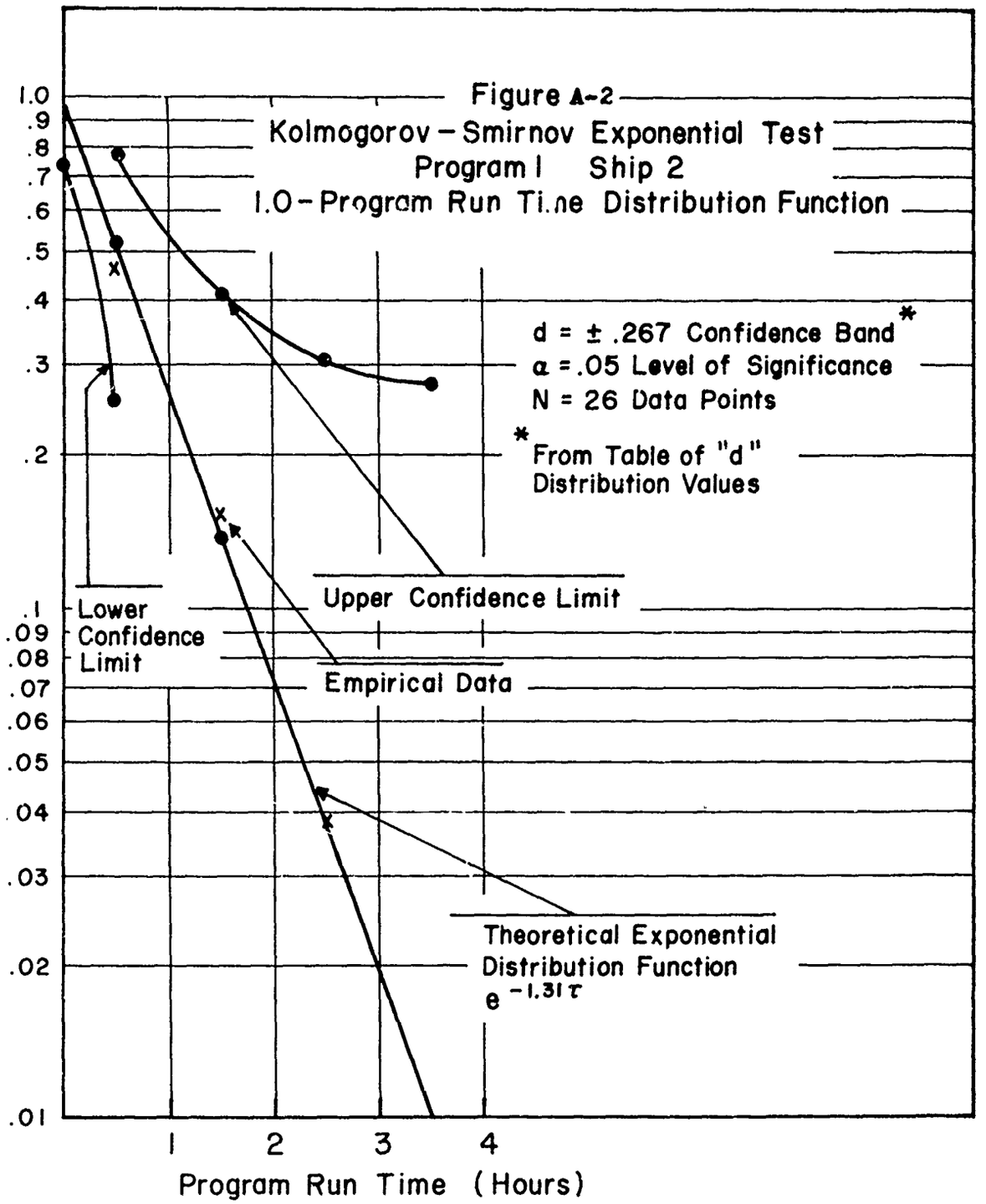


Figure A-2
Kolmogorov – Smirnov Exponential Test
Program I Ship 2
1.0 – Program Run Time Distribution Function
d = ± .267 Confidence Band*
α = .05 Level of Significance
N = 26 Data Points
* From Table of "d" Distribution Values
Upper Confidence Limit
Lower Confidence Limit
Empirical Data
Theoretical Exponential Distribution Function
e -1.31τ
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
.09
.08
.07
.06
.05
.04
.03
.02
.01
1
2
3
4
Program Run Time (Hours)

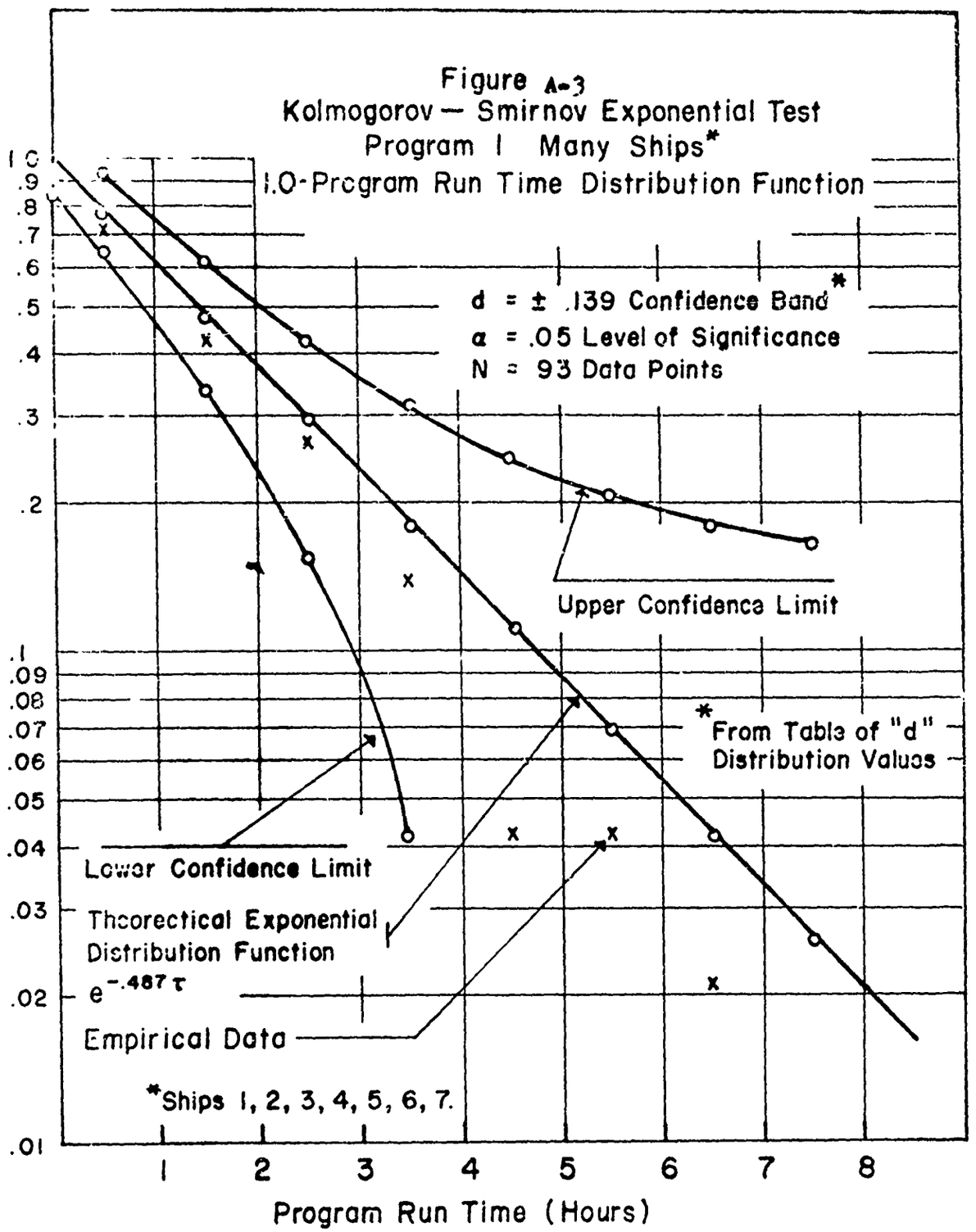


Figure A-3
Kolmogorov — Smirnov Exponential Test
Program I Many Ships*
1.0-Program Run Time Distribution Function
d = ± .139 Confidence Band*
α = .05 Level of Significance
N = 93 Data Points
Upper Confidence Limit
*From Table of "d" Distribution Values
Lower Confidence Limit
Theorectical Exponential Distribution Function
e^-.487τ
Empirical Data
*Ships 1, 2, 3, 4, 5, 6, 7.
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
.09
.08
.07
.06
.05
.04
.03
.02
.01
1
2
3
4
5
6
7
8
Program Run Time (Hours)

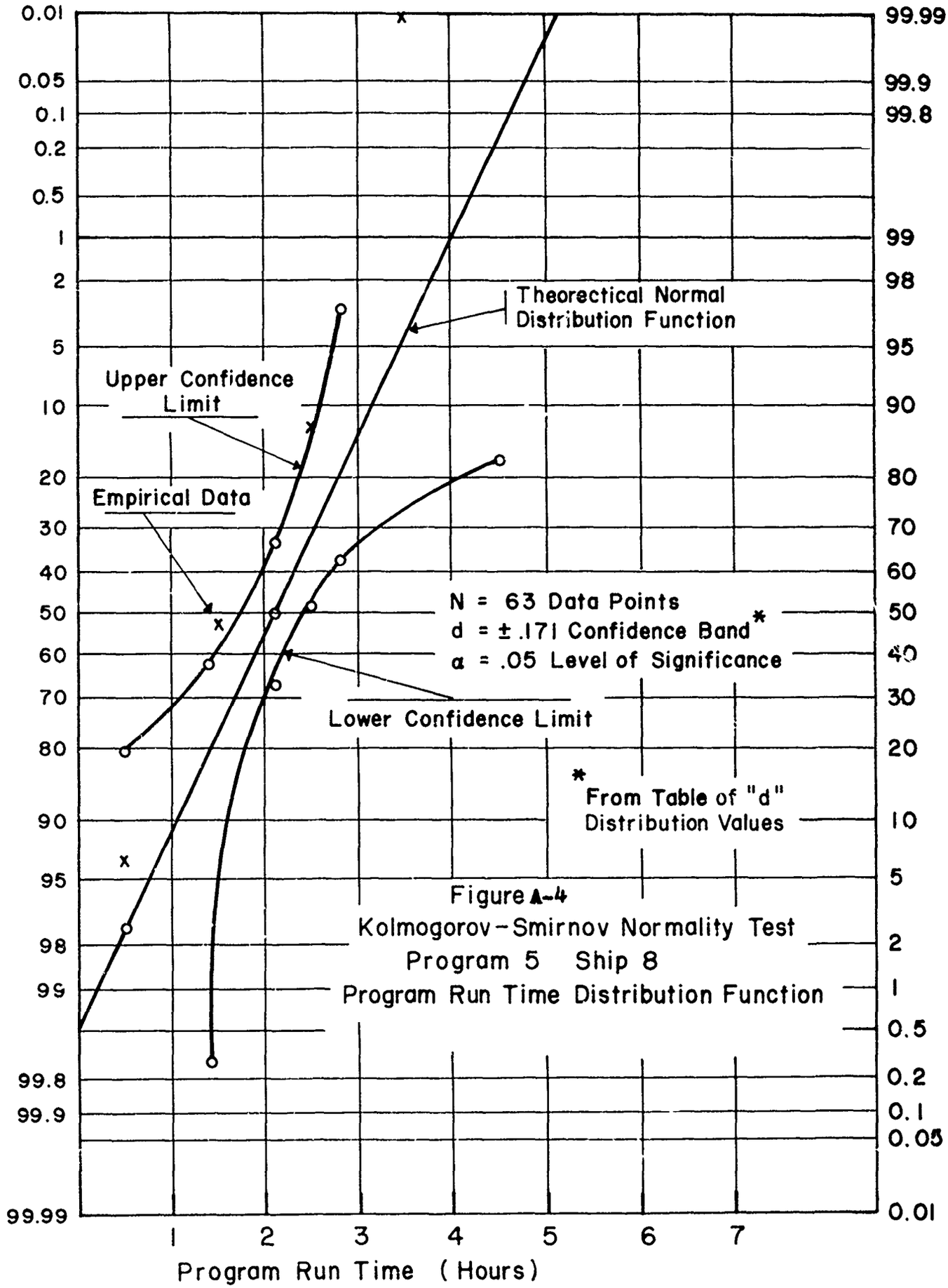


Figure A-4
Kolmogorov-Smirnov Normality Test
Program 5 Ship 8
Program Run Time Distribution Function

B. Exponential Reliability Function - Calculation of Lower Confidence Limits

$$R_\ell = \exp\left(-\frac{t\chi^2_{2n,1-\alpha}}{2n\bar{t}}\right)^{*}, \text{ where} \qquad (1)$$

R_ℓ is lower reliability confidence limit

t is operating time

n is number of troubles

α is level of significance

χ^2 is Chi Square distribution**

$\bar{t}$ is mean time between troubles (MTBT)

Program 1, Ship 1

$n = 33, \quad 2n = 66$

$\alpha = .05$

$\bar{t} = 2.94$ hours

$\chi^2_{66,.95} = 86$

$$\left(\frac{t\chi^2_{2n,1-\alpha}}{2n\bar{t}}\right) = \frac{86t}{(2)(33)(2.94)} = \underline{.44\ t}$$

$R_\ell = \underline{e^{-.44t}}$ = lower confidence limit

Lower confidence limit of MTBT =

$$T_\ell = 2n\bar{t}/\chi^2_{2n,1-\alpha} \qquad (2)$$

Lower confidence limit of time between troubles = 1/.44 = 2.27 hours

---

* Lipow, Myron and Lloyd, David K., "Reliability: Management, Methods, and Mathematics," Prentice Hall, 1962, pp. 194-195.

** $\chi^2 = n\left(1 - \frac{2}{9n} + z_{1-\alpha}\sqrt{\frac{2}{9n}}\right)^3$, where $z_{1-\alpha}$ is normal deviate.

C Example of Determining Required Reliability, MTBT and Test Performance

Assume desired reliability of Program 1 for Ship 1 is given as follows.

(1) .95 for first .5 hour of operation

(2) .90 for next 1.0 hour of operation

(3) .85 for next 6.0 hours of operation

Use exponential distribution for reliability function. (It has been previously determined that Program 1, Ship 1 can be represented by an exponential reliability function.)

Lower limit on MTBT for exponential distribution =

$$T_\ell = \frac{2n\bar{t}}{\chi^2_{2n,1-\alpha}}$$

Lower limit on exponential reliability =

$$R_\ell = \exp(-t/T_\ell)$$

(1) For $R_\ell = .95$ and $t = .5$ hours, $.95 = \exp(-.5/T_\ell)$
$\log .95 = -.5/T_\ell$, $-.0513 = -.5/T_\ell$
$T_\ell = 9.73$ hours.

(2) For $R_\ell = .90$ and $t = 1.5$ hours, $.90 = \exp(-1.5/T_\ell)$
$\log .90 = -1.5/T_\ell$, $-.1054 = -1.5/T_\ell$
$T_\ell = 14.2$ hours.

(3) For $R_\ell = .85$ and $t = 7.5$ hours, $.85 = \exp(-7.5/T_\ell)$
$\log .85 = -7.5/T_\ell$, $-.1625 = -7.5/T_\ell$, $T_\ell =$ 46.1 hours.

Required MTBT =

$$\bar{t}_r = T_\ell \, \chi^2_{2(n+r),1-\alpha} / 2(n+r), \quad \text{where} \tag{3}$$

r is number of future troubles occurring during remaining test time

$$\text{required MTBT} = \frac{\text{Test Time to Date + Future Test Time}}{\text{Number of troubles to Date + Number of future troubles}}$$

$$\bar{t}_r = \frac{n \bar{t}_e + t_r}{n + r} = \frac{T_\ell \, \chi^2_{2(n+r),1-\alpha}}{2(n+r)}, \quad \text{where} \tag{4}$$

$\bar{t}_e$ is mean time between troubles to date

T_r is required additional test time

$$T_r = \frac{T_\ell \, \chi^2_{2(n+r),1-\alpha}}{2} - n \bar{t}_e \tag{5}$$

1. Requirements If Zero Troubles Arise During Future Testing

$$\bar{t}_r = \frac{T_\ell \, \chi^2_{66,.95}}{66} = \frac{(46.1)(86)}{66} = \underline{60.0 \text{ hours}}$$

Required additional test time if no more troubles occur:

$$T_r = \frac{(46.1)(86)}{2} - (33)(2.94) = 1980 - 97 = \underline{1883 \text{ hours}}$$

Check: $\text{MTBT} = \frac{97 + 1883}{33 + 0} = \underline{60.0 \text{ hours}}$

Required reliability = $R = e^{-t/60} = e^{-.0167t}$

Lower reliability limit = $R_\ell = e^{-t/46.1} = e^{-.022t}$

2. Requirements If 10 Troubles Arise During Future Testing

Now in example, use $r = 10$. Required MTBT =

$$\bar{t}_r = \frac{T_\ell}{86} \frac{\chi^2_{86,.95}}{86} = \frac{(46.1)(108.6)}{86} = 58.2 \text{ hours}$$

Required additional test time if 10 more troubles occur:

$$T_r = \frac{(46.1)(108.6)}{2} - 97 = 2406 \text{ hours}$$

Check: $\text{MTBT} = \frac{97 + 2406}{33 + 10} = 58.2$ hours

Required reliability = $R = e^{-t/58.2} = e^{-.0172t}$

Lower reliability limit $R_\ell = e^{-.022t}$

The foregoing calculations are summarized below.

3. Summary

		Requirement for Satisfying Reliability Objectives	
	Existing	Zero Future Troubles	Ten Future Troubles
MTBT	2.94 hrs	60.0	58.2
MTBT Lower Limit	2.27 hrs	46.1	46.1
Required Reliability	$e^{-.34t}$	$e^{-.0167t}$	$e^{-.0172t}$
Reliability Lower Limit	$e^{-.44t}$	$e^{-.022t}$	$e^{-.022t}$
Additional Test Time	---	1883 hrs	2406 hrs
Total Test Time	97 hrs	1980 hrs	2503 hrs

D. Expression for Estimating Required Test Time
Exponential Reliability Function

From (3)
$$\bar{t}_r = T_\ell \, \chi^2_{2n,1-\alpha}/2n \tag{6}$$

$$R_\ell = e^{-t/T_\ell}, \quad T_\ell = \frac{t}{\log \frac{1}{R_\ell}} \tag{7}$$

Using (6) and (7)

$$\bar{t}_r = \left(\frac{t}{\log \frac{1}{R_\ell}}\right)\left(\frac{\chi^2_{2n,1-\alpha}}{2n}\right) \tag{8}$$

Also,

$$\bar{t}_r = \frac{T}{n}, \text{ where} \tag{9}$$

T is test time and n is number of troubles .

Equating (8) and (9),

$$T = \frac{t \, \chi^2_{2n,1-\alpha}}{2 \log \frac{1}{R_\ell}} \tag{10}$$

This gives required test time T in terms of lower confidence limit reliability R_ℓ, operating time t, number of troubles n, Chi Square distribution $\chi^2_{2n,1-\alpha}$ and level of significance α.