



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Reports and Technical Reports

All Technical Reports Collection

---

1997-05-12

**MCTSSA Software Reliability Handbook,  
Volume IV: Schneidewind Software Reliability  
and Metrics Model Tool List**

Schneidewind, Norman F.

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/15416>

---

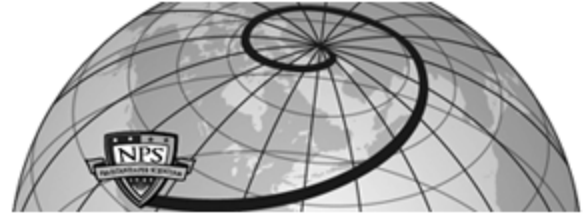
*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>



Author(s)	Schneidewind, Norman F.
Title	MCTSSA software reliability handbook. Volume IV
Publisher	Monterey, California. Naval Postgraduate School
Issue Date	1997-05-12
URL	<a href="http://hdl.handle.net/10945/15416">http://hdl.handle.net/10945/15416</a>

This document was downloaded on March 15, 2013 at 10:39:06



<http://www.nps.edu/library>

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School  
411 Dyer Road / 1 University Circle  
Monterey, California USA 93943**



<http://www.nps.edu/>

# NAVAL POSTGRADUATE SCHOOL Monterey, California



## MCTSSA Software Reliability Handbook

### Volume IV

## Schneidewind Software Reliability and Metrics Model Tool List

by

Norman F. Schneidewind

12 May 1997

Approved for public release; distribution is unlimited.

Prepared for: U.S. ~~Marine~~ Corps  
Tactical Systems Support Activity  
Camp Pendleton, CA 92244-5171

19970918 133

**NAVAL POSTGRADUATE SCHOOL  
Monterey, California**

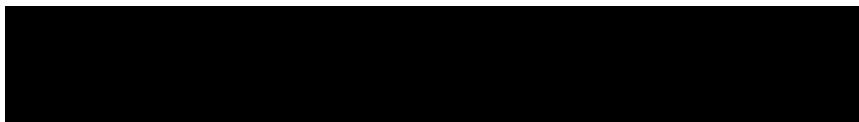
RADM M.J. Evans  
Superintendent

Richard Elster  
Provost

This report was prepared for and funded by the U.S. Marine Corps, Systems Support Activity, Camp Pendleton, CA 92255-5171.

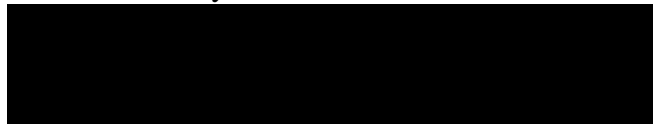
Reproduction of all or part of this report is authorized.

This report was prepared by:



. Norman F. Schneidewind  
Department of Systems Management

Reviewed by:



Reuben T. Harris, Chairman  
Systems Management Department

Released by:



D.W. Netzer, Associate Provost and  
Dean of Research

**REPORT DOCUMENTATION PAGE**

Form Approved

OMB No 0704-0188

Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

MCTSSA Software Reliability Handbook

Volume IV

Schneidewind Software Reliability and Metrics Model Tool List

5. FUNDING

RLACH

6. AUTHOR(S)

Dr. Norman F. Schneidewind

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Systems Management

Naval Postgraduate School

Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION  
REPORT NUMBER

NPS-SM-97-005

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Marine Corps Tactical Systems Support Activity

Box 555171 Building 31345

Camp Pendleton, CA 92255-5171

10. SPONSORING/MONITORING  
AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

The purpose of this handbook is threefold. Specifically, it:

- o Serves as a reference guide for implementing standard software reliability practices at Marine Corps Tactical Systems Support Activity and aids in applying the software reliability model
- o Serves as a tool for managing the software reliability program
- o Serves as a training aid

This handbook consists of four volumes. The content of each of the volumes is as follows:

Volume I: Software Reliability Engineering Process and Modeling for a Single Function System

Volume II: Data Collection Demonstration and Software Reliability Modeling for a Multi-Function Distributed System

Volume III: Integration of Software Metrics with Quality and Reliability

Volume IV: Schneidewind Software Reliability and Metrics Models Tool List

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION  
OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION  
OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION  
OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF  
ABSTRACT

SAR

# **MCTSSA SOFTWARE RELIABILITY HANDBOOK**

## **VOLUME IV**

### **SCHNEIDEWIND SOFTWARE RELIABILITY AND METRICS MODELS TOOL LIST**

12 May 1997

Revised: 15 July 1997

Dr. Norman F. Schneidewind

Code SM/Ss  
Naval Postgraduate School  
Monterey, CA 93943

Voice: (408) 656-2719

Fax : (408) 656-3407

Email: [schneidewind@nps.navy.mil](mailto:schneidewind@nps.navy.mil)

**DTIC QUALITY INSPECTED 3**

## TABLE OF CONTENTS

INTRODUCTION .....	4
SOFTWARE RELIABILITY MODEL EQUATIONS	
.....	-
NOTATION .....	5
EQUATIONS IMPLEMENTED IN SMERFS	
.....	6
Parameter Estimation	
.....	6
Method 1 .....	6
Method 2 .....	7
Method 3 .....	7
Failures in an Interval Range	
.....	7
Maximum Failures .....	7
Remaining Failures .....	8
Time to Next Failure .....	8
Mean Square Error Criterion for Remaining Failures, Maximum Failures, and Total Test Time	
.....	8
Mean Square Error Criterion for Time to Next Failure(s) .....	8
EQUATIONS IMPLEMENTED IN STATGRAPHICS .....	9
Cumulative Failures .....	9
Remaining Failures .....	9
Fraction of Remaining Failures: .....	9
Operational Quality .....	9
Total Test Time to Achieve Specified Remaining Failures	
.....	9
Time to Next N Failures and Remaining Failures Tradeoffs .....	9
Mean Square Error Criterion for Failure Counts .....	9
Criteria for Safety .....	10
<b>Risk</b> Assessment .....	10

DISTRIBUTED SYSTEM MODEL EQUATIONS

.....	13
NOTATION .....	13
System Nodes .....	13
Node Failure Probabilities .....	13
Node Failure Count .....	13
Types of Software Defects .....	13
System Failure Probability Components .....	13
System Failure Probability .....	14
EQUATIONS IMPLEMENTED USING DEFECT CONTROL SYSTEM DATABASE .....	14
Node Failure Count .....	14
EQUATIONS IMPLEMENTED USING WINDOWS CALCULATOR .....	14
Node Failure Probabilities .....	14
System Failure Probability Components .....	14
EQUATION IMPLEMENTED IN STATGRAPHICS .....	15
System Failure Probability .....	15
EQUATION IMPLEMENTED IN SMERFS .....	16
Time to Failure Prediction .....	16
METRICS MODEL SEQUATIONS .....	17
DISCRIMINATIVE POWER VALIDATION MODEL	
.....	17
NOTATION .....	17
EQUATIONS IMPLEMENTED IN STATGRAPHICS	
.....	17
EQUATION IMPLEMENTED USING WINDOWS CALCULATOR	
.....	18
PREDICTABILITY VALIDATION MODEL	
.....	21
NOTATION .....	21
EQUATIONS IMPLEMENTED IN STATGRAPHICS	
.....	21



## INTRODUCTION

The following is a complete listing, as of this date, of *Schneidewind Software Reliability Model* equations and *Schneidewind Software Metrics Model* equations divided into tool implementation categories (i.e., *SMERFS*, *Statgraphics*, *Defect Control System Database*, and *Windows Calculator*). The purpose is to show which equations are implemented in which tool. The list is divided as follows:

### o SOFTWARE RELIABILITY MODEL EQUATIONS

- NOTATION
- EQUATIONS IMPLEMENTED IN SMERF'S
- EQUATIONS IMPLEMENTED IN STATGRAPHICS
- \* TABLE 1

### o DISTRIBUTED SYSTEM MODEL EQUATIONS

- NOTATION
- EQUATIONS IMPLEMENTED USING DEFECT CONTROL SYSTEM DATABASE
- EQUATIONS IMPLEMENTED USING WINDOWS CALCULATOR
- EQUATIONS IMPLEMENTED IN STATGRAPHICS
- \* TABLE 2
- EQUATION IMPLEMENTED IN SMERF'S

### o METRICS MODELS EQUATIONS

- DISCRIMINATIVE POWER VALIDATION MODEL
- \* NOTATION
- \* EQUATIONS IMPLEMENTED IN STATGRAPHICS
- \* EQUATION IMPLEMENTED USING WINDOWS CALCULATOR
- \*\* TABLE 3
- PREDICTABILITY VALIDATION MODEL
- \* NOTATION
- \* EQUATIONS IMPLEMENTED IN STATGRAPHICS
- \*\*\* TABLE 4

The reason for TABLES 1...4 is that the syntax of the *STATGRAPHICS* equation editor does not correspond identically to that in the equation notation (e.g., no Greek symbols, subscripts, and superscripts available). Also the limited space available for a *STATGRAPHICS* equation definition does not always allow these definitions to be identical to the mathematical definitions. Thus in order to use the *STATGRAPHICS* package, it is necessary to see the equations as they are written, using its syntax. The tables define the syntax.

## SOFTWARE RELIABILITY MODEL EQUATIONS

### NOTATION

$\alpha$	failure rate at the beginning of interval $s$
$\beta$	negative of derivative of failure rate divided by failure rate (i.e., relative failure rate)
$F(i)$	predicted failure count in the range $[1, i]$ ; used in computing MSE,
$F_{ij}$	observed failure count during interval $j$ since interval $i$ ; used in computing $MSE_T$
$F(t)$	predicted failure count in the range $[1, t]$
$F_t$	given number of failures to occur after interval $t$ ; used in predicting $T_F(t)$
$F(t_1, t_2)$	predicted failure count in the range $[t_1, t_2]$
$F(\infty)$	predicted failure count in the range $[1, \infty]$ ; maximum failures over the life of the software
$i$	current interval
$j$	next interval $j > i$ where $F_{ij} > 0$
$J$	maximum $j \leq t$ where $F_{ij} > 0$
$MSE_F$	mean square error criterion for selecting $s$ for failure count predictions
$MSE_r$	mean square error criterion for selecting $s$ for remaining failure predictions
$MSE_T$	mean square error criterion for selecting $s$ for time to next failure predictions
$p(t)$	fraction of remaining failures predicted at time $t$
$Q(t)$	operational quality predicted at time $t$ ; the complement of $p(t)$ ; the degree to which software is free of remaining faults (failures)
$r_c$	critical value of remaining failures; used in computing RCM $r(t_t)$
$r(t)$	remaining failures predicted at time $t$
$r(t_t)$	remaining failures predicted at total test time $t_t$
$\Delta r(T_F, t)$	reduction in remaining failures that would be achieved if the software were executed for a time $T_F$ , predicted at time $t$
RCM $r(t_t)$	risk criterion metric for remaining failures at total test time $t_t$
RCM $T_F(t_t)$	risk criterion metric for time to next failure at total test time $t_t$
$s$	starting interval for using observed failure data in parameter estimation
$s^*$	optimal starting interval for using observed failure data, as determined by MSE criterion
$t$	cumulative time in the range $[1, t]$ ; last interval of observed failure data; current interval
$t_m$	mission duration (end time-start time); used in computing RCM $T_F(t_t)$

$t_t$	total test time (observed or predicted)
$T_F(t)$	time to next failure(s) predicted at time $t$
$T_F(t_t)$	time to next failure predicted at total test time $t_t$
$T_F(\Delta r, t)$	time to next $N$ failures that would be achieved if remaining failures were reduced by $\Delta r$ , predicted at time $t$
$T_{ij}$	time since interval $i$ to observe number of failures $F_{ij}$ during interval $j$ ; used in computing $MSE_T$
$x_k$	number of observed failures in interval $k$
$X_i$	observed failure count in the range $[1, i]$
$X_{s-1}$	observed failure count in the range $[1, s-1]$
$X_{s,i}$	observed failure count in the range $[i, s-1]$
$X_{s,i}$	observed failure count in the range $[s, i]$
$X_{s,t}$	observed failure count in the range $[s, t]$
$X_{s,t_1}$	observed failure count in the range $[s, t_1]$
$X_t$	observed failure count in the range $[1, t]$
$X_{t_1}$	observed failure count in the range $[1, t_1]$

**EOUATIONS IMPLEMENTED IN SMERES**

**Parameter Estimation**

The log of the likelihood function is:

$$\log L = X_t [\log X_t - 1 - \log(1 - \exp(-\beta t))] + X_{s-1} [\log(1 - \exp(-\beta(s-1)))] + X_{s,t} [\log(1 - \exp(-\beta))] - \beta \sum_{k=0}^{t-s} (s+k-1) x_{s,k}$$

This function is used to derive the equations for estimating  $\alpha$  and  $\beta$  for each of the three methods. In the equations that follow,  $\alpha$  and  $\beta$  are *estimates* of the population parameters.

**Method 1**

Use all of the failure counts from interval 1 through  $t$  ( $s=1$ ). This method is used if it is assumed that all of the historical failure counts from 1 through  $t$  are representative of the future failure process. The following two equations are used to estimate  $\beta$  and  $\alpha$ , respectively.

$$\frac{1}{\exp(\beta)-1} - \frac{t}{\exp(\beta t)-1} = \sum_{k=0}^{t-1} k \frac{X_{k,1}}{X_t}$$

$$\alpha = \frac{\beta X_t}{1 - \exp(-\beta t)}$$

## Method 2

Use failure counts only in the intervals  $s$  through  $t$  ( $1 \leq s \leq t$ ). This method is used if it is assumed that only the historical failure counts from  $s$  through  $t$  are representative of the future failure process. The following two equations are used to estimate  $\beta$  and  $\alpha$ , respectively.

$$\frac{1}{\exp(\beta) - 1} \frac{t - s + 1}{\exp(\beta(t - s + 1)) - 1} = \sum_{k=0}^{t-s} k \frac{X_{s+k}}{X_{s,t}}$$

$$\alpha = \frac{\beta X_{s,t}}{1 - \exp(-\beta(t - s + 1))}$$

Method 2 is equivalent to Method 1 for  $s=1$ .

## Method 3

Use the cumulative failure count in the interval 1 through  $s-1$  and individual failure counts in the intervals  $s$  through  $t$  ( $2 \leq s \leq t$ ). This method is used if it is assumed that the historical cumulative failure count from 1 through  $s-1$  and the individual failure counts from  $s$  through  $t$  are representative of the future failure process. This method is intermediate to Method 1, which uses all the data, and Method 2, which discards "old" data. The following two equations are used to estimate  $\beta$  and  $\alpha$ , respectively.

$$\frac{(s-1)X_{s-1}}{\exp(\beta(s-1)) - 1} + \frac{X_{s,t}}{\exp(\beta) - 1} - \frac{tX_t}{\exp(\beta t) - 1} = \sum_{k=0}^{t-s} (s+k-1)X_{s+k}$$

$$\alpha = \frac{\beta X_t}{1 - \exp(-\beta t)}$$

Method 3 is equivalent to Method 1 for  $s=2$ .

## Failures in an Interval Range

Predicted failure count in the range  $[t_1, t_2]$ :

$$F(t_1, t_2) = (\alpha/\beta)[1 - \exp(-\beta((t_2 - s + 1)))] - X_{s,t_1}$$

## Maximum Failures

Predicted failure count in the range  $[1, \infty]$  (i.e., maximum failures over the life of the software):

$$F(\infty) = \alpha/\beta + X_{s-1}$$

(Note: Implemented in SMERFS but the user must make the manual correction of adding  $X_{s-1}$  to the quantity  $\alpha/\beta$  that SMERFS computes).

### **Remaining Failures**

Predicted *remaining failures*  $r(t)$  at time  $t$ :

$$r(t) = (\alpha/\beta) - X_{s,t} = F(\infty) - X_t$$

(Note: Implemented in SMERFS but the user must make the manual correction of adding  $X_s$  to the quantity  $\alpha/\beta - X_t$  that SMERFS computes).

### **Time to Next Failure**

Predicted *time for the next  $F_t$  failures* to occur, when the current time is  $t$ :

$$T_r(t) = [(\log[\alpha/(\alpha - \beta(X_{s,t} + F_t))]/\beta)] - (t - s + 1)$$

$$\text{for } (\alpha/\beta) > (X_{s,t} + F_t)$$

### **Mean Square Error Criterion for Remaining Failures, Maximum Failures, and Total Test Time (For Method 2 and Method 1 (s=1))**

Mean Square Error (MSE<sub>r</sub>) criterion for number of *remaining failures*, etc.:

$$MSE_r = \frac{\sum_{i=s}^t [F(i) - X_i]^2}{t - s + 1}$$

where

$$F(i) = (\alpha/\beta) [1 - \exp(-\beta((i-s+1)))] + X_{s-1}$$

### **Mean Square Error Criterion for Time to Next Failure(s) (For Method 2 and Method 1 (s=1))**

Mean Square Error criterion for *time to next failure(s)*:

$$MSE_T = \frac{\sum_{i=s}^{J-1} [([\log[\alpha/(\alpha - \beta(X_{s,i} + F_{ij}))]/\beta] - (i - s + 1)) - T_{ij}]^2}{(J - s)}$$

$$\text{for } (\alpha/\beta) > (X_{s,i} + F_{ij})$$

## **EQUATIONS IMPLEMENTED IN STATGRAPHICS**

### **Cumulative Failures**

Predicted *failure count in the range [1, t]*:

$$F(t) = (\alpha/\beta)[1 - \exp(-\beta((t-s+1)))] + X_{s-1}$$

### **Remaining Failures**

Predicted *remaining failures* as a function of *total test time*  $t_t$ :

$$r(t_t) = (\alpha/\beta)(\exp(-\beta[t_t - (s-1)]))$$

### **Fraction of Remaining Failures:**

*Fraction of remaining failures* predicted at time  $t$ :

$$p(t) = r(t)/F(\infty)$$

### **Operational Quality**

*Operational quality* predicted at time  $t$ :

$$Q(t) = 1 - p(t)$$

### **Total Test Time to Achieve Specified Remaining Failures**

Predicted *total test time* required to achieve a specified *number of remaining failures* at  $t_t$ ,  $r(t_t)$ :

$$t_t = [\log[\alpha/(\beta[r(t_t)])]]/\beta + (s-1)$$

### **Time to Next N Failures and Remaining Failures Tradeoffs**

*Time to next N failures* that would be achieved if *remaining failures* were reduced by  $\Delta r$ , predicted at time

$$T_F(\Delta r, t) = (-1/\beta)[\log[1 - ((\beta \Delta r/\alpha)(\exp(\beta(t-s+1)))]]$$

for  $((\beta \Delta r/\alpha)(\exp(\beta(t-s+1)))) < 1$ .

*Reduction in remaining failures* that would be achieved if the software were executed for a time  $T_F$  predicted at time  $t$ :

$$\Delta r(T_F, t) = (\alpha/\beta)[\exp(-\beta(t-s+1))][1 - \exp(-\beta(T_F))]$$

### **Mean Square Error Criterion for Failure Counts (For Method 2 and Method 1 (s=1))**

Mean Square Error criterion for *failure counts*:

$$MSE_F = \frac{\sum_{i=s}^t [\alpha/\beta(1-\exp(-\beta(i-s+1))) - X_{s,i}]^2}{t-s+1}$$

### **Criteria for Safety**

1) predicted *remaining failures*  $r(t) < r_c$ ,  
where  $r_c$  is a specified critical value, and

2) predicted *time to next failure*  $T_F(t) > t_m$ ,  
where  $t_m$  is mission duration.

### **Risk Assessment**

**Risk** criterion metric for *remaining failures* at total test time  $t_t$ :

$$RCM\ r(t) = (r(t) - r_c) / r_c = (r(t) / r_c) - 1$$

**Risk** criterion metric for *time to next failure* at total test time  $t_t$ :

$$RCM\ T_F(t) = (t_m - T_F(t)) / t_m = 1 - (T_F(t)) / t_m$$

Note: Although *Criteria for Safety* and *Risk Assessment* equations are not covered in the other volumes of the handbook, they are listed here because they are part of the *Schneidewind Software Reliability Model*. These items are covered in: Norman F. Schneidewind, "Reliability Modeling for Safety Critical Software", IEEE Transactions on Reliability, Vol. 46, No.1, March 1997, pp.88-98.

**TABLE 1: STATGRAPHICS (SGPLUS) EQUATION IMPLEMENTATIONS  
SOFTWARE RELIABILITY MODEL EQUATIONS**

Math Notation	Sgplus Notation	Statgraphics Definition	Sgplus Function
$\alpha$	alpha	Beginning failure rate	From SMERFS
$\beta$	beta	Relative failure rate	From SMERFS
$\Delta r$	deltaR	Delta Remaining Failures	Given value
$f(t)$	d	Predicted Failure Rate	$(\alpha) * (\text{EXP}(-(\beta * (i - (s - 1))))))$
$F(i)$	f	Predicted Cumulative Failures	$((\alpha/\beta) * (1 - \text{EXP}(-(\beta * ((i - s) + 1)))) + Xs$
$F_{ij}$	Fij	Number of failures at j since i in MSETf	From failure data
$F_t$	Fij	Number of failures to occur after interval t in tf	Given value
$F(t)$	Ft	Predicted Maximum Failures	$(\alpha/\beta) + (Xs)$
$i$	$\mathbf{1}$	Execution time index	From failure data
$J$	J	Maximum $j \leq t$ where $F_{ij} > 0$	From failure data
$m(i)$	mi	Predicted failures in intervals	$(\alpha/\beta) * (\text{EXP}(-(\beta * (i - s)))) * (1 - \text{EXP}(-(\beta)))$
$MSE_F$	MSE	MSE: Cumulative Failures	$(\text{SUM}(((\text{EVAL } f) - X_{si})^2)) / ((t - s) + 1)$
$MSE_r$	MSEr	MSE: Remaining Failures	$\text{SUM}(((\text{EVAL } f) - X_t)^2) / ((t - s) + 1)$
$MSE_T$	MSETf	MSE: Time to Failure	$(\text{SUM}(((\text{EVAL } tf) - T_{ij})^2)) / ((J - s))$
$p(t)$	P	Fraction Remaining Failures	$(R_t) / (\text{EVAL } F_t)$
$Q(t)$	Q	Predicted Program Quality	$(1 - (\text{EVAL } p))$
$r_c$	Rc	Remaining Failures Criterion	Given value



$r(t)$	$r$	Predicted remaining failures using $X_t$	$(\alpha/\beta)-(Xst)$
$r(t_i)$	$rt$	Predicted remaining failures, given $\mathfrak{t}$	$(\alpha/\beta)^*(EXP(-\beta*(\mathfrak{t}-(s-1))))$
None	$R$	Predicted remaining failures using $p$	$p^*(EVAL Ft)$
None	$R_{tt}$	Number of remaining failures in computing $p$ and $\mathfrak{t}$	Given value
$\Delta r(T_F, t)$	$dR$	Predicted delta Remaining Failures	$(\alpha/\beta)^*(EXP(-(\beta*(i-(s-1)))))* (1-(EXP(-(\beta*TR))))$
RCM $r(t)$	riskR	Risk of Remaining Failure	$((EVAL rt)-Rc)/Rc$
RCM $T_F(t)$	riskT	Risk of Time to Failure	$(tm-(EVAL tf))/tm$
$s$	$s$	First failure interval	From SMERFS
$t$	$t$	Execution time	From failure data
$t_i$	$tt$	Predicted Total Test Time, given $R_{tt}$	$((LOG(\alpha/(\beta*R_{tt}))/\beta)+(s-1))$
$T_F(t)$	$tf$	Predicted Time to Failure	$((1/\beta)^*(LOG(\alpha/(\alpha-(\beta*(X_{si}+F_{ij})))))) - (i-(s-1))$
$T_F(\Delta r, t)$	$Tf$	Time to Failure for delta Remaining Failures	$(-1/\beta)^*(LOG(1-((\beta/\alpha)^*(\Delta r)^*(EXP(\beta*(i-(s-1)))))))$
$T_{ij}$	$T_{ij}$	Time since $i$ to fail at $j$	From failure data
$t_m$	$tm$	Time to Failure Criterion	Given value
$T_F$	$TR$	Given $Tf$ for Predicted delta Remaining Failures	Given value
$X_{s-1}$	$Xs$	Observed failure count in the range $[1, s-1]$	From failure data
$X_{s,i}$	$X_{si}$	Observed failure count in the range $[s, i]$	From failure data
$X_{s,t}$	$X_{st}$	Observed failure count in the range $[s, t]$	From failure data
$X_t$	$X_t$	Observed failure count in the range $[1, t]$	From failure data

## DISTRIBUTED SYSTEM MODEL EQUATIONS

### NOTATION

#### System Nodes

- $N_{cc}$ : Number of Critical Client nodes  
 $N_{nc}(t)$ : Number of Non-Critical Client nodes  
 $N_{cs}$ : Number of Critical Server nodes  
 $N_{ns}(t)$ : Number of Non-Critical Server nodes  
 $N(t) = N_{cc} + N_{nc}(t) + N_{cs} + N_{ns}(t)$ : Total number of nodes

#### Node Failure Probabilities

- $p_{cc}$ : probability of a software defect causing a critical client node to fail  
 $p_{nc}$ : probability of a software defect causing a non-critical client node to fail  
 $p_{cs}$ : probability of a software defect causing a critical server node to fail  
 $p_{ns}$ : probability of a software defect causing a non-critical server node to fail  
 $p_{sw}$ : probability of a node failure due to software

#### Node Failure Count

- $i$ : identification of an interval of operating time of the software  
 $f_{cc}(i)$ : critical client node failure count in interval  $i$   
 $f_{nc}(i)$ : non-critical client node failure count in interval  $i$   
 $f_{cs}(i)$ : critical server node failure count in interval  $i$   
 $f_{ns}(i)$ : non-critical server node failure count in interval  $i$   
 $d(i)$ : total defect count in interval  $i$   
 $D$ : total defect count across all intervals

#### Types of Software Defects (Examples Only)

- S**: Software Defect  
**G**: General Protection Fault  
**N**: Network Related Defect  
**C**: System Crash

#### System Failure Probability Components

- $t$ : cumulative time in the range  $[1, t]$ ; last interval of observed failure data; current interval  
 $P_{cc}$ : probability that **one or more** critical clients  $N_{cc}$  fail, given that the software fails

$P_{nc}(t)$ : probability that **all** non-critical clients  $N_{nc}(t)$  have failed by time  $t$ , given that the software fails

$p_{cs}$ : probability that **one or more** critical servers  $N_{cs}$ , fail, given that the software fails

$P_{ns}(t)$ : probability that **all** non-critical servers  $N_{ns}(t)$  have failed by time  $t$ , given that the software fails

### **System Failure Probability**

$P_{sys}/\text{node fails}(t)$ : probability of a system failure by time  $t$ , given that **a** node **fails**

### **EQUATIONS IMPLEMENTED USING DEFECT CONTROL SYSTEM DATABASE (Examples Only)**

#### **Node Failure Count**

$f_{cc}(I) = \text{COUNT as failures WHERE } (S \wedge G \wedge N \wedge \text{not} C)$  in interval  $I$

$f_{nc}(I) = \text{COUNT as failures WHERE } (S \wedge G \wedge \text{not} N \wedge \text{not} C)$  in interval  $I$

$f_{cs}(I) = \text{COUNT as failures WHERE } (S \wedge \text{not} G \wedge N \wedge C)$  in interval  $I$

$f_{ns}(I) = \text{COUNT as failures WHERE } (S \wedge \text{not} G \wedge \text{not} N \wedge C)$  in interval  $I$

$d(I) = \text{total defect count in interval } I$

$D = \sum_i d(I)$

### **EQUATIONS IMPLEMENTED USING WINDOWS CALCULATOR**

#### **Node Failure Probabilities**

Probability of a software defect causing a critical client node to fail:

$$p_{cc} = \sum_i f_{cc}(I) / D$$

Probability of a software defect causing a non-critical client node to fail:

$$p_{nc} = \sum_i f_{nc}(I) / D$$

Probability of a software defect causing a critical server node to fail:

$$p_{cs} = \sum_i f_{cs}(I) / D$$

Probability of a software defect causing a non-critical server node to fail:

$$p_{ns} = \sum_i f_{ns}(I) / D$$

Probability of a node failure due to software:

$$P_{sw} = p_{cc} + p_{nc} + p_{cs} + p_{ns}$$

#### **System Failure Probability Components**

Probability that **one or more** critical clients  $N_{cc}$  fail, given that the software fails:

$$P_{cc} = 1 - (1 - p_{cc})^{N_{cc}}$$

Probability that **all** non-critical clients  $N_{nc}(t)$  have failed by time  $t$ , given that the software fails:

$$P_{nc}(t) = (p_{nc})^{N_{nc}(t)}$$

Probability that **one or more** critical servers  $N_{cs}$  fail, given that the software fails:

$$P_{cs} = 1 - (1 - p_{cs})^{N_{cs}}$$

Probability that **all** non-critical servers  $N_{ns}(t)$  have failed by time  $t$ , given that the software fails:

$$P_{ns}(t) = (p_{ns})^{N_{ns}(t)}$$

**EQUATION IMPLEMENTED IN STATGRAPHICS**

**System Failure Probability**

Probability of system failure, by time  $t$ , given a node failure:

$$P_{sys/node\ fails}(t) = [P_{cc}] [P_{nc}(t)] + [P_{cs}] [P_{ns}(t)] =$$

$$[1 - (1 - p_{cc})^{N_{cc}}] [(p_{nc})^{N_{nc}(t)}] + [1 - (1 - p_{cs})^{N_{cs}}] [(p_{ns})^{N_{ns}(t)}]$$

----- .....

Probability of Client Failure    Probability of Server Failure

**TABLE 2: STATGRAPHICS (SGPLUS) EQUATION IMPLEMENTATIONS  
DISTRIBUTED SYSTEM MODEL EQUATIONS**

Math Notation	Sgplus Notation	Statgraphics Definition	Sgplus Function
$N_{cc}$	Ncc	Number of critical clients	From system configuration (constant)
$N_{nc}(t)$			From system configuration (vector as a function of time)
$N_{cs}$	Ncs	Number of critical servers	From system configuration (constant)
$N_{ns}(t)$	Nns	Number of non-critical servers	From system configuration (vector as a function of time)
$p_{cc}$	pcc	Probability of critical client failure	From Windows Calculator
	pnc	Probability of non-critical client failure	From Windows Calculator
$P_{cs}$	pcs	Probability of critical server failure	From Windows Calculator
$p_{ns}(t)$	pns	Probability of non-critical server failure	From Windows Calculator
$P_{sys}/node\ fails(t)$	Psys	Probability System Failure/Node Failure	$((1-(1-pcc)^{Ncc})*((pnc)^{Nnc})) + ((1-(1-pcs)^{Ncs})*((pns)^{Nns}))$

**EQUATION IMPLEMENTED IN SMER**

**Time to Failure Prediction**

Predicted *time for the next  $F_i$  failures* to occur, when the current time is t, for each of the four types of node failures::

$$T_F(t) = [(\log[\alpha / (\alpha - \beta(X_{s,t} + F_i))]) / \beta] - (t - s + 1)$$

for  $(\alpha / \beta) > (X_{s,t} + F_i)$

## METRICS MODELS EQUATIONS

### DISCRIMINATIVE POWER VALIDATION MODEL

#### NOTATION

Defined in Table 3.

#### EQUATIONS IMPLEMENTED IN STATGRAPHICS

Maximum vertical difference between the CDFs of two samples (e.g., the CDFs of  $M_{ij}$  for  $drcount \leq F_c$  and  $drcount > F_c$ ):

$$K-S(M_{c_j}) = \max\{[CDF(M_{ij}/(F_i \leq F_c))] - [CDF(M_{ij}/(F_i > F_c))]\}$$

Module count, based on BDFs of  $F_i$  and  $M_{ij}$ , that are calculated over the  $n$  modules for  $m$  metrics:

$$C_{11} = \text{COUNT}_{i=1}^n \text{ FOR } ((F_i \leq F_c) \wedge (M_{i1} \leq M_{c1}) \wedge (M_{ij} \leq M_{cj}) \wedge (M_{im} \leq M_{cm}))$$

$$C_{12} = \text{COUNT}_{i=1}^n \text{ FOR } ((F_i \leq F_c) \wedge ((M_{i1} > M_{c1}) \vee (M_{ij} > M_{cj}) \vee (M_{im} > M_{cm})))$$

$$C_{21} = \text{COUNT}_{i=1}^n \text{ FOR } ((F_i > F_c) \wedge (M_{i1} \leq M_{c1}) \wedge (M_{ij} \leq M_{cj}) \wedge (M_{im} \leq M_{cm}))$$

$$C_{22} = \text{COUNT}_{i=1}^n \text{ FOR } ((F_i > F_c) \wedge ((M_{i1} > M_{c1}) \vee (M_{ij} > M_{cj}) \vee (M_{im} > M_{cm})))$$

Proportion of Type 1 Misclassifications:

$$P_1 = C_{21}/n$$

Proportion of Type 2 Misclassifications:

$$P_2 = C_{12}/n$$

Proportion of Type 1+Type 2 Misclassifications:

$$P_{12} = (C_{21} + C_{12})/n$$

Proportion of low quality (i.e.,  $drcount > 0$ ) software correctly classified:

$$LQC = C_{22}/n_2$$

Remaining Factor RF (e.g., remaining *drcount*). This is the sum of F, not caught by inspection:

$$RF = \sum_{i=1}^n F_i \text{ FOR } (F_i > F_c) \wedge (M_{i1} \leq M_{c1}) \dots \wedge (M_{ij} \leq M_{cj}) \dots \wedge (M_{im} \leq M_{cm})$$

Proportion of RF, where TF is the total F<sub>i</sub> prior to inspection:

$$RFP = RF/TF$$

$$TF = \sum_{i=1}^n F_i$$

Density of RF:

$$RFD = RF/n$$

Proportion of modules remaining that have  $F_i > F_c$ :

$$RMP = RFM/n,$$

where RFM is given by:

$$RFM = \text{COUNT}_{i=1}^n \text{ FOR } ((F_i > 0) \wedge (M_{i1} \leq M_{c1}) \dots \wedge (M_{ij} \leq M_{cj}) \dots \wedge (M_{im} \leq M_{cm}))$$

Proportion of modules that must be inspected:

$$I = (C_{12} + C_{22})/n$$

Wasted inspection:

$$RI = C_{22}/C_{12}$$

### **EQUATION IMPLEMENTED USING WINDOWS CALCULATOR**

Quality Inspection Ratio:

$$QIR = (\Delta RFP \sqrt{RFP_i}) / (\Delta I / I_i)$$

**TABLE 3: STATGRAPHICS (SGPLUS) AND WINDOWS CALCULATOR EQUATION IMPLEMENTATIONS SOFTWARE METRICS MODELS EQUATIONS**

**DISCRIMINATIVE POWER VALIDATION MODEL**

Math Notation	Sgplus Notation	Statgraphics Definition	Sgplus Function
$C_{11}$	C11	Module count for C11	SUM ((drcount LE Dc) AND (M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (sample EQ Ss))
$C_{12}$	c 12	Module count for C12	SUM ((drcount LE Dc) AND ((MI GT M1c) OR (M2 GT M2c) OR (M3 GT M3c) OR (M4 GT M4c)) AND (sample EQ Ss))
$C_{21}$	c 21	Module count for C21	SUM ((drcount GT Dc) AND (M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (sample EQ Ss))
$C_{22}$	c22	Module count for C22	SUM ((drcount GT Dc) AND ((MI GT M1c) OR (M2 GT M2c) OR (M3 GT M3c) OR (M4 GT M4c)) AND (sample EQ Ss))
$F_c$	Dc	Quality factor critical value	Given value
$F_i$	drcount (example)	Vector of quality factor values	From quality factor data
I	I	Proportion of modules that must be inspected	$((\text{EVAL C12})+(\text{EVAL C22}))/n)* 100 \%$
$\Delta I$	None	Difference in two successive values of I	Windows Calculator computation
$i$	Module name	Module index	From metrics file
$j$	Metric name	Metric index	From metrics file
K-S( $M_{cj}$ )	maxcdfdiff	Maximum vertical difference between two CDFs	MAX (EVAL (cdfdiff)), where cdfdiff= (ABS(m1-m2))/100 & m1, m2=metric vectors
LQC	LQC	Proportion of low quality software correctly classified	$((\text{EVAL C22})/(\text{EVAL n2}))* 100 \%$
$M_{cj}$	M1c...M4c	Vector of j metric critical values	From metrics data and K-S test
$M_{ij}$	M1 ...M4	Matrix of modules and metrics	From metrics data and K-W test
$N_1$	N1	Count of accepted modules	SUM ((M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (sample EQ Ss))



$N_2$	N2	Count of rejected modules	SUM(((M1 GT M1c) OR (M2 GT M2c) OR (M3 GT M3c) OR (M4 GT M4c)) AND (sample EQ Ss))
n	n	Number of modules in sample	Given value
$n_1$	n1	Count of high quality modules	(EVAL C11)+(EVAL C12)
$n_2$	n2	Count of low quality modules	(EVAL C21)+(EVAL C22)
$P_1$	P1	Proportion of Type 1 misclassifications	((EVAL C21)/n)*100 %
$P_2$	P2	Proportion of Type 2 misclassifications	((EVAL C12)/n)*100 %
$P_{12}$	P12	Proportion of Type 1+Type 2 misclassifications	((EVAL C12)+(EVAL C21))/n)* 100 %
QIR	None	Quality Inspection Ratio	Windows Calculator computation
RF	RF	Remaining Quality Factor	SUM (drcount SELECT((M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (drcount GT Dc) AND (sample EQ Ss)))
RFD	RFD	Density of RF	(EVAL RF)/n
RFP	RFP	Proportion of RF	((EVAL RF)/(EVAL TF))* 100 %
ARFP	None	Difference in two successive values of RFP	Windows Calculator computation
RFM	RFM	Count of modules with Remaining Quality Factor	SUM ((drcount GT Dc) AND (M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (sample EQ Ss))
RMP	RMP	Proportion of RFM	((EVAL RFM)/n)* 100 %
RI	RI	Wasted Inspection	(EVAL C22)/(EVAL C12)
None	Ss	Sample Identification	Given value
TF	TF	Total Quality Factor	SUM (drcount SELECT sample EQ Ss)
$\chi_c^2$	$\chi_c^2$	Critical value of Chi-Square	Function of C11, C12, C21, and C22

## PREDICTABILITY VALIDATION MODEL

### NOTATION

Defined in Table 4

### EQUATIONS IMPLEMENTED IN STATGRAPHICS

Proportion of modules with  $F_i > 0$  in the Validation Sample prior to inspection and correction of defects:

$$p_n = (\text{COUNT FOR } F_i > 0) / n$$

Two-sided confidence limits of  $p_n$ , used as predicted limits of  $p_n'$  in the Application Sample:

$$CL_{p_n} = p_n \pm Z_{\alpha/2} \sqrt{\frac{(p_n)(1-p_n)}{n}}$$

Proportion of modules **not flagged** for inspection (i.e., contained in  $N_1$ ) with  $F_i > 0$  in the Validation Sample:

$$pN_1 = \text{RFM} / N_1$$

One-sided upper confidence limit of  $pN_1$ , used as predicted limit of  $pN_1'$  in the Application Sample:

$$UL_{pN_1} = pN_1 + Z_{\alpha} \sqrt{\frac{(pN_1)(1-pN_1)}{N_1}}$$

Proportion of modules **flagged** for inspection (i.e., contained in  $N_2$ ) with  $F_i > 0$  in the Validation Sample:

$$pN_2 = ((p_n)(n) - (\text{RFM})) / N_2$$

One-sided lower confidence limit of  $pN_2$ , used as predicted limit of  $pN_2'$  in the Application Sample:

$$LL_{pN_2} = pN_2 - Z_{\alpha} \sqrt{\frac{(pN_2)(1-pN_2)}{N_2}}$$

Proportion of quality factor that occurs on modules **not flagged** for inspection (i.e., contained in  $N_1$ ) in the Validation Sample:

$$d_1 = \text{RF} / \text{TF} \text{ (same as RFP if FWP is expressed as a proportion)}$$

One-sided upper confidence limit of  $d_1$ , used as predicted limit of  $d_1'$  in the Application Sample

$$UL_{d_1} = d_1 + Z_{\alpha} \sqrt{\frac{(d_1)(1-d_1)}{\text{TF}}}$$

Proportion of quality factor that occurs on modules **flagged** for inspection (i.e., contained in  $N_2$ ) in the Validation Sample:

$$d_2 = 1 - d_1$$

One-sided lower confidence limit of  $d_2$ , used as predicted limit of  $d_2'$  in the Application Sample:

$$LLd_2 = d_2 - Z_{\alpha} \sqrt{\frac{(d_2)(1-d_2)}{TF}}$$

Expected quality factor count (e.g., *drcount*) that occurs on modules **not flagged** for inspection (i.e., contained in  $N_1'$ ) in the Application Sample:

$$D_1 = (RF/N_1)(N_1')$$

Expected quality factor count (e.g., *drcount*) that occurs on modules **flagged** for inspection (i.e., contained in  $N_2'$ ) in the Application Sample):

$$D_2 = ((TF - RF)/N_2)(N_2')$$

**TABLE 4: STATGRAPHICS (SGPLUS) EQUATION IMPLEMENTATIONS  
SOFTWARE METRICS MODELS EQUATIONS**

**PREDICTABILITY VALIDATION MODEL**

Math Notation	Sgplus Notation	Statgraphics Definition	Sgplus Function
$p_n$	pn	Proportion of modules with $F_i > 0$	(SUM((drcount GT 0) AND (sample EQ Ss)))/n
$CL_{p_n}$	CLpn	Two-sided confidence limits of pn	((EVAL pn)+(Z*(SQRT (((EVAL pn)* (1-(EVAL pn)))/n))))
$p_{N_1}$	pN1	Proportion of modules <b>not flagged</b> for inspection	(EVAL RFM)/(EVAL N1)
$UL_{p_{N_1}}$	ULpN1	Upper Confidence limit of pN1	((EVAL pN1)+(Z*(SQRT (((EVAL pN1)* (1-(EVAL pN1)))/EVAL N1))))
$p_{N_2}$	pN2	Proportion of modules <b>flagged</b> for inspection	((n*(EVAL pn))-(EVAL RFM))/(EVAL N2)
$LL_{p_{N_2}}$	LLpN2	Lower confidence limit of pN1	((EVAL pN2)-(Z*(SQRT (((EVAL pN2)* (1-(EVAL pN2)))/EVAL N2))))
$d_1$	d1	Proportion of quality factor count that occurs on modules <b>not flagged</b> for inspection	(EVAL RF)/(EVAL TF)
$UL_{d_1}$	ULd1	Upper confidence limit of d1	((EVAL d1)+(Z*(SQRT (((EVAL d1)* (EVAL d2))/EVAL TF))))
$d_2$	d2	Proportion of <i>drcount</i> that occurs on modules <b>flagged</b> for inspection	(1-EVAL (d1))
$LL_{d_2}$	LLd2	Lower confidence limit of d2	((EVAL d2)-(Z*(SQRT (((EVAL d1)* (EVAL d2))/EVAL TF))))
$D_1$	D1	Expected quality factor count that occurs on modules <b>not flagged</b> for inspection	((EVAL RF)/(EVAL N1))*N1a
$D_2$	D2	Expected quality factor count that occurs on modules <b>flagged</b> for inspection	((EVAL TF)-(EVAL RF))/(EVAL N2))*N2a
$N_1'$	N1a	Count of accepted modules in <i>Application Sample</i>	SUM((M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (sample EQ Ss))
$N_2'$	N2a	Count of rejected modules in <i>Application Sample</i>	SUM(((M1 GT M1c) OR (M2 GT M2c) OR (M3 GT M3c) OR (M4 GT M4c)) AND (sample EQ Ss))
$Z_\alpha$	Z	Standardized difference between variable and mean of normal distribution	Given value based on choice of $\alpha$

## DISTFUBUTION LIST

<u>Agency</u>	<u>No. of Copies</u>
Defense Technical Information Center <b>8725 John J. King Rd., STE 0944</b> Ft. Belvoir, VA <b>22314</b>	<b>2</b>
Dudley Knox Library, Code <b>013</b> Naval Postgraduate School Monterey, CA <b>93943</b>	<b>2</b>
Office of Research Administration, Code <b>91</b> Naval Postgraduate School Monterey, CA <b>93943</b>	<b>1</b>
Department of Systems Management Library, Code SM/Eb Naval Postgraduate School <b>555 Dyer Rd Rm 239 Bldg. 330</b> Monterey, CA <b>93943</b>	<b>1</b>
Commanding Officer Marine Corps Tactical Systems Support Activity Box <b>555171</b> Camp Pendleton. CA <b>92055-5171</b>	<b>1</b>
Capt. Kenneth Warburton Marine Corps Tactical Systems Support Activity Box <b>555171</b> Bldg. <b>31345</b> Camp Pendleton, CA <b>92055-5171</b>	<b>5</b>
Dr. Norman F. Schneidewind Naval Postgraduate School Code SM/Ss Monterey, CA <b>93943</b>	<b>10</b>