1997-05-12

MCTSSA Software Reliability Handbook, Volume IV: Schneidewind Software Reliability and Metrics Model Tool List
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Schneidewind, Norman F.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>MCTSSA software reliability handbook. Volume IV</td>
</tr>
<tr>
<td>Publisher</td>
<td>Monterey, California. Naval Postgraduate School</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1997-05-12</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10945/15416">http://hdl.handle.net/10945/15416</a></td>
</tr>
</tbody>
</table>

This document was downloaded on March 15, 2013 at 10:39:06
The purpose of this handbook is threefold. Specifically, it:
- 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
- Serves as a tool for managing the software reliability program
- 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

**13. ABSTRACT (Maximum 200 words.)**

Approved for public release; distribution unlimited.
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
# TABLE OF CONTENTS

INTRODUCTION ........................................................................................................ 4

SOFTWARE RELIABILITY MODEL EQUATIONS

- NOTATION .................................................................................................................. 5
- EQUATIONS IMPLEMENTED IN SMERFS
  - 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
  - Risk Assessment .................................................................................................... 10
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:

- SOFTWARE RELIABILITY MODEL EQUATIONS
  - NOTATION
  - EQUATIONS IMPLEMENTED IN SMERFS
  - EQUATIONS IMPLEMENTED IN STATGRAPHICS
    * TABLE 1

- 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 SMERFS

- 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
- \( F(t) \): predicted failure count in the range \([1, t]\)
- \( F_i \): 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 \( F_{ij} > 0 \)
- \( \text{MSE}_f \): mean square error criterion for selecting \( s \) for failure count predictions
- \( \text{MSE}_r \): mean square error criterion for selecting \( s \) for remaining failure predictions
- \( \text{MSE}_\tau \): 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 \( r(t_c) \)
- \( r(t) \): remaining failures predicted at time \( t \)
- \( r(t_1) \): remaining failures predicted at total test time \( t_1 \)
- \( \Delta r(T_F, t) \): reduction in remaining failures that would be achieved if the software were executed for a time \( T_F \), predicted at time \( t \)
- \( \text{RCM} r(t_c) \): risk criterion metric for remaining failures at total test time \( t_c \)
- \( \text{RCM} T_F(t_1) \): risk criterion metric for time to next failure at total test time \( t_1 \)
- \( s \): 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 \( T_F(t_c) \)
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 t, t)  time to next N failures that would be achieved if remaining failures were reduced by \Delta t, predicted at time t
T_{ij}  time since interval i to observe number of failures F_{ij} during interval j; used in computing MSE
x_k  number of observed failures in interval k
X_t  observed failure count in the range [1,t]
X_{s-1}  observed failure count in the range [1,s-1]
x_{t,s}  observed failure count in the range [t,s]
x_{s,t}  observed failure count in the range [s,t]
x_{s,t;i}  observed failure count in the range [s_t]
x_t  observed failure count in the range [1,t]
x_{t;i}  observed failure count in the range [1,t_i]

EQUATIONS IMPLEMENTED IN SMERFS

Parameter Estimation

The log of the likelihood function is:

\[
\log L = \sum_{i=1}^{t} \left[ \log x_i - \log(1 - \exp(-\beta t)) \right] - \sum_{i=1}^{s} \left[ \log(1 - \exp(-\beta(s-1))) \right]
\]

This function is used to derive the equations for estimating \( a \) and \( \beta \) for each of the three methods. In the equations that follow, \( a \) 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 \( a \), respectively.

\[
\frac{1}{\exp(\beta)-1} \cdot \frac{t}{\exp(\beta_0)-1} \cdot \sum_{k=0}^{s-1} \frac{x_{k,t}}{x_t}
\]
\[ a = \frac{\beta X_i}{1 - \exp(-\beta t)} \]

**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 \( a \), respectively.

\[ \frac{1}{\exp(\beta - 1)} \sum_{x_i} \frac{X_{x_i}}{X_{x_{i+1}}} \]

\[ a = \frac{\beta X_i}{1 - \exp(-\beta (t-s+1))} \]

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 \( a \), respectively.

\[ \frac{(s-1)X_{x_{s-1}}}{\exp(\beta(s-1)) - 1} \cdot X_{x_{s+1}} \cdot \sum_{x_i} \frac{X_{x_i}}{X_{x_{i+1}}} \]

\[ a = \frac{\beta X_i}{1 - \exp(-\beta t)} \]

Method 3 is equivalent to Method 1 for \( s=2 \).

**Failures in an Interval Range**

Predicted failure count in the range \([t_1, t_2] \):

\[ F(t_1, t_2) = (\alpha/\beta)[1 - \exp(-\beta(t_2 - t_1))] \cdot X_{x_{t_1}} \]

**Maximum Failures**

Predicted failure count in the range \([1, \infty] \) \( \text{(i.e., maximum failures over the life of the software)} \):
F(∞)=α/β+X_{s,1}

(Note: Implemented in SMERFS but the user must make the manual correction of adding X_{s,1} to the quantity α/β that SMERFS computes).

**Remaining Failures**

Predicted remaining failures r(t) at time t:

r(t)=(α/β)-X_{s,t}+F(∞)-X_t

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

**Time to Next Failure**

Predicted time for the next F failures to occur, when the current time is t:

T_t(t)=\{log[α/(α-β(X_{s,t}-F_t))]/β\}-(t-s+1)

for (α/β)>(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=I))**

Mean Square Error (MSE_t) criterion for number of remaining failures, etc.:

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

where

F(i)=(α/β)[1-exp(-β((i-s+1)))]+X_{s,1}

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

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

\[
\text{MSE}_t = \sum_{i=s}^{J} \frac{[[\log[α/(α-β(X_{s,t}-F_i))]/β-(i-s+1)]-T_{i,t}]^2}{(J-s)}
\]

for (α/β)>(X_{s,t}-F_i)
EQUATIONS IMPLEMENTED IN STATGRAPHICS

Cumulative Failures

Predicted failure count in the range \([1, t]\):
\[ F(t) = \left( \frac{\alpha}{\beta} \right) [1 - \exp(-\beta((t-s+1))) + X_{t-1}] \]

Remaining Failures

Predicted remaining failures as a function of total test time \(t_r\):
\[ r(t) = \left( \frac{\alpha}{\beta} \right) \exp(-\beta[t_r-(s-1)]) \]

Fraction of Remaining Failures:

Fraction \(\alpha\) 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 \(\alpha\) remaining failures at \(t_r, r(t_r)\):
\[ t_r = \frac{\log\left[ 1 - \left( \frac{\alpha}{\beta} \frac{r(t)}{r(t_r)} \right) \right]}{\beta(s-1)} \]

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 \(T_f(\Delta r, t)\):
\[ T_f(\Delta r, t) = \frac{\log[1 - ((\beta \Delta r/\alpha) \exp(\beta(t-s+1)))]}{\beta(s-1)} \]

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 \(t\):
\[ \Delta r(T_f, t) = \left( \frac{\alpha}{\beta} \right) [\exp(-\beta(t-s+1))] [1 - \exp(-\beta(T_f))] \]

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

Mean Square Error criterion for failure counts:
Criteria for Safety

1) predicted remaining failures \( r(t_i) < r_c \), where \( r_c \) is a specified critical value, and

2) predicted time to next failure \( T_F(t_i) > t_m \), where \( t_m \) is mission duration.

Risk Assessment

Risk criterion metric for remaining failures at total test time \( t_* \):

\[
\text{RCM } r(t_i) = (r(t_i) - r_c)/r_c = (r(t_i)/r_c) - 1
\]

Risk criterion metric for time to next failure at total test time \( t_* \):

\[
\text{RCM } T_F(t_i) = (t_m - T_F(t_i))/t_m = 1 - (T_F(t_i))/t_m
\]

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

<table>
<thead>
<tr>
<th>Math Notation</th>
<th>Sgplus Notation</th>
<th>Statgraphics Definition</th>
<th>Sgplus Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>alpha</td>
<td>Beginning failure rate</td>
<td>From SMERFS</td>
</tr>
<tr>
<td>β</td>
<td>beta</td>
<td>Relative failure rate</td>
<td>From SMERFS</td>
</tr>
<tr>
<td>Δt</td>
<td>deltaR</td>
<td>Delta Remaining Failures</td>
<td>Given value</td>
</tr>
<tr>
<td>f(t)</td>
<td>d</td>
<td>Predicted Failure Rate</td>
<td>(alpha)<em>(EXP(-(beta</em>(i-(s-1))))))</td>
</tr>
<tr>
<td>F(i)</td>
<td>f</td>
<td>Predicted Cumulative Failures</td>
<td>((alpha/beta)<em>(1-EXP((-beta</em>((i-s)+1))))+Xs</td>
</tr>
<tr>
<td>Fij</td>
<td>Fij</td>
<td>Number of failures at j since i in MSEtf</td>
<td>From failure data</td>
</tr>
<tr>
<td>F_t</td>
<td>Fij</td>
<td>Number of failures to occur after interval t in tf</td>
<td>Given value</td>
</tr>
<tr>
<td>F(t)</td>
<td>Ft</td>
<td>Predicted Maximum Failures</td>
<td>(alpha/beta)+(Xs)</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
<td>Execution time index</td>
<td>From failure data</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>Maximum j≤t where Fij&gt;0</td>
<td>From failure data</td>
</tr>
<tr>
<td>m(i)</td>
<td>mi</td>
<td>Predicted failures in intervals</td>
<td>(alpha/beta)<em>(EXP(-(beta</em>(i-s)))))</td>
</tr>
<tr>
<td>MSE_f</td>
<td>MSE</td>
<td>MSE: Cumulative Failures</td>
<td>(SUM(((EVAL f)-Xs)^2))/((t-s)+1)</td>
</tr>
<tr>
<td>MSE</td>
<td>MSEr</td>
<td>MSE: Remaining Failures</td>
<td>SUM (((EVAL f)-Xt)^2)/((t-s)+1)</td>
</tr>
<tr>
<td>MSE_T</td>
<td>MSEtf</td>
<td>MSE: Time to Failure</td>
<td>(SUM(((EVAL tf)-Tij)^2))/((J-s))</td>
</tr>
<tr>
<td>p(t)</td>
<td>P</td>
<td>Fraction Remaining Failures</td>
<td>(Rtt)/(EVAL Ft)</td>
</tr>
<tr>
<td>Q(t)</td>
<td>Q</td>
<td>Predicted Program Quality</td>
<td>(1-(EVAL p))</td>
</tr>
<tr>
<td>r_c</td>
<td>Rc</td>
<td>Remaining Failures Criterion</td>
<td>Given value</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Formula</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>( r(t) )</td>
<td>Predicted remaining failures using ( X_t )</td>
<td>((\alpha/\beta)-(X_{st}))</td>
<td></td>
</tr>
<tr>
<td>( r(t) )</td>
<td>Predicted remaining failures, given ( \tau )</td>
<td>((\alpha/\beta))((\exp(-\beta \cdot (t-(s-1)))))</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>Predicted remaining failures using ( p )</td>
<td>(p)((\text{EVAL } F_t))</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>Number of remaining failures in computing ( p ) and ( \tau )</td>
<td>Given value</td>
<td></td>
</tr>
<tr>
<td>( \Delta r(T_F, t) )</td>
<td>Predicted delta Remaining Failures</td>
<td>((\alpha/\beta))((\exp(-\beta \cdot (t-(s-1)))))((1-\exp(-\beta \cdot TR))))</td>
<td></td>
</tr>
<tr>
<td>RCM</td>
<td>Risk of Remaining Failure</td>
<td>((\text{EVAL } rt)-Rc)/Rc)</td>
<td></td>
</tr>
<tr>
<td>RCM</td>
<td>Risk of Time to Failure</td>
<td>((tm-(\text{EVAL } tf))/tm)</td>
<td></td>
</tr>
<tr>
<td>( s )</td>
<td>First failure interval</td>
<td>From SMERFS</td>
<td></td>
</tr>
<tr>
<td>( t )</td>
<td>Execution time</td>
<td>From failure data</td>
<td></td>
</tr>
<tr>
<td>( t_F )</td>
<td>Predicted Total Test Time, given ( R_{tt} )</td>
<td>(((\log(\alpha/(\beta \cdot R_{tt}))/\beta)+(s-1))</td>
<td></td>
</tr>
<tr>
<td>( T_F(t) )</td>
<td>Predicted Time to Failure</td>
<td>(((1/\beta)\cdot(\log(\alpha/(\alpha-(\beta \cdot (X_{si}+F_{ij})))))/(i-(s-1)))))</td>
<td></td>
</tr>
<tr>
<td>( T_F(\Delta r, t) )</td>
<td>Time to Failure for delta Remaining Failures</td>
<td>(-1/\beta)((\log(1-((\beta/\alpha)\cdot(\delta R)\cdot\exp(\beta \cdot (i-(s-1))))))</td>
<td></td>
</tr>
<tr>
<td>( T_{ij} )</td>
<td>Time since ( j ) to fail at ( j )</td>
<td>From failure data</td>
<td></td>
</tr>
<tr>
<td>( t_m )</td>
<td>Time to Failure Criterion</td>
<td>From failure data</td>
<td></td>
</tr>
<tr>
<td>( T_F )</td>
<td>Given ( T_f ) for Predicted delta Remaining Failures</td>
<td>Given value</td>
<td></td>
</tr>
<tr>
<td>( X_{s-1} )</td>
<td>Observed failure count in the range ([1,s-1])</td>
<td>From failure data</td>
<td></td>
</tr>
<tr>
<td>( X_{s,i} )</td>
<td>Observed failure count in the range ([s,i])</td>
<td>From failure data</td>
<td></td>
</tr>
<tr>
<td>( X_{s,t} )</td>
<td>Observed failure count in the range ([s,t])</td>
<td>From failure data</td>
<td></td>
</tr>
<tr>
<td>( X_t )</td>
<td>Observed failure count in the range ([1,t])</td>
<td>From failure data</td>
<td></td>
</tr>
</tbody>
</table>
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

\( e_{CS} \): probability that one or more critical servers \( N_c \) 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**

\( P_{sys/node fails}(t) \): 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**

\( f_{cc}(I) = \text{COUNT as failures WHERE} (S \land G \land N \land \lnot C) \text{ in interval } I \)

\( f_{nc}(I) = \text{COUNT as failures WHERE} (S \land G \land \lnot N \land \lnot C) \text{ in interval } I \)

\( f_{cs}(I) = \text{COUNT as failures WHERE} (S \land \lnot G \land N \land \lnot C) \text{ in interval } I \)

\( f_{ns}(I) = \text{COUNT as failures WHERE} (S \land \lnot G \land \lnot N \land \lnot C) \text{ in interval } I \)

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

\( D = \sum d(I) \)

**EQUATIONS IMPLEMENTED USING WINDOWS CALCULATOR**

**Node Failure Probabilities**

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

\( p_{cc} = \frac{\sum f_{cc}(I)}{D} \)

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

\( p_{ns} = \frac{\sum f_{nc}(I)}{D} \)

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

\( p_{cs} = \frac{\sum f_{cs}(I)}{D} \)

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

\( p_{ns} = \frac{\sum f_{ns}(I)}{D} \)

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:

\( P_{ns}(t) = (p_{ns})^{N_{nc}(t)} \)
Probability that **one or more** critical servers $N_c$ 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**:

$$P_{sys/node fails}(t) = [P_{cc}][P_{nc}(t)] + [P_{cs}][P_{ns}(t)] = [1 - (1 - p_{cc})^{N_{cc}}][P_{nc}(t)] + [1 - (1 - p_{cs})^{N_{cs}}][P_{ns}(t)]$$

------------------------  ------------------------

Probability of Client Failure  Probability of Server Failure
### TABLE 2: STATGRAPHICS (SGPLUS) EQUATION IMPLEMENTATIONS DISTRIBUTED SYSTEM MODEL EQUATIONS

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

**EQUATION IMPLEMENTED IN SMERFS**

**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:

\[
T_{f}(t) - [\log[\alpha/(\alpha - \beta(X_{x,t} + F_i))] / \beta] - (t-s-1)
\]

for $(\alpha/\beta) > (X_{x,t} + F_i)$
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 $drcount > F_c$):

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

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

$$C_{1j} = \text{COUNT}_{i=1}^{n} \text{FOR} \ ((F_i \leq F_c) \land (M_{ij} \leq M_{n}), \land (M_{ij} \leq M_{n}), \ldots)$$

$$C_{12} = \text{COUNT}_{i=1}^{n} \text{FOR} \ ((F_i \leq F_c) \land (M_{ij} > M_{n}), \lor (M_{ij} > M_{n}), \lor (M_{ij} > M_{n}), \lor (M_{ij} > M_{n})))$$

$$C_{21} = \text{COUNT}_{i=1}^{n} \text{FOR} \ ((F_i > F_c) \land (M_{ij} \leq M_{n}), \land (M_{ij} \leq M_{n}), \ldots)$$

$$C_{22} = \text{COUNT}_{i=1}^{n} \text{FOR} \ ((F_i > F_c) \land (M_{ij} > M_{n}), \lor (M_{ij} > M_{n}), \lor (M_{ij} > M_{n}), \lor (M_{ij} > M_{n})))$$

Proportion of Type 1 Misclassifications:

$$P_1 = \frac{C_{21}}{n}$$

Proportion of Type 2 Misclassifications:

$$P_2 = \frac{C_{12}}{n}$$

Proportion of Type 1+Type 2 Misclassifications:

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

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

$$LQC = \frac{C_{22}}{n_2}$$
Remaining Factor RF (e.g., remaining drcount). This is the sum of F, not caught by inspection:

\[ RF = \sum_{i=1}^{n} F_i \text{ FOR } (F_i > F_c) \land (M_{ij} \leq M_{cl}) \land \ldots \land (M_{ik} \leq M_{cl}) \land \ldots \land (M_{im} \leq M_{cm}) \]

Proportion of RF, where TF is the total F prior to inspection:

\[ RFP = \frac{RF}{TF} \]

\[ TF = \sum_{i=1}^{n} F_i \]

Density of RF:

\[ RFD = \frac{RF}{n} \]

Proportion of modules remaining that have F,>F,:

\[ RMP = \frac{RFM}{n}, \]

where RFM is given by:

\[ RFM = \text{COUNT} \text{ FOR } ((F_i > 0) \land (M_{ij} \leq M_{cl}) \land \ldots \land (M_{ik} \leq M_{cl}) \land \ldots \land (M_{im} \leq M_{cm})) \]

Proportion of modules that must be inspected:

\[ I = \frac{C_{12} + C_{22}}{n} \]

Wasted inspection:

\[ RI = \frac{C_{22}}{C_{12}} \]

**EQUATION IMPLEMENTED USING WINDOWS CALCULATOR**

Quality Inspection Ratio:

\[ QIR = (\frac{\text{\Delta RFP}}{RFP})/(\frac{\text{\Delta I}}{I}) \]
## TABLE 3: STATGRAPHICS (SGPLUS) AND WINDOWS CALCULATOR EQUATION IMPLEMENTATIONS

**SOFTWARE METRICS MODELS EQUATIONS**

### DISCRIMINAT VE POWER VALIDATION MODEL

<table>
<thead>
<tr>
<th>Math Notation</th>
<th>Sgplus Notation</th>
<th>Statgraphics Definition</th>
<th>Sgplus Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₁₁</td>
<td>C₁₁</td>
<td>Module count for C₁₁</td>
<td>SUM((drcount LE Dc) AND (M₁ LE M₁c) AND (M₂ LE M₂c) AND (M₃ LE M₃c) AND (M₄ LE M₄c) AND (sample EQ Sₛ))</td>
</tr>
<tr>
<td>C₁₂</td>
<td>c₁₂</td>
<td>Module count for C₁₂</td>
<td>SUM((drcount LE Dc) AND ((M₁ GT M₁c) OR (M₂ GT M₂c) OR (M₃ GT M₃c) OR (M₄ GT M₄c)) AND (sample EQ Sₛ))</td>
</tr>
<tr>
<td>C₂₁</td>
<td>c₂₁</td>
<td>Module count for C₂₁</td>
<td>SUM((drcount GT Dc) AND (M₁ LE M₁c) AND (M₂ LE M₂c) AND (M₃ LE M₃c) AND (M₄ LE M₄c) AND (sample EQ Sₛ))</td>
</tr>
<tr>
<td>C₂₂</td>
<td>c₂₂</td>
<td>Module count for C₂₂</td>
<td>SUM((drcount GT Dc) AND ((M₁ GT M₁c) OR (M₂ GT M₂c) OR (M₃ GT M₃c) OR (M₄ GT M₄c)) AND (sample EQ Sₛ))</td>
</tr>
<tr>
<td>Fₑ</td>
<td>Dc</td>
<td>Quality factor critical value</td>
<td>Given value</td>
</tr>
<tr>
<td>Fᵢ</td>
<td>drcount</td>
<td>Vector of quality factor values</td>
<td>From quality factor data</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>Proportion of modules that must be inspected</td>
<td>(((EVAL C₁₂)+(EVAL C₂₂))/n)*100 %</td>
</tr>
<tr>
<td>ΔI</td>
<td>None</td>
<td>Difference in two successive values of I</td>
<td>Windows Calculator computation</td>
</tr>
<tr>
<td>1</td>
<td>Module name</td>
<td>Module index</td>
<td>From metrics file</td>
</tr>
<tr>
<td>j</td>
<td>Metric name</td>
<td>Metric index</td>
<td>From metrics file</td>
</tr>
<tr>
<td>K-S(Mₑᵢ)</td>
<td>maxcdfdiff</td>
<td>Maximum vertical difference between two CDFs</td>
<td>MAX (EVAL (cdfdiff)), where cdfdiff=(ABS(m₁-m₂))/100 &amp; m₁, m₂=metric vectors</td>
</tr>
<tr>
<td>LQC</td>
<td>LQC</td>
<td>Proportion of low quality software correctly classified</td>
<td>((EVAL C₂₂)/(EVAL n₂))*100 %</td>
</tr>
<tr>
<td>Mₑᵢ</td>
<td>M₁c...M₄c</td>
<td>Vector of j metric critical values</td>
<td>From metrics data and K-S test</td>
</tr>
<tr>
<td>Mᵢⱼ</td>
<td>M₁...M₄</td>
<td>Matrix of modules and metrics</td>
<td>From metrics data and K-W test</td>
</tr>
<tr>
<td>N₁</td>
<td>N₁</td>
<td>Count of accepted modules</td>
<td>SUM((M₁ LE M₁c) AND (M₂ LE M₂c) AND (M₃ LE M₃c) AND (M₄ LE M₄c) AND (sample EQ Sₛ))</td>
</tr>
<tr>
<td>N2</td>
<td>N2</td>
<td>Count of rejected modules</td>
<td>SUM(((M1 GT M1c) OR (M2 GT M2c) OR (M3 GT M3c) OR (M4 GT M4c)) AND (sample EQ Ss))</td>
</tr>
<tr>
<td>----</td>
<td>-----</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>n</td>
<td>n</td>
<td>Number of modules in sample</td>
<td>Given value</td>
</tr>
<tr>
<td>n1</td>
<td>n1</td>
<td>Count of high quality modules</td>
<td>(EVAL C11)+(EVAL C12)</td>
</tr>
<tr>
<td>n2</td>
<td>n2</td>
<td>Count of low quality modules</td>
<td>(EVAL C21)+(EVAL C22)</td>
</tr>
<tr>
<td>P1</td>
<td>P1</td>
<td>Proportion of Type 1 misclassifications</td>
<td>((EVAL C21)/n)*100 %</td>
</tr>
<tr>
<td>P2</td>
<td>P2</td>
<td>Proportion of Type 2 misclassifications</td>
<td>((EVAL C12)/n)*100 %</td>
</tr>
<tr>
<td>P12</td>
<td>P12</td>
<td>Proportion of Type 1+Type 2 misclassifications</td>
<td>(((EVAL C12)+(EVAL C21))/n)*100 %</td>
</tr>
<tr>
<td>QIR</td>
<td>None</td>
<td>Quality Inspection Ratio</td>
<td>Windows Calculator computation</td>
</tr>
<tr>
<td>FF</td>
<td>RF</td>
<td>Remaining Quality Factor</td>
<td>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)))</td>
</tr>
<tr>
<td>RFD</td>
<td>RFD</td>
<td>Density of RF</td>
<td>(EVAL RF)/n</td>
</tr>
<tr>
<td>RFP</td>
<td>RFP</td>
<td>Proportion of RF</td>
<td>((EVAL RF)/(EVAL TF))*100 %</td>
</tr>
<tr>
<td>ARFP</td>
<td>None</td>
<td>Difference in two successive values of RFP</td>
<td>Windows Calculator computation</td>
</tr>
<tr>
<td>RFM</td>
<td>RFM</td>
<td>Count of modules with Remaining Quality Factor</td>
<td>SUM ((drcount GT Dc) AND (M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c) AND (M4 LE M4c) AND (sample EQ Ss))</td>
</tr>
<tr>
<td>RMP</td>
<td>RMP</td>
<td>Proportion of RFM</td>
<td>((EVAL RFM)/n)*100 %</td>
</tr>
<tr>
<td>RI</td>
<td>RI</td>
<td>Wasted Inspection</td>
<td>(EVAL C22)/(EVAL C12)</td>
</tr>
<tr>
<td>None</td>
<td>SS</td>
<td>Sample Identification</td>
<td>Given value</td>
</tr>
<tr>
<td>TF</td>
<td>TF</td>
<td>Total Quality Factor</td>
<td>SUM (drcount SELECT sample EQ Ss)</td>
</tr>
<tr>
<td>$\chi^2_c$</td>
<td>$\chi^2_c$</td>
<td>Critical value of Chi-Square</td>
<td>Function of C11, C12, C21, and C22</td>
</tr>
</tbody>
</table>
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:

$$p_p = \frac{\text{( COUNT FOR } F_i > 0)}{n}.$$  

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

$$\text{CL}_{p_p} = p_p \pm Z_{\alpha/2} \sqrt{\frac{(p_p)(1-p_p)}{n}}.$$  

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

$p_{N_1} = \frac{RFM}{N_1}$

One-sided upper confidence limit of $p_{N_1}$, used as predicted limit of $p_{N_1}'$ in the Application Sample:

$$\text{UL}_{p_{N_1}} = \frac{(p_{N_1})(n) - (RFM)}{N_1}.$$  

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

$p_{N_2} = \frac{(p_{N_1})(n) - (RFM)}{N_2}$

One-sided lower confidence limit of $p_{N_2}$, used as predicted limit of $p_{N_2}'$ in the Application Sample:

$$\text{LL}_{p_{N_2}} = \frac{(p_{N_1})(n) - (RFM)}{N_2}.$$  

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

$d_i = \frac{RF}{TF}$ (same as RFP if FWP is expressed as a proportion)

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

$$\text{UL}_{d_i} = \frac{(d_i)(1-d_i)}{TF}.$$  

21
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:

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

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>
<thead>
<tr>
<th>Math Notation</th>
<th>Sgplus Notation</th>
<th>Statgraphics Definition</th>
<th>Sgplus Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_s$</td>
<td>$pn$</td>
<td>Proportion of modules with $F &gt; 0$</td>
<td>$\frac{\text{SUM}((\text{drcount GT 0}) \text{ AND (sample EQ Ss)})}{n}$</td>
</tr>
<tr>
<td>CL$p_s$</td>
<td>CL$pn$</td>
<td>Two-sided confidence limits of $pn$</td>
<td>$((\text{EVAL } pn)+(Z*(\text{SQR}T ((\text{EVAL } pn)^* (1-(\text{EVAL } pn)/n)))$</td>
</tr>
<tr>
<td>$p_{N1}$</td>
<td>$pN1$</td>
<td>Proportion of modules not flagged for inspection</td>
<td>$\frac{\text{EVAL RFM}}{\text{EVAL N1}}$</td>
</tr>
<tr>
<td>UL$p_{N1}$</td>
<td>UL$pN1$</td>
<td>Upper confidence limit of $pN1$</td>
<td>$((\text{EVAL } pN1)+(Z*(\text{SQR}T ((\text{EVAL } pN1)^* (1-(\text{EVAL } pN1))/((\text{EVAL } N1)))))$</td>
</tr>
<tr>
<td>$p_{N2}$</td>
<td>$pN2$</td>
<td>Proportion of modules flagged for inspection</td>
<td>$((n*(\text{EVAL } pn))-(\text{EVAL RFM}))/((\text{EVAL } N2))$</td>
</tr>
<tr>
<td>LL$p_{N2}$</td>
<td>LL$pN2$</td>
<td>Lower confidence limit of $pN1$</td>
<td>$((\text{EVAL } pN2)-(Z*(\text{SQR}T ((\text{EVAL } pN2)^* (1-(\text{EVAL } pN2))/((\text{EVAL } N2))))))$</td>
</tr>
<tr>
<td>$d_1$</td>
<td>$d1$</td>
<td>Proportion of quality factor count that occurs on modules not flagged for inspection</td>
<td>$\frac{\text{EVAL RF}}{\text{EVAL TF}}$</td>
</tr>
<tr>
<td>UL$d_1$</td>
<td>UL$d1$</td>
<td>Upper confidence limit of $d1$</td>
<td>$((\text{EVAL } d1)+(Z*(\text{SQR}T ((\text{EVAL } d1)^* (1-(\text{EVAL } d1))/((\text{EVAL } TF)))))$</td>
</tr>
<tr>
<td>$d_2$</td>
<td>$d2$</td>
<td>Proportion of $drcount$ that occurs on modules flagged for inspection</td>
<td>$1-\text{EVAL } (d1)$</td>
</tr>
<tr>
<td>LL$d_2$</td>
<td>LL$d2$</td>
<td>Lower confidence limit of $d2$</td>
<td>$((\text{EVAL } d2)-(Z*(\text{SQR}T ((\text{EVAL } d1)^* (1-(\text{EVAL } d2))/((\text{EVAL } TF)))))$</td>
</tr>
<tr>
<td>$D_1$</td>
<td>$D1$</td>
<td>Expected quality factor count that occurs on modules not flagged for inspection</td>
<td>$\frac{\text{EVAL RF}}{\text{EVAL N1}})*N1a$</td>
</tr>
<tr>
<td>$D_2$</td>
<td>$D2$</td>
<td>Expected quality factor count that occurs on modules flagged for inspection</td>
<td>$((\text{EVAL TF})-(\text{EVAL RF}))/((\text{EVAL N2})))*N2a$</td>
</tr>
<tr>
<td>$N_1$</td>
<td>$N1a$</td>
<td>Count of accepted modules in Application Sample</td>
<td>$\text{SUM}((\text{M1 LE M1c) AND (M2 LE M2c) AND (M3 LE M3c)) AND (M4 LE M4c) AND (sample EQ Ss})$</td>
</tr>
<tr>
<td>$N'_1$</td>
<td>$N2a$</td>
<td>Count of rejected modules in Application Sample</td>
<td>$\text{SUM}((\text{M1 GT M1c) OR (M2 GT M2c) OR (M3 GT M3c) OR (M4 GT M4c)) AND (sample EQ Ss})$</td>
</tr>
<tr>
<td>$Z_e$</td>
<td>$Z$</td>
<td>Standardized difference between variable and mean of normal distribution</td>
<td>Given value based on choice of $\alpha$</td>
</tr>
</tbody>
</table>
## DISTRIBUTION LIST

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