SENG 521 Software Reliability & Quality

Software Reliability Tools (Chapter 12)

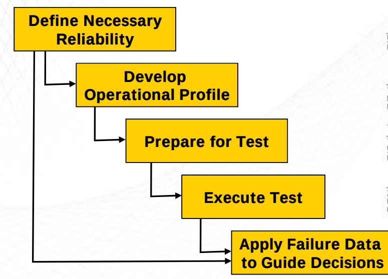
Department of Electrical & Computer Engineering, University of Calgary

B.H. Far (far@ucalgary.ca)

http://www.enel.ucalgary.ca/People/far/Lectures/SENG521

SRE: Process (Review)

- 5 steps in SRE process:
  - Define necessary reliability
  - Develop operational profiles
  - Prepare for test
  - Execute test
  - Apply failure data to guide decisions

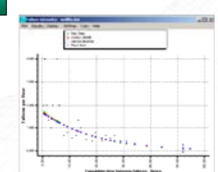


Section 1

Software Reliability Engineering Tools

SRE Tools: Reliability Growth

Time(s)	Cumulative Failures	Failures in interval
30	2	2
60	5	3
90	7	2
120	8	1
150	10	2
180	11	1
210	12	1
240	13	1
270	14	1



Failure Intensity

Tasks Handled by SRE Tools

- Collecting failure and test time information
- Calculating estimates of model parameters using this information
- Testing to fit a model against the collected information
- Selecting a model to make predictions of remaining faults, time to test, etc.
- Applying the model

Available Options /1

- Selection of a tool is one of the important decisions in performing the SRE study.
- An inappropriate choice may not handle the type of data collected for the project, or does not have a robust set of models that may fit to the project to make accurate predictions of important information.
- Engineers may choose between:
  - Using a general-purpose application program such as a spreadsheet or a statistical package such as SAS and developing their own models using a general-purpose programming language such as JAVA or C.
  - Using a shareware, freeware or commercially available SRE tool.

Available Options /2

Advantages of dedicated (commercial or freeware) SRE tools:

- Provide a general framework for reliability estimation and prediction.
- Provide most of the features needed in executing a software reliability analysis, resulting in a decrease of programming time.
- Comparing multiple models on the same failure data and changing the analysis to use a different model is easier to accomplish.
- Provide better error detection because many potential types of errors have been identified and are checked for automatically. The chance of a bug in the tool itself is very small.
- The basic structure of the models is from the theories developed by academic researchers and uses the terminology of those models.

Selecting SRE Tool /1

- Criteria for selecting SRE tools:
  - Availability of the tool, either in-house or on a network, for running on the company's computer system(s)
  - Cost of installing and maintaining the tool
  - Number of studies likely to be done
  - Types of systems to be studied
  - Quality of the tool documentation and support

Selecting SRE Tool /2

- Criteria for selecting a SRE tool: (contd.)
  - Ease of learning the tool
  - Flexibility and power of the tool
  - Goals and questions to be answered by the study
  - Models and statistical techniques understood by the analyst
  - Schedule for the project and type of data collected
  - Tool's ability to communicate the nature of the model and the results to a person other than the analyst (e.g., the end user or a manager).

Input Data Specification /1

- All of the SRE tools use one of two basic types of input data:
  - time-domain data
    (*i.e.*, time-between-failures data)
  - interval-domain data
    (*i.e.*, failure-count data)

Input Data Specification /2

1) Time of failure
2) Time interval between failures
3) Cumulative failure up to a given time
4) Failures experienced in a time interval

Time based failure specification

Failure no.	Failure times (hours)	Failure interval (hours)
1	10	10
2	19	9
3	32	13
4	43	11
5	58	15
6	70	12
7	88	18
8	103	15
9	125	22
10	150	25
11	169	19
12	199	30
13	231	32
14	256	25
15	296	40

Input Data Specification /3

1) Time of failure
2) Time interval between failures
3) Cumulative failure up to a given time
4) Failures experienced in a time interval

Failure based failure specification

Time(s)	Cumulative Failures	Failures in interval
30	2	2
60	5	3
90	7	2
120	8	1
150	10	2
180	11	1
210	12	1
240	13	1
270	14	1

SRE Tools

- CASRE
- SoftRel
- SMERFS
- SoRel
- SRMP
- ProConf
- Relex
- MEADEP (MEAsure and DEPendability)

etc.

SRE tool repository:

1. Open channel software: http://www.openchannelsoftware.org/discipline/Reliability_Analysis/

2. Univ of Maryland http://www.enre.umd.edu/tool.htm

SRE Tools (cont'd)

- **ACARA II:** Availability, Cost, And Resource Allocation, Version 2 *(no charge per license)*
- **ARAM:** Automated Reliability/Availability/Maintainability, Version 2.0 *($400 source code license)*
- **ETARA:** Event Time Availability, Reliability Analysis *($200 source code license)*
- **GO:** Graphics Oriented Program *($150 source code license)*
- **HARP:** Hybrid Automated Reliability Predictor, Version 7.0 *($500 source code license, for Unix or PC)*
- **HARPO:** Hybrid Automated Reliability Predictor Output Graphics Display *($150 source code license)*
- **SPRPM:** Software Problem Report Metrics Program *(no charge per license, requires EXCEL)*

SRE Tools: SMERFS /1

- Statistical Modeling and Estimation of Reliability Functions for Software
- SMERFS is a public-domain software package designed and implemented at the Naval Surface Warfare Center.
- SMERFS is a program for estimating and predicting the reliability of software during the testing phase.
- The body of code is in Fortran.

SRE Tools: SMERFS /2

- It offers flexibility in data collection and provides multiple time domain and interval domains. Therefore useful for multi-model debugging.
- SMERFS prompts the user for the name of the
  - History file
  - Plot file
  - Input data file
  - Output data file

SRE Tools: SMERFS /3

- **History file** is an output file created by SMERFS. It is a trace file that contains all of the user input and SMERFS outputs for a particular run so that the user can go back and look at the run at a later time.
- **Plot file** contains the raw output data in plotting results.

SRE Tools: SMERFS /4

- **Input data file** contains the failure history data on which SMERFS will actually operate to produce the reliability estimates and predictions.
- The user must also specify the type of data contained in the input data.
- If the selected data type does not correspond to the type of data actually in the input file, the estimates and predictions made by SMERFS will not be valid.

SRE Tools: SMERFS /5

- **Output data file** is a file that the user can specify to which SMERFS will write failure history data created or edited by the user during the current SMERFS session.
- This is different from the history file, since the history file is a trace file which records all user input and SMERFS responses.
- The output data file can be used in subsequent sessions as an input data file.
- The output file is in SMERFS format, not ASCII format.

SRE Tools: SRMP /1

Statistical Modeling and Reliability Program

- The SRMP was developed by the Reliability and Statistical Consultants, Limited of UK in 1988.
- SRMP is a command-line-oriented tool developed for an IBM PC/AT and also UNIX based workstations.
- SRMP contains nine models.
- SRMP uses the maximum likelihood estimation technique to compute the model parameters, and provides the following reliability indicators:
  - reliability function, failure rate, mean time to failure, median time to failure, and the model parameters for each model.

SRE Tools: SRMP /2

- SRMP requires an ASCII data file as input.
- The file contains the name (or other identification) of the project, the number of failures involved in the reliability analysis, and the inter-failure times of all the failures.
- The input file also specifies the initial sample size used by SRMP for the initial fitting of each reliability model to the data.
- The remaining failures are used by SRMP for assessing a reliability model's prediction accuracy.
- The input file contains certain mathematical parameters, chosen by the analyst, which are needed to initiate and control the SRMP algorithm's search for a convergent solution.
- Analysts must be knowledgeable in setting up the data file, as many parameters are at their discretion.

SRE Tools: SoftRel /1

- Characteristics of SoftRel
  - Console-based application written in C (about 1300 lines of code)
  - Source code is available
  - One input project file (formatted text)
  - Generates one output file (CSV)

SRE Tools: SoftRel /2

- The fundamental difference is that SoftRel is a *simulation tool*, rather than a *reliability growth* modeling tool, i.e., one can simulate the interdependencies between components.
- **Example:** what will be the effect of producing more documentation vs. more coding? (assuming requirement-design-coding-test lifecycle)
- SoftRel uses Piecewise-Poisson Markov Process to simulate project occurrences
- **Limitations:** SoftRel is limited to studying a project that has the standard waterfall lifecycle.

SRE Tools: SoRel /1

- **So**ftware **Rel**iability Program
- SoRel is a Macintosh-based reliability measurement tool that was developed by LAAS, a lab of the National Center for Scientific Research in France, in 1991.
- SoRel is composed of two parts:
  - The first part allows several reliability trend tests. These tests allow us to identify whether the reliability function is increasing or decreasing so that an appropriate model can be applied.
  - The second part allows reliability growth model application and contains 4 models.

SRE Tools: SoRel /2

- SoRel uses the maximum likelihood parameter estimation technique and provides the following reliability indicators:
  - mean time to failure, cumulative number of failures, failure intensity, model parameters to evaluate other reliability functions.
- Only one model is executed at a time. Execution results are automatically saved to ASCII files which can be imported into spreadsheets or other applications for model comparisons.

SRE Tools: SoRel /3

- SoRel uses ASCII input files that are created using a spreadsheet.
- Numerical results are displayed on the screen during execution; the user can also request plots of the data.

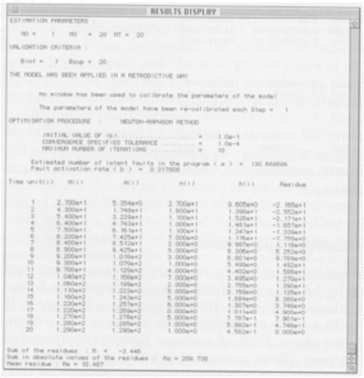


Figure from SRE Handbook

SRE Tools: CASRE /1

Computer-**A**ided **S**oftware **R**eliability **E**stimation Tool

- CASRE is copyrighted by NASA.
- CASRE is a PC-based tool that was developed in 1993 by the Jet Propulsion Laboratories to address the ease-of-use issues of other tools.
- CASRE requires the Windows operating environment.
- It has a pull-down, menu-driven user interface and uses the same model library as the SMERFS tool with the additional feature of allowing linear combinations of models to create new ones at the user's discretion.
- Four combined models are permanently available in CASRE.
- CASRE ver. 3.0 is available (http://www.openchannelsoftware.org/discipline/Reliability_Analysis)

SRE Tools: CASRE /2

- CASRE allows an analyst to invoke a text editor or other application from within CASRE to create the ASCII input data set. The input data set contains fields for the test interval number, number of failures observed in the interval, length of the test interval, fraction of the program tested, and severity of the failure.
- Once the data is entered, CASRE automatically provides the analyst with a raw data plot.
- CASRE provides the analyst with the ability to convert from time-domain data to interval-domain data and vice versa.
- Model parameters can be estimated using either maximum likelihood or least squares decided by the analyst. After the application of several models to a data set, multiple model results can be displayed in the graphical display window for analysis.

SRE Tools: CASRE /3

- CASRE provides operations to transform or smooth the failure data; the user can select and/or define multiple models for application to the data and make reliability predictions based on the best model.

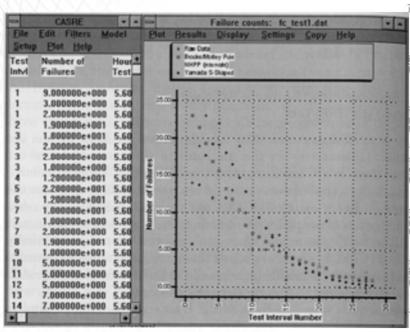


Figure from SRE Handbook

More Info

- Download tools:
  - IEEE Software Reliability Engineering Working Group (SREWG)
    http://www.srewg.org/Tools/
- SRE tools repository:
  - Center for Reliability Engineering at the University of Maryland
    http://www.enre.umd.edu/tool.htm
  - Open Channel Foundation
    http://www.openchannelsoftware.org/discipline/Reliability_Analysis

Section 2

How to use CASRE

CASRE: Introduction

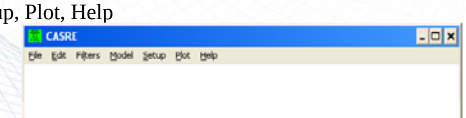
- Software Reliability Estimation tool running on Windows
- CASRE extends the SMERFS package by adding a menu based GUI
- Uses ASCII text input data files
- Displays results in tabular and/or graphical form
- Can use many different models

CASRE Program Structure

- Main Window
  - The window where the input data file is loaded and displayed.
  - Menu options allow the user to apply models and filters to the input data.
- Graphical Display Window
  - Displays a plot of the input data, as well as the results of any models applied to the data.
- Model Results Table
  - Displays the tabulated results from the models that were used in the calculation.

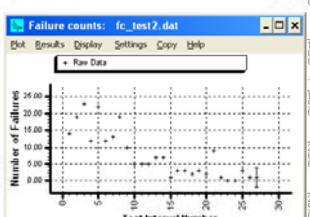
Main Window

- The main window is the starting point for CASRE sessions.This is the place where the user selects the models and filters to apply to the input data.
- Menu Options
  - File (Open, Save, Print, Exit)
  - Edit (Change Data Type, External Application, Escape to DOS)
  - Filters (Shaping and Scaling, Change time unit, etc.
  - Model (Select and Run, Define Combination, Edit/Remove Models, Parameter Estimation, Select Data Range, Predictions)
  - Setup, Plot, Help



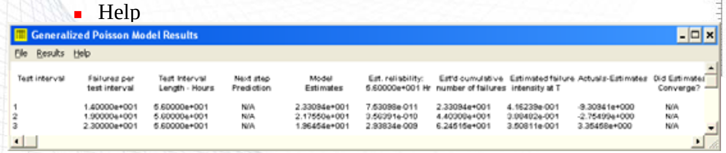
Graphical Display

- Provides the plots of the input and calculated data.
- Each individual data set on a plot has its own unique symbol and colour.
- Menu Options
  - Plot (Save as, Draw from File, Setup Printer, Print Plot)
  - Results (Select Model Results, Model Results Table)
  - Display
    - Graphs - Time between failures, Failure counts, Failure intensity, Test interval lengths, Cumulative failures, Reliability
    - Model Evaluation - Goodness-of-fit, Prequential likelihood, Relative accuracy, Bias, Bias trend, Bias scatter plot, Model noise, Model ranking
  - Settings, Copy, Help



Model Results Window

- Displays the detailed calculated results in a tabular format.
- Reliability estimates, parameter estimates, and convergence information are all displayed in this table for a selected model.
- Menu Options
  - File
  - Results – Select Results, Previous Model, Next Model
  - Help



CASRE Data Input

- ASCII based text file with a .dat extension
- Two file formats
  - Time Between Failures (error #, time since last failure, failure severity class)
  - Failure Counts (interval #, # errors in interval, interval length, failure severity class)
- The format of the file must be strictly adhered to
- No direct manipulation of the data file is allowed but CASRE has menu links to common text editors

Using CASRE /1

1. Prepare input data
   - Input data can be either failure count or failure per interval data

Sample failure count data

<failure number> <number of natural or time units since previous failure> <severity class>

1	30	1
2	55	1
3	70	1
4	60	1
5	90	1
6	110	1
7	100	1
8	150	1
9	120	1
10	215	1

Example from Musa's Book

Using CASRE /2

1. Prepare input data
   - Input data can be either failure count or failure per interval data

Sample failure per interval data

<interval number> <failure in interval> <duration of interval> 0 0 0 <severity class>

1	5	2.5	0	0	0	1
2	1	1	0	0	0	1
3	4	3	0	0	0	1
4	1	2	0	0	0	1
5	0	1.5	0	0	0	1
6	1	3	0	0	0	1
7	2	4	0	0	0	1
8	1	2.5	0	0	0	1
9	2	3	0	0	0	1
10	2	5	0	0	0	1

Example from Musa's Book

Using CASRE /3

2. Check if data shows reliability growth (trend test)
3. Read input file
4. Select data range
5. Filter or smooth input data if required
6. Select parameter estimation method
7. Select and run model(s)
8. View and interpret model results
   - Goodness of fit test
   - Model ranking
   - Prediction based on plots

CASRE Reliability Models /1

- Time between failure models
  - Geometric
  - Jelinski-Moranda
  - Littlewood-Verrall
  - Musa-Basic
  - Musa-Okumoto
  - NHPP

CASRE Reliability Models /2

- Failure Count models
  - Generalized Poisson
  - NHPP
  - Schneidewind
  - Shick-Wolverton
  - Yamada S-shaped

CASRE Reliability Models /3

- Combination models:
  - Four predefined models
    - Dynamically weighted
    - Equally weighted
    - Median Weighted
    - Unequally weighted
  - Other combination models can be defined

Trend Check

- CASRE models should only be used on data where the overall reliability is increasing as testing continues.
- Reliability is increasing if the mean time between failures increases as the total number of failures increases during testing.
- CASRE version 2 does not have the automatic trend test option, so a visual inspection of the cumulative failure plot is necessary.
- CASRE version 3 has an automatic trend test option, that will inform the user if the data is applicable to the reliability models.

Trend Related Questions …

- Is the system reliability increasing, decreasing or stable?
- Which reliability growth model fits best the gathered data?
- Can the same model be used in all cases of reliability growth, decrease and stable?

And Trend Related Answers …

- Reliability trends can be analyzed by "trend tests".
- Trend tests can be used to help determine whether the system undergoes reliability growth, decrease or stable reliability.
- Trend analysis also helps select appropriate reliability model for each phase.

Failure Data for Trend Tests

- The trend tests work with the failure data.
- The trend can be analyzed using
  - Inter-failure times data (i.e. time of failure known) or
  - Failure count data (i.e. failure per interval known)

Inter-failure Times Data /1

- Two trend tests are commonly carried:
  - Arithmetical mean test
  - Laplace tests

Inter-failure Times Data /2

- The arithmetical mean of the inter-failure times consists of calculating arithmetical mean $\tau(i)$ of the observed inter-failure times θj.

$$\tau(i) = \frac{1}{i}\sum_{j=1}^{i}\theta_j$$

- An increasing series of $\tau(i)$ indicates reliability growth and a decreasing series suggests reliability decrease.

Inter-failure Times Data /3

- For $N(T)$ as the cumulative number of failures over the time period $[0, T]$, the Laplace factor $u(T)$ is derived:

$$u(i) = \frac{\frac{1}{i-1}\sum_{n=1}^{i-1}\sum_{j=1}^{n}\theta_j - \frac{\sum_{j=1}^{i}\theta_j}{2}}{\sum_{j=1}^{i}\theta_j\sqrt{\frac{1}{12(i-1)}}}$$

- For the case that T is equal to the time of occurrence of failure i.

Inter-failure Times Data /4

- Negative values of the Laplace factor $u(i)$ indicate a decreasing failure intensity, i.e., reliability growth.
- Positive values of the Laplace factor $u(i)$ indicate an increasing failure intensity, i.e., reliability decrease.
- Values between –2 and +2 indicate stable reliability.

Inter-failure Times Data /5

- Midpoint of the observation interval: $T/2$
- Statistical center of data:

$$\frac{1}{N(T)\sum_{n=1}^{N(T)}\sum_{j=1}^{n}\theta_j}$$

- For the failure intensity decrease, the interfailure times θj tend to occur before the midpoint; hence the statistical center tend to be smaller than the mid-interval.
- For the failure intensity increase, the statistical center tend to be larger than the mid-interval.

Failure Count Data /1

- For the time period $[0, T]$, divided into k units of equal length and for $n(i)$ be the number of failures observed during the time interval i, the Laplace factor $u(k)$ is derived by:

$$u(k) = \frac{\sum_{i=1}^{k}(i-1)n(i) - \left(\frac{k-1}{2}\right)\sum_{i=1}^{k}n(i)}{\sqrt{\frac{k^2-1}{12}\sum_{i=1}^{k}n(i)}}$$

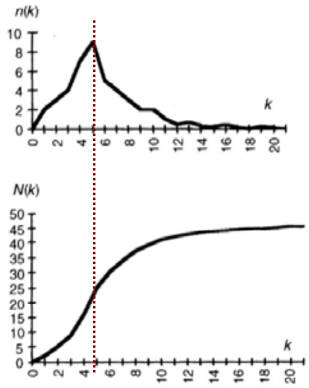
Failure Count Data /2

- Negative values of the Laplace factor $u(k)$ indicate a decreasing failure intensity, i.e., reliability growth.
- Positive values of the Laplace factor $u(k)$ indicate an increasing failure intensity, i.e., reliability decrease.

Typical Plots /1

- Typical graphs for failure intensity *n(k)* and cumulative failure intensity *N(k)*

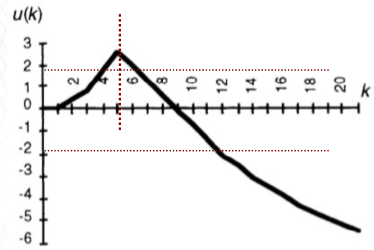
  (*k* is number of intervals)



Typical Plots /2

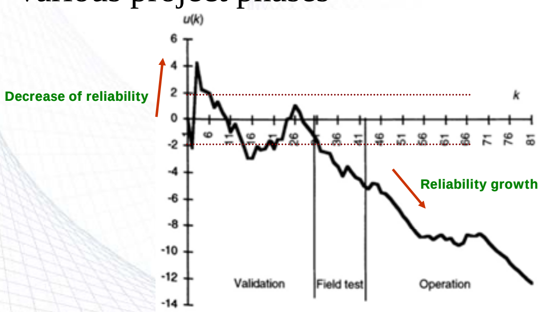
- Typical plot for the Laplace factor *u(k)*

  (*k* is number of intervals)



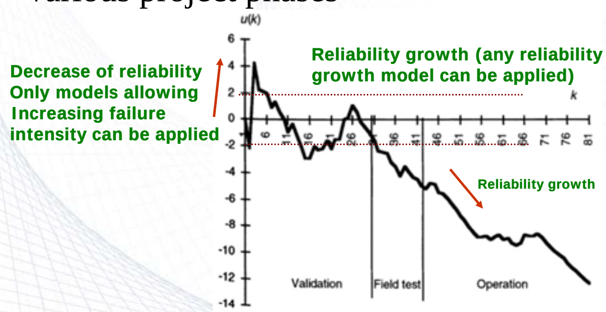
Typical Plots /3

- Typical plot for Laplace factor during various project phases



Selecting Models

- Typical plot for Laplace factor during various project phases



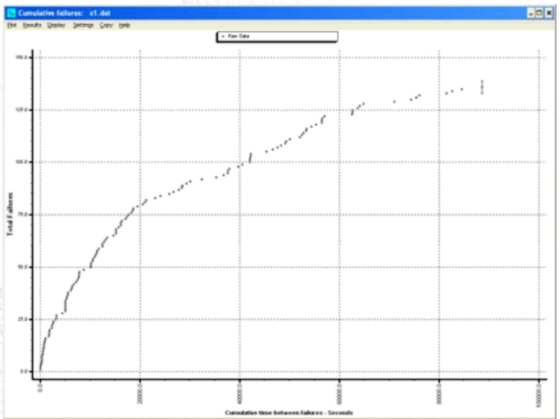
CASRE: Case Study

- Project X is a web based application for accessing a database using a browser.
- This version of the software is a minor release with changes to the GUI display and data access engine.
- Two programmers were assigned to the project. One programmer worked on the GUI, and the other on the data access engine.
- The project took approximately 4 weeks to complete.

Case Study (contd.)

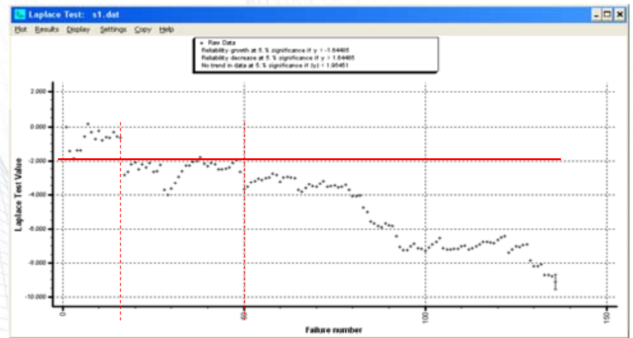
- A single tester was assigned to the project.
- The test phase was completed in approximately 25 hours (3 working days or 90,000 seconds).
- 136 failures were discovered during the testing.
- Using the dates and times recorded for the failures discovered during testing, a “time between failures” input file was generated for CASRE.
- The severity of all the failures was set to
  - 1 - Low Severity

Time Between Failures Plot



Trend Analysis

- Laplace test shows reliability growth.



Project Results

- In order to determine which models would provide the best fit for the project data, the following models were run
  - Geometric
  - Jelinski - Moranda
  - Littlewood - Verrall
  - Musa Basic
  - Musa - Okumoto

Goodness of Fit Test

On Graphic display window select:
Display → Goodness of fit

Goodness of Fit: s1.dat

Model Name	KS Distance	95% Fit?
Geometric	8.811010e-002	Yes
Jelinski-Moranda	9.443151e-002	Yes
Littlewood-Verrall	7.504933e-002	Yes
Musa Basic	9.203077e-002	Yes
Musa-Okumoto	8.791571e-002	Yes

CASRE Model Ranking

On Graphic display window select:
Display → Model rankings → Rank summary or Rank details

Model Ranking Summary: s1.dat

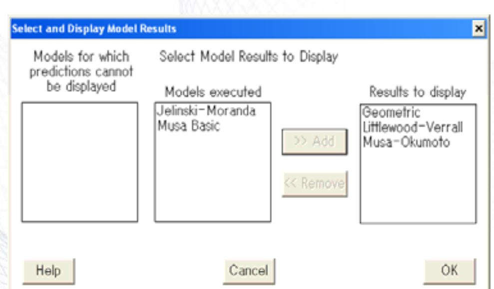
Model Name	Rank	Reliability
Geometric	1	1.463635e-001
Littlewood-Verrall	1	9.667713e-002
Musa-Okumoto	1	1.578047e-001
Musa Basic	4	3.278631e-001
Jelinski-Moranda	5	3.702861e-001

Model Ranking Detail: s1.dat

Model Name	-ln PL	Model Bias	Bias Trend	Model Noise	KS Distance	Rank
Geometric	5.310828e+002 (1)	7.806553e-002 (1)	1.084881e-001 (3)	2.026753e+000 (2)	8.811010e-002 (3)	1
Littlewood-Verrall	5.314613e+002 (3)	1.023081e-001 (3)	9.225496e-002 (1)	1.632047e+000 (1)	7.504933e-002 (1)	1
Musa-Okumoto	5.311823e+002 (2)	8.197740e-002 (2)	1.056971e-001 (2)	2.307074e+000 (3)	8.791571e-002 (2)	1
Musa Basic	5.376974e+002 (4)	1.935277e-001 (4)	1.556185e-001 (5)	5.351962e+000 (4)	9.203077e-002 (4)	4
Jelinski-Moranda	5.387762e+002 (5)	2.201526e-001 (5)	1.542180e-001 (4)	5.582339e+000 (5)	9.443151e-002 (5)	5

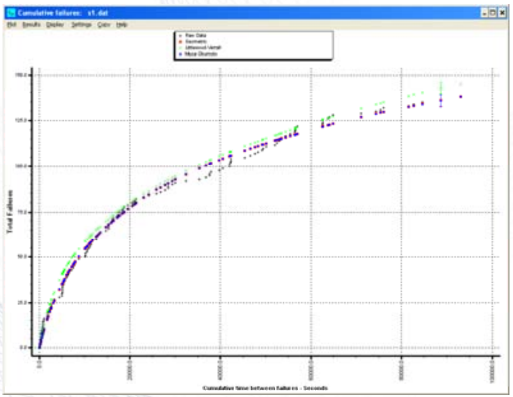
Display Resuls

On Graphic display window select:
Results → Select model results

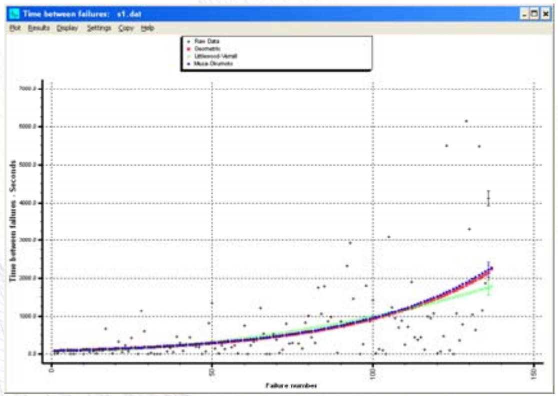


Only 3 graphs can be displayed at a time

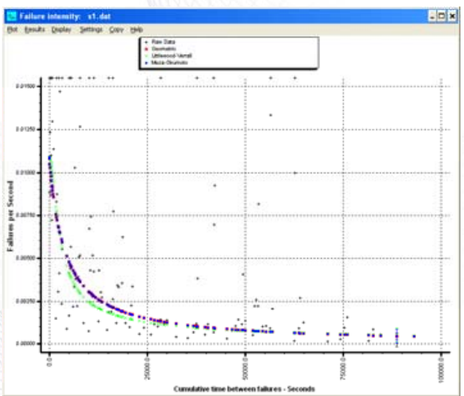
Display: Cumulative Failures



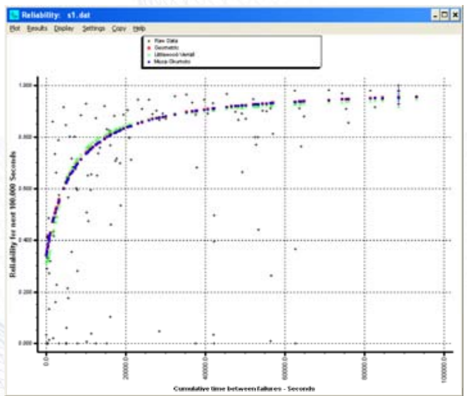
Display: Time Between Failures



Display: Failure Intensity



Display: Reliability



Interpreting Results /1

- Accuracy of estimation of the failure intensity λ depends on the number of failures experienced (i.e., the sample size).
- Good results in estimating failure intensity are generally experienced for programs with 5,000 or more developed source lines.
- Satisfactory results are obtained for programs with 1,000 or more developed source lines.

How to Handle Defects?

- Table below gives the time between failures for a software system:

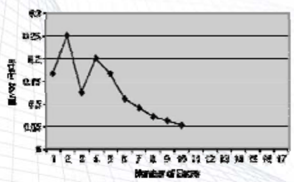
Error no.	1	2	3	4	5	6	7	8	9	10
Time since last failure (hours)	6	4	8	5	6	9	11	14	16	19

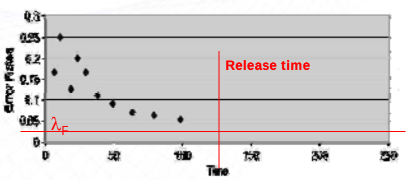
- What can we learn from this data?
  - system reliability?
  - total number of errors in the system?
  - time to (approximately) remove all errors?

What to Learn from Data?

- The inverses of the inter-error times are the failure intensity data points (or error rate if plotted against time)

Error no.	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Time since last failure (hours)	*6*	*4*	*8*	*5*	*6*	*9*	*11*	*14*	*16*	*19*
Failure intensity	*0.166*	*0.25*	*0.125*	*0.20*	*0.166*	*0.111*	*0.09*	*0.071*	*0.062*	*0.053*





Interpreting Results /2

- When the failure intensity λ is very large and the trend indicates little chance of achieving the λ_F by the scheduled release date, what can be done?
  - Adding additional test and debugging resources
  - Adjusting the balance among the objectives for failure intensity, development time, and development cost
  - Deferring features

Conclusions

- CASRE is a valuable tool for software reliability estimation.
- CASRE is easy to learn and use. It is possible to become proficient at the software in a few hours.
- For accuracy in the calculated results, CASRE should be applied to projects where the expected number of failures is greater than 40 to 50 failures.