An Open Source Software Defect Estimation Tool (SweET)- User's Manual

Lance Fiondella
Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth
e-mail: lfiondella@umassd.edu.

August 22, 2018

Executive Summary

Engineered systems increasingly depend on software. As a result, system and software engineers require efficient methods to track defect identification and removal efforts during the software development lifecycle. To support such activities, we have developed a free and open source version of the SoftWare Error Estimation Program (SWEEP), named SweET (Software Defect Estimation Tool), which has not been publicly available to the software engineering community for several years. SWEEPs four modes have been simplified and combined into three modes namely, (i) time-based, (ii) phase-based, and (iii) defect insertion in SweET. Moreover, SweET uses the Weibull model, which is more flexible than the Rayleigh model included in SWEEP. Furthermore, the model fitting performed with least squares estimation in SWEEP has been replaced with an expectation conditional maximization algorithm, which is both stable and efficient.

Contents

1 Introduction **2**

2 Installation and Launching SweET **3**

3 SweET Graphical User Interface **4**
3.1 Modes of SweET 4
3.2 Input data format 5
3.2.1 Data Upload 7
3.3 Mode A - Time based model 8
3.4 Mode B - Phase based model 12
3.5 Mode C - Defect insertion model 13

4 Data saving options **18**

5 Mathematical Model **20**
5.1 Generalization of Weibull to Rayleigh distribution 20
5.2 Comparison of Rayleigh and Weibull model 20
5.3 Parameter Estimation 22

List of Figures

1 Options to manually enter or modify the input data in each mode 6
2 Default tab view of the SweET tool 7
3 Data Upload 7
4 Mode A computation - Progress 8
5 Mode A results 9
6 Mode A - Incidence Curve 10
7 Mode A - Time based model output data sheet 10
8 Mode A - Estimated defects 11
9 Mode B - Initial view 12
10 Mode B - Model fit 13
11 Mode B- Defect discovery profile with 10% tolerance 14
12 Mode C - Initial view 14
13 Mode C - Output options 15
14 Mode C - Inserted and defected detects 15
15 Mode C Inserted/detected/leakage defects by phase 16
16 Mode C - Percentage leakage by phase 16
17 Mode C - Final Computation 17
18 SweET - Data saving options 18
19 Data saving - Configure subplot 18
20 Data saving - Figure Options 19
21 Model fit 21
22 Failure intensity 21

List of Tables

1 Time based data for Mode A 5
2 Phase based data for Mode B 5
3 Normalized phase based data for Mode C 6
4 Comparison of Weibull and Rayleigh model 21

Acknowledgment

This work was supported by the Naval Air Warfare Center (NAVAIR) under contract N00421-16-T-0373.

1 Introduction

Modern commercial and defense systems are increasingly dependent on software for their functionality. Thus, it is important to identify and remove defects during the software development life cycle. Uncorrected defects at various stages of development will propagate to subsequent stages and may lead to system failure once discovered. Therefore, the SoftWare Error Estimation Program (SWEEP) [1, 2] was developed to quantify the effectiveness of software testing. The original SWEEP implemented a Rayleigh model [3] and fit that model with the method of least squares estimation (LSE) [4] to estimate the model parameters. Another major disadvantage of the SWEEP tool is that it is no longer accessible to the software engineering community.

This document describes how to use the free and open source implementation of SWEEP named Software Defect Estimation Tool (SweET), which implements the more flexible Weibull non-homogeneous Poisson process (NHPP) software reliability growth model (SRGM) [5] in place of the Rayleigh model included in SWEEP. Instead of using LSE, the method of maximum likelihood estimation (MLE) [6] is implemented through a procedure known as the expectation conditional maximization [7] algorithm, which is especially suitable for automated tools because it is both computationally efficient and stable. Stability ensures that the algorithm can identify the curve of best fit with no need to provide initial estimates close to the numerical values that best characterize the data. SweET

- Provides all the functionality given in SWEEP.
- SWEEPs four modes have been simplified and combined into three modes in SweET, which provides an intuitive graphical user interface programmed in Python 3.x.
- The free and open source nature of the tool will allow organizations to ensure the reliability of their software as well as to incorporate useful enhancements from which the broader software engineering community can benefit.

SweET runs under the Python programming framework and can be used on computers running Windows, Mac OS X, or Linux. Please report any issues to lfiondella@umassd.edu.

2 Installation and Launching SweET

An automated installation script is available on the Github repository at https://github.com/LanceFiondella/SweET. The user can run this file at the command prompt to install python using the command **pip install -r requirements.txt** along with required packages to run the SweET.

For manual installation, the necessary steps are as follows:

- Install Python 3.x and pyQt5 package available at from https://www.python.org/downloads/ or https://anaconda.org/anaconda/python. Unix/Linux and iOS based machines may already have python installed. Type 'python' in the command prompt to see the python version available if it is already installed on your machine.
- Install the following packages by typing **pip3 install packagename**. This command should be typed in the terminal on Unix/iOS and on command prompt such as Anaconda on Windows.

  1. matplotlib
  2. math
  3. scipy
  4. numpy

  Note: Anaconda includes most of the libraries listed above.

Once python is successfully installed on your machine, download the source code from the Github repository from https://github.com/LanceFiondella/SweET and save the folder in a desired location on your computer. Navigate to the saved folder using the command prompt and type 'python start.py' to launch the application.

3 SweET Graphical User Interface

This section provides a detailed description of the SweET user interface considering a example input data.

3.1 Modes of SweET

SweET possesses three modes which contain the functionality of the four modes included in SWEEP. They are

1. **MODE A - Time-based model:** accepts a failure count data set and provides predictions, including the number of defects to be discovered in a specified number of intervals or the number of intervals required to expose a specified percentage of defects.

2. **MODE B - Phase-based model:** accepts the number of defects discovered per phase and assists in planning by computing the defect discovery profile with user specified upper and lower tolerance levels along with the projected number of latent defects and efficiency of the defect discovery process (EFV).

3. **MODE C - Defect insertion model:** accepts similar or normalized data like Mode B and computes quality metrics to estimate the defects that escape to successive phases and to identify the phases with highest defect detection efficiency. Leakage refers to the number of defects or faults that go undiscovered during a phase.

3.2 Input data format

SweET accepts input data in spreadsheet format. The user can provide input data for all three modes in a single file. Data for different modes must be provided on separate sheets of the spreadsheet with names Mode A, Mode B, and Mode C respectively. The header of each column of the data set should follow the same name as given in the example data file [1] and the sheets in the excel file should be named as **Mode A, Mode B, and Mode C** respectively. The sample data set is taken from a historical project

Table 1 shows the time based data format used for model fitting in Mode A, defects discovered during 33 months of testing.

Table 1: Time based data for Mode A

Time	Defect	Time	Defect	Time	Defect	Time	Defect
1	1	11	41	21	51	31	3
2	0	12	71	22	51	32	4
3	1	13	77	23	30	33	15
4	15	14	80	24	29		
5	15	15	80	25	31		
6	32	16	42	26	20		
7	29	17	60	27	31		
8	45	18	92	28	30		
9	34	19	31	29	7		
10	67	20	68	30	15		

In Table 1, the column names "Time" and "Defect" indicates the testing time and number of defects or failures discovered during the corresponding time interval respectively. The column names should not be changed in order to be compatible with SweET input data requirements.

Table 2 shows Mode B input data collected at various phases of the software development life cycle.

Table 2: Phase based data for Mode B

Phase Name	Data Points
Preliminary Design	20
Detailed Design	21
Coding	22
Unit Test	24
Integration Test	25
System Test	26

Table 2 lists the names of six different phases of software development and testing and the corresponding number of data points. The number of phases are not restricted however the column headings should be the same as shown in Table 2.

Table 3 shows the normalized phase based data used for analysis in Mode C collected during software development effort.

Table 3: Normalized phase based data for Mode C

Phase Name	Defects Detected
HLD	10.2
LLD	6.32
Code	7.48
Unit	6.27
IntTest	4.07
SysTest	2.11

SWeET supports manual addition or modification of the input data using the options shown in Figure 1. These options are available for all three modes of SweET.

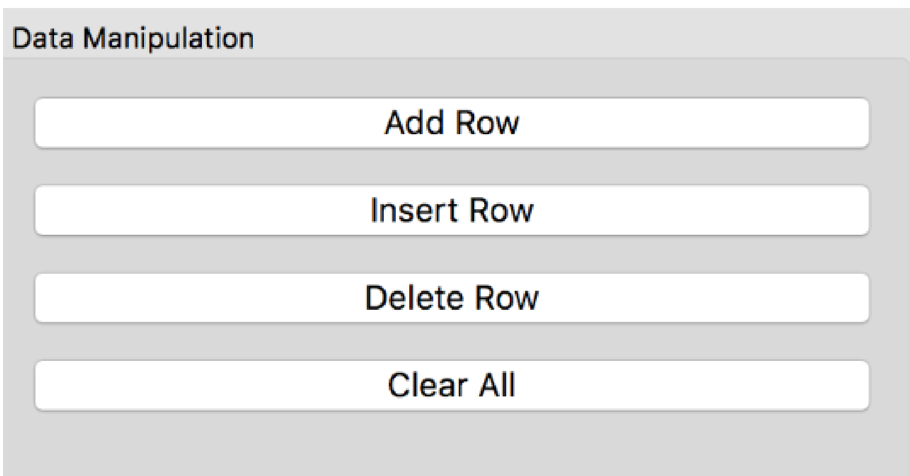


Figure 1: Options to manually enter or modify the input data in each mode

3.2.1 Data Upload

To begin using SweET, open the tool by following the instructions in Section 2. Figure 2 shows the default tab view of the SweET tool.

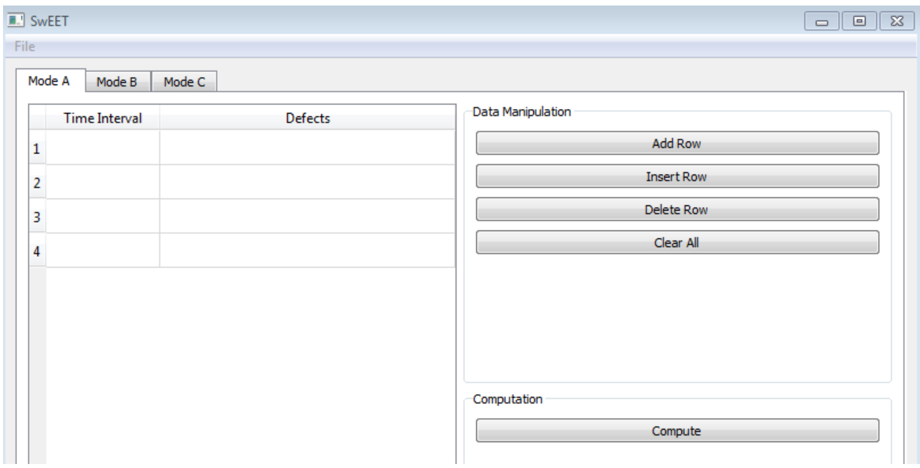


Figure 2: Default tab view of the SweET tool

The thick black rectangle in Figure 2, indicates where the user can switch between the three modes. The user can switch between modes simply by clicking on the tab associated with that mode. Data manipulation options are provided in the right hand side, which allow the user to manually add or modify data without having to modify or reload the input file directly. Below the data manipulation option is the **Computation** section to initiate the calculations for the present mode on the data. Depending on the mode, the user may be prompted to provide additional input in this section before the computations can be performed.

After the input data file is formatted according to the instructions provided in Section 3.2, the user can upload the data by selecting **Open Project** under the **File** menu. Figure 3 shows the data upload option in SweET.

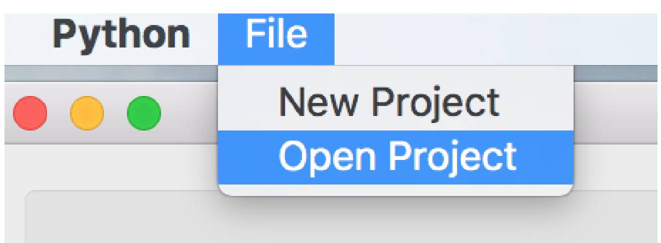


Figure 3: Data Upload

Clicking on the **Open Project** option allows the user to browse through their file systems and upload. Upon successfully uploading the data file, data from different sheets will appear on the corresponding modes.

3.3 Mode A - Time based model

Figure 4 shows a screenshot of Mode A after loading data and executing the computation.

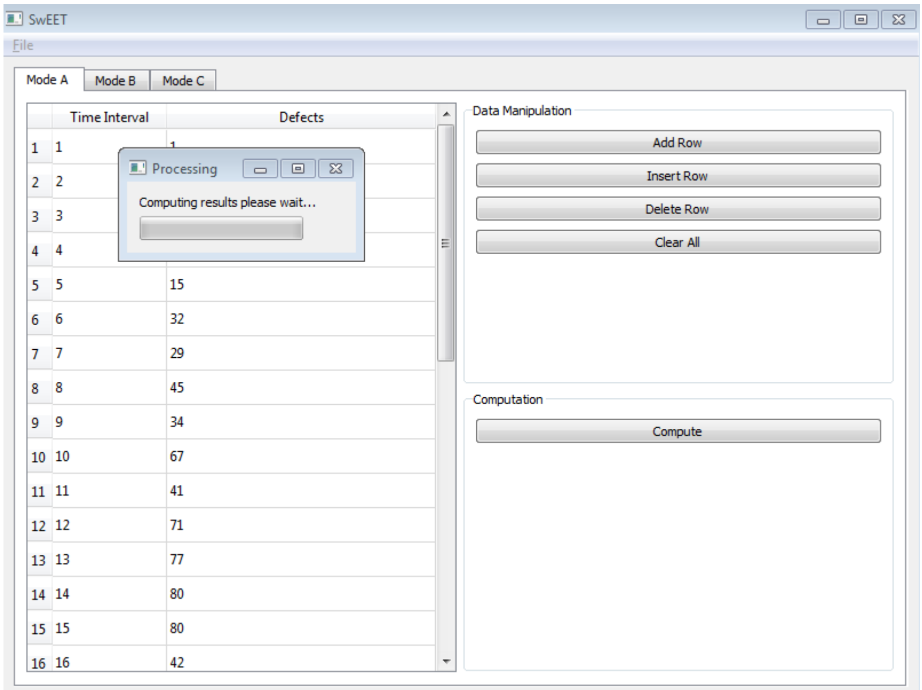


Figure 4: Mode A computation - Progress

Figure 4 shows a progress bar which indicates that the computation is being performed. Once the results are available, they are displayed in Figure 5, which displays the cumulative curve of the Weibull model fit and empirical failure data.
The top of the Figure 5 displays the

- **Defects Discovered to date**, which is the sum of the number of defects entered in the data set.
- **Total defects projected**, which reports the estimated number of defects injected in the software.
- **Percentage of projected defects found to date**, which is simply the percentage of total estimated defects that would be discovered through the number of intervals provided as input.

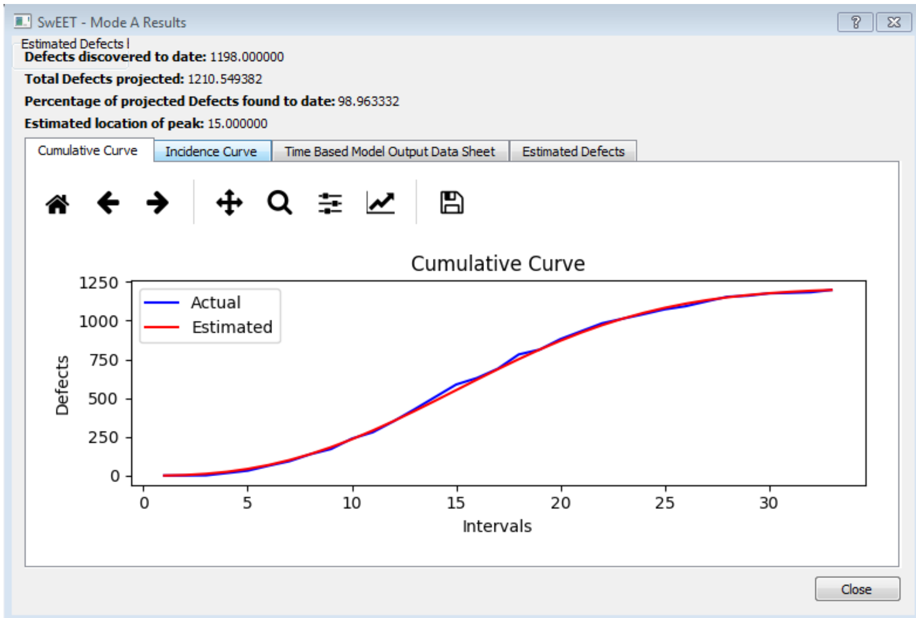


Figure 5: Mode A results

- **Estimated location of peak**, location of the peak in the fitted curve identified in the incidence curve.

Below these estimations are four tabs to display

1. **Cumulative curve**: shows how well the number of failures estimated by the Weibull mean value function matches the observed failure data.
2. **Incidence curve**: the failure intensity curve with the estimated model parameters and the observed data.
3. **Time-Based Model Output Data Sheet**: displays the estimated and observed values for each interval as well as several additional statistics for each interval.
4. **Estimated Defects**: estimates defects in specified number of future intervals or intervals to expose a specified percentage of defects.

The curve in Figure 4 shows the model fit of the input data and the estimated data using the Weibull model. In Figure 4, the blue curve indicates the cumulative observed data (Table 1) and the red curve indicates the fitted data when the Weibull SRGM is applied. The curve closely matches, suggesting that the Weibull model fits the data well and hence can be used to make predictions about the software.

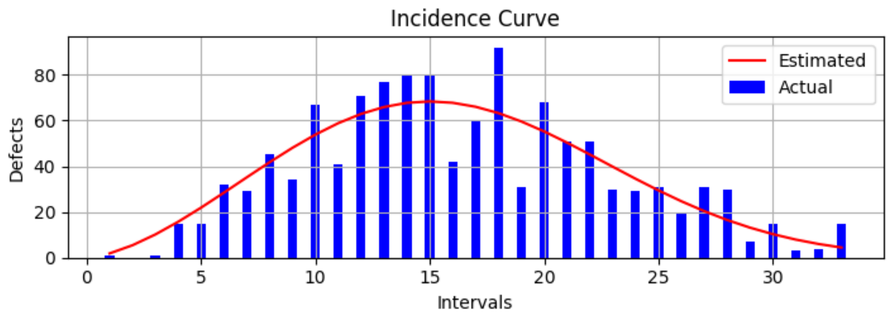
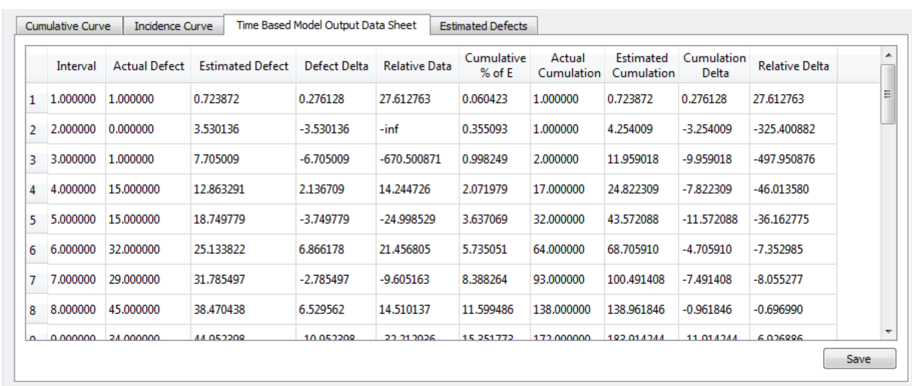


Figure 6: Mode A - Incidence Curve

Clicking the Incidence Curve tab of Figure 4 produces the plot shown in Figure 6.
Figure 6 displays the number of defects found in each interval as well as the corresponding curve of best fit for the Weibull failure intensity.

Clicking the third tab of Figure 5 labelled Time Based Model Output Data Sheet displays the table in Figure 7.



	Interval	Actual Defect	Estimated Defect	Defect Delta	Relative Data	Cumulative % of E	Actual Cumulation	Estimated Cumulation	Cumulation Delta	Relative Delta
1	1.000000	1.000000	0.723872	0.276128	27.612763	0.060423	1.000000	0.723872	0.276128	27.612763
2	2.000000	0.000000	3.530136	-3.530136	-inf	0.355093	1.000000	4.254009	-3.254009	-325.400882
3	3.000000	1.000000	7.705009	-6.705009	-670.500871	0.998249	2.000000	11.959018	-9.959018	-497.950876
4	4.000000	15.000000	12.863291	2.136709	14.244726	2.071979	17.000000	24.822309	-7.822309	-46.013580
5	5.000000	15.000000	18.749779	-3.749779	-24.998529	3.637069	32.000000	43.572088	-11.572088	-36.162775
6	6.000000	32.000000	25.133822	6.866178	21.456805	5.735051	64.000000	68.705910	-4.705910	-7.352985
7	7.000000	29.000000	31.785497	-2.785497	-9.605163	8.388264	93.000000	100.491408	-7.491408	-8.055277
8	8.000000	45.000000	38.470438	6.529562	14.510137	11.599486	138.000000	138.961846	-0.961846	-0.696990

Figure 7: Mode A - Time based model output data sheet

Figure 7 shows the actual and estimated defects for each interval as well as the error and relative error between these values. The first two columns in Figure 7 is the input data and the other columns include

- **Estimated Defect** - Number of defects estimated by the Weibull SRGM.
- **Defect Delta** - Difference between the input and the estimated data.
- **Relative Data** - Difference between the defect delta and the actual defect in column 2, divided by the actual defect expressed in %.

- **Cumulative % of E** - Estimated percentage of Total Defects projected through n intervals.
- **Actual Cumulation** - Sum of the input data in column 2.
- **Estimated Cumulation** - Sum of the estimated data.
- **Cumulation Delta** - Difference between the actual and estimated cumulative values.
- **Relative Delta** - Relative difference between the Cumulative Delta and the Actual Cumulation, expressed in %.

Finally, clicking the fourth tab Estimated Defects produces two sets of outputs shown in Figure 8.

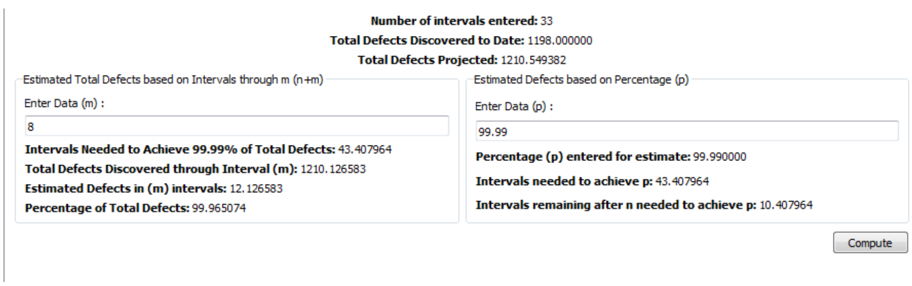


Figure 8: Mode A - Estimated defects

In Figure 8, the user enters a specified number of future intervals (day, week, month) to obtain: **Intervals Needed to achieve 99.99% of Total Defects**, which includes the intervals observed already. **Total Defects Discovered through Interval (m)** is the estimated total defects that would be discovered after m additional intervals, including those already observed. **Estimated Defects in (m) intervals** is the difference between the estimated number of faults after m additional intervals and the number of faults observed so far. **Percentage of Total Defects** is the estimate of the fraction of defects that would be discovered after m additional intervals.

Similarly, the user enters a specified percentage of defects (p) in Figure 8 to obtain: **Intervals needed to achieve p**, which is the total number of intervals required to discover p percent of the defects. **Intervals remaining after n needed to achieve p** is the difference between the estimated number of intervals needed to achieve p and the number of intervals observed so far.

3.4 Mode B - Phase based model

The left side of Figure 9 shows the input data from the spreadsheet for Mode B.

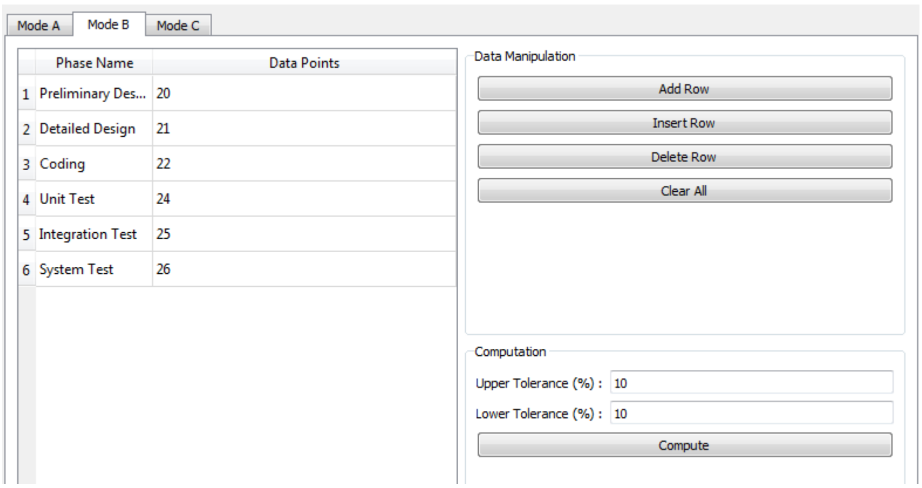


Figure 9: Mode B - Initial view

In Figure 9, phases are named intervals in a process defined by an organization. The top right provides a data manipulation options similar to those for Mode A. The computation section located in the bottom right allows the user to specify the upper and lower tolerance levels as a percentage to view the defect discovery profile. This mode generates the two plots shown in Figures 10 and 11.

Figure 10 provides a bar chart of the actual (blue) and estimated (red) defects per phase.
Figure 10 indicates that the estimated data matches the actual data well.

Figure 11 shows the defect discovery profile with the 10% upper and lower tolerance levels specified in Figure 9.

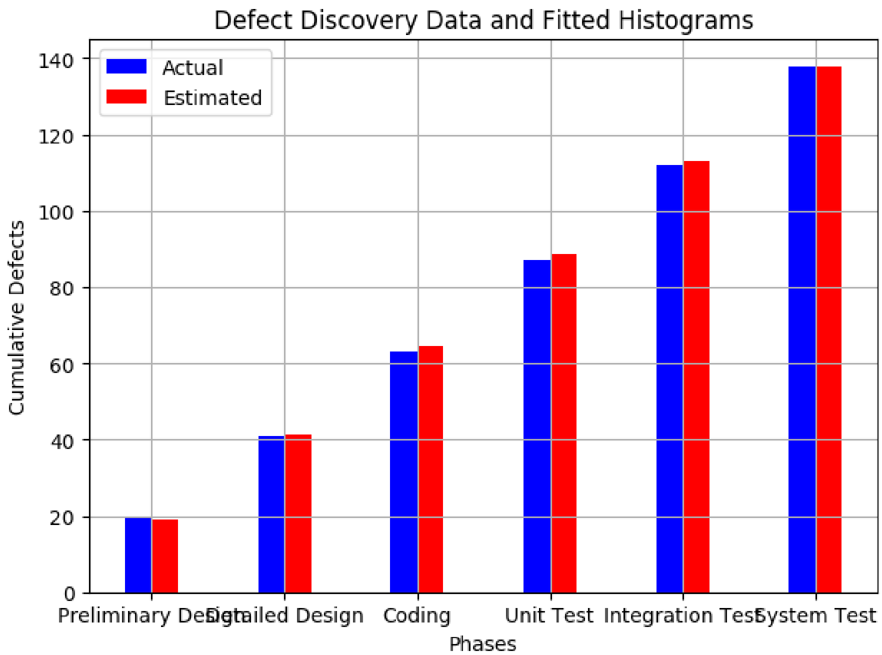


Figure 10: Mode B - Model fit

3.5 Mode C - Defect insertion model

Unlike the first three modes, Mode C is an analytical model that uses the defect discovery profile to estimate the defect injection profile and the defect leakage rate between phases.

Figure 12 shows the initial view of Mode C with data loaded from the spreadsheet.

The computation section on the right side of Figure 12 provides an input box to specify the latent defects, which are the defects present in the system that cause failures only when a certain set of conditions are met during operations. This can be estimated by calculating the difference between the total defects projected and defects discovered to date. Clicking compute generates a result window with four tabs to display plots and tables as shown in Figure 13. The four output options in Figure 13 includes

1. **Defects/KSLOC Inserted and Detected in Phase**: 1. Defect insertion refers to the number of defects that were introduced, which may be detected in the same or a later phase. Defects are usually tracked according to their density per thousand source lines of code (KSLOC). This tab shows the predicted number of undetected defects from observed data and the actual number discovered in each phase per KSLOC. Users are advised to measure effective KSLOC, which considers whether code is new

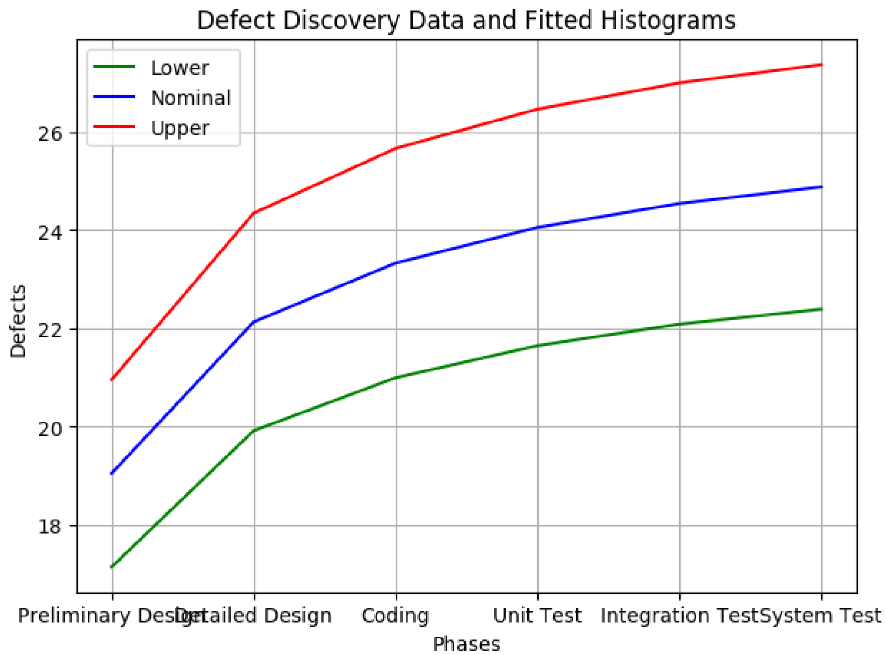


Figure 11: Mode B- Defect discovery profile with 10% tolerance

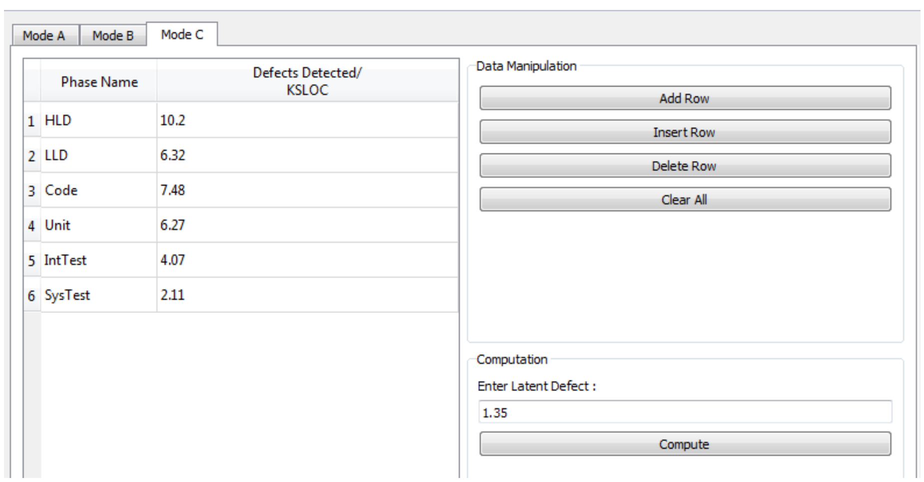


Figure 12: Mode C - Initial view

or reused.

2. **Phase Insertion/Detection/Leakage**: This tab reports the defects in-

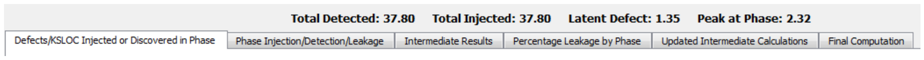


Figure 13: Mode C - Output options

serted, detected, and leaked in each phase.

3. **Percentage Leakage by Phase**: This tab reports the percentage of defects leaked in each phase. This tab also reports the efficiency and effectiveness of defect detection.

4. **Final Computation**: This tab reports software quality metrics based on the input data. This tab also shows the fault insertion matrix, which is used to compute the metrics.

Figure 14 shows the first tab with a bar chart displaying the inserted and discovered defects per KSLOC by phase.

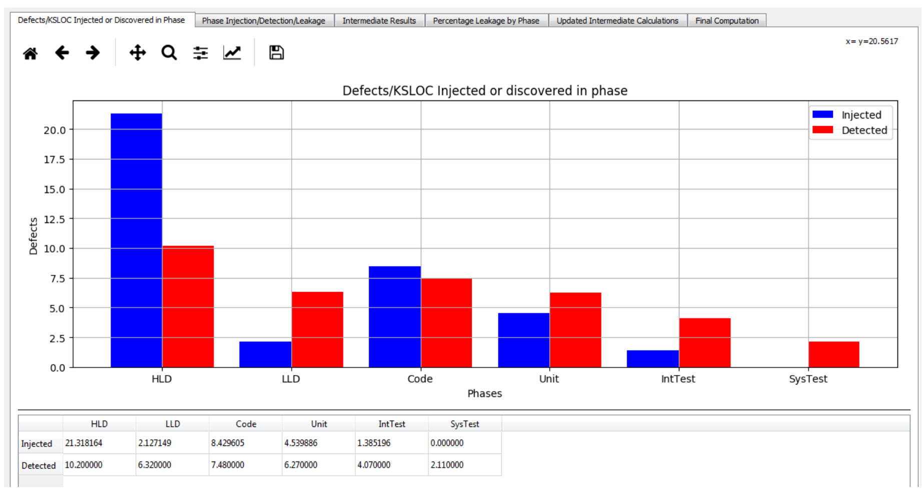


Figure 14: Mode C - Inserted and defected detects

Figure 15 displays the inserted, detected, and leaked defects by phase.

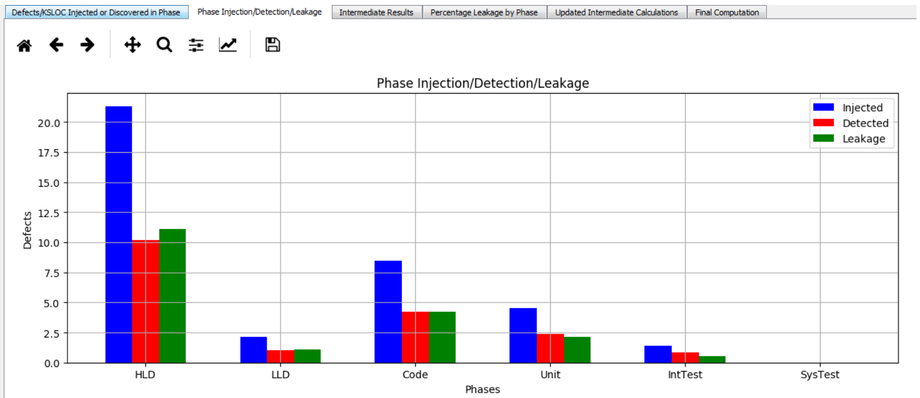
Figure 16 shows the third tab in which a bar chart indicates percentage leakage by phase and the efficiency and effectiveness below the chart.

The last tab is shown in Figure 17. It displays the quality metrics estimated in Mode C and the fault insertion matrix.

In Figure 17, the list of quality metrics computed in Figure 17 include:

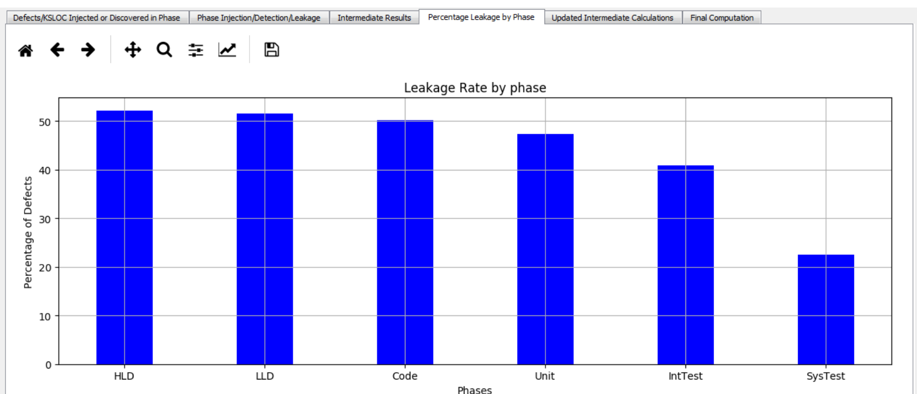
- **Overall Defect Discovery Efficiency**: Percentage of defects discovered in software development, ideally 100%.

- **Initial Average Phase Defect Discovery Efficiency**: Percentage of defects inserted and discovered during software development, ideally 100%.



	HLD	LLD	Code	Unit	IntTest	SysTest
Injected	21.318164	2.127149	8.429605	4.539886	1.385196	0.000000
Detected	10.200000	1.032039	4.203538	2.392162	0.819459	0.000000
Leakage	11.118164	1.095110	4.226067	2.147724	0.565737	0.000000

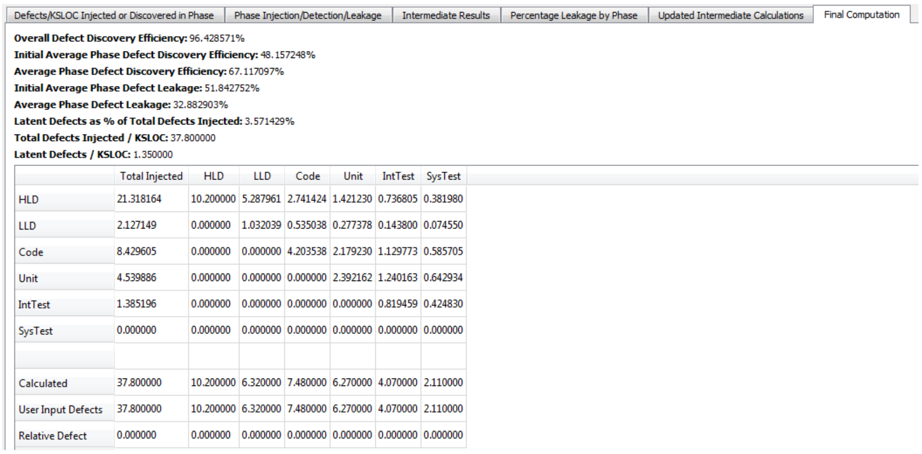
Figure 15: Mode C Inserted/detected/leakage defects by phase



	HLD	LLD	Code	Unit	IntTest	SysTest
Percentage of Defects	52.153479	51.482507	50.133627	47.307884	40.841625	22.495392
Efficiency	47.846521	48.517493	49.866373	52.692116	59.158375	77.504608
Effectiveness	47.846521	47.714991	48.714035	50.504221	54.050465	60.982659

Figure 16: Mode C - Percentage leakage by phase

- **Average Phase Defect Discovery Efficiency**: Average number of defects inserted and discovered in each phase, ideally 100%.
- **Initial Average Phase Defect Leakage**: Percentage of defects inserted that were undetected during the development phase, ideally 0%.
- **Average Phase Defect Leakage**: Average number of defects inserted in a given phase that are not discovered in the same phase, ideally 0%.



	Total Injected	HLD	LLD	Code	Unit	IntTest	SysTest
HLD	21.318164	10.200000	5.287961	2.741424	1.421230	0.736805	0.381980
LLD	2.127149	0.000000	1.032039	0.535038	0.277378	0.143800	0.074550
Code	8.429605	0.000000	0.000000	4.203538	2.179230	1.129773	0.585705
Unit	4.539886	0.000000	0.000000	0.000000	2.392162	1.240163	0.642934
IntTest	1.385196	0.000000	0.000000	0.000000	0.000000	0.819459	0.424830
SysTest	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Calculated	37.800000	10.200000	6.320000	7.480000	6.270000	4.070000	2.110000
User Input Defects	37.800000	10.200000	6.320000	7.480000	6.270000	4.070000	2.110000
Relative Defect	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Figure 17: Mode C - Final Computation

- **Latent Defects as % of Total Defects Inserted**: Percentage of defects that are not discovered during software development. Ideally, it should be 0%.
- **Total defects inserted/KSLOC**: Total number of defects per KSLOC inserted into the software during development.
- **Latent defects/KSLOC**: Number of latent defects per KSLOC inserted but undetected during development, ideally 0.

4 Data saving options

SweET has data saving options associated with each mode. The tables can be saved in .csv format, whereas the plots can be saved in .png, .pdf, and other formats.

Figure 18 shows more options associated with each mode. In Figure 18, is

Figure 18: SweET - Data saving options

associated with eight options to modify or save the figure. The options include:

- **Home button**: Restores the original figure if the user has made some modifications.
- **Backward and Forward arrows**: Allows the user to navigate one step to the previous or next modifications that they might have done on the figure.
- **Pan/Zoom**: This allows the user to pan axes by clicking left mouse button and enlarging the area selected by then clicking the right mouse button.
- **Zoom**: Allows the user to zoom into the selected area. To restore the figure, the user can either use the home button or forward/backward arrows.
- **Configure Subplots**: Allows the user to set the figure layout, define the borders as well as spacing as shown in Figure 19 below:

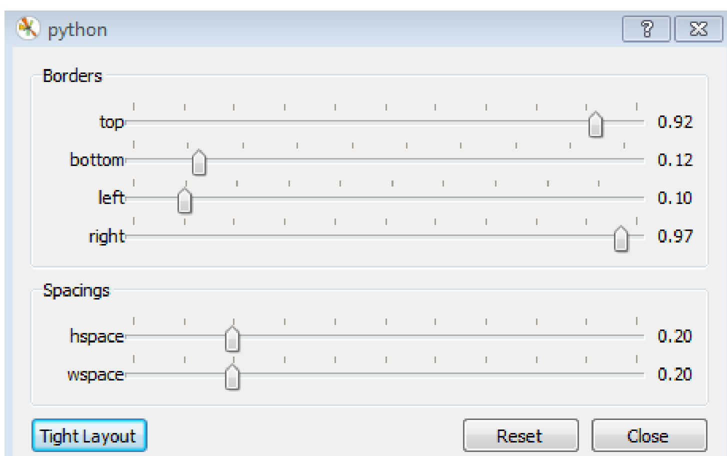


Figure 19: Data saving - Configure subplot

- **Figure Options**: This enables defining the axes range, changing the title and axes label, curves label, legends, line, and marker styles as shown in Figure 20 below:

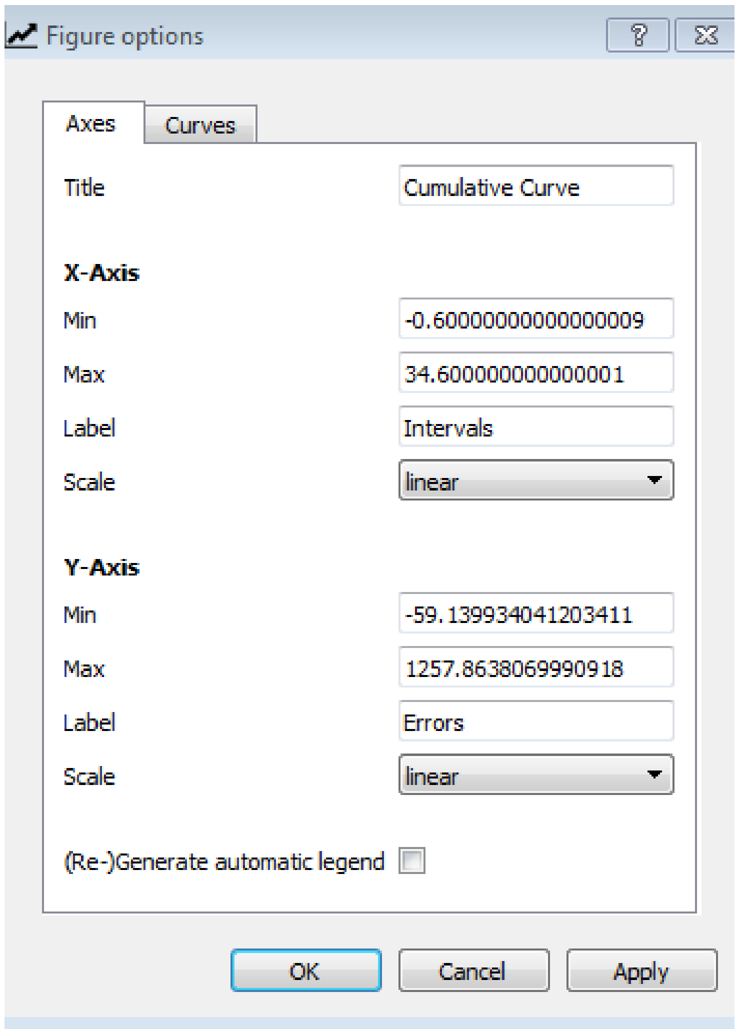


Figure 20: Data saving - Figure Options

- **Save the Figure**: Allows the user to save the data plot in the desired format including .png, .pdf, and .eps
- On the right corner after all these options, the x- and y-coordinates are displayed based on the cursor.

5 Mathematical Model

This section provides a detailed description of the mathematical modeling and estimation associated with each of the four models.

5.1 Generalization of Weibull to Rayleigh distribution

The original SWEEP tool applies Rayleigh model to identify the estimates and predictions based on least squares estimation (LSE) [8] method. However, LSE is not efficient for non-linear models and moreover, the Rayleigh model does not necessarily handle the missing data well during model fitting and estimation. Therefore, the open source version of the SWEEP tool implements the Weibull NHPP SRGM, estimating the parameters with the method of maximum likelihood . In particular, we employ the expectation conditional maximization (ECM) algorithm [9, 7].

The cumulative distribution function (CDF) of the Weibull is

$$F(t) = 1 - e^{-(\frac{t}{\lambda})^c} \tag{1}$$

where λ and c are the scale and shape parameters respectively.

When $c = 2$, the Weibull distribution models a linearly increasing failure rate, which is also known as the Rayleigh distribution. Therefore, setting $c = 2$ and $\lambda = \sqrt{2}\sigma$ in Equation (1) simplifies to

$$\begin{aligned} F(t) &= 1 - e^{-(\frac{t}{\sqrt{2}\sigma})^2} \\ &= 1 - e^{-\frac{t^2}{2\sigma^2}} \end{aligned} \tag{2}$$

where σ denotes the scale parameter.

However, the CDF of the Weibull used in SweET is

$$F(t) = 1 - e^{-bt^c} \tag{3}$$

where b and c are the scale and shape parameters respectively.

Parameters in Equation (3) can be obtained from Equation (1) through the following transformation

$$e^{-(\frac{t}{\lambda})^c} = e^{-(\frac{1}{\lambda})^c t^c} = e^{-bt^c}$$

Therefore,

$$b = \left(\frac{1}{\lambda}\right)^c$$

5.2 Comparison of Rayleigh and Weibull model

This section compares the parameter estimates and model fit of the Rayleigh and Weibull model when applied to the failure count data listed in Table 1.

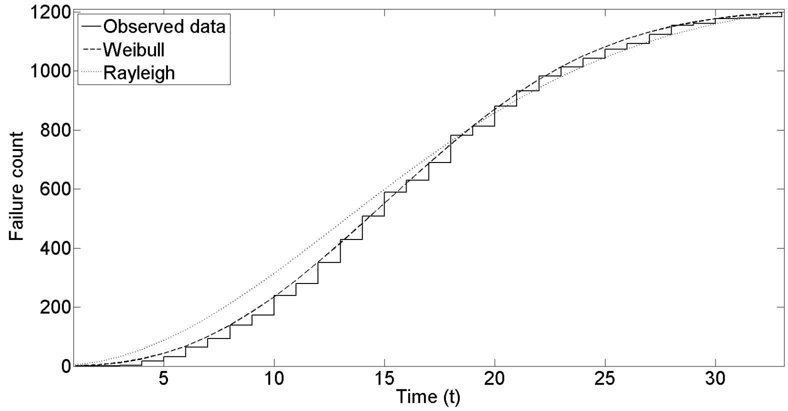


Figure 21: Model fit

Figure 21 shows the model fit of the Weibull and Rayleigh models when applied to Table 1.

Figure 22 shows the failure intensity of the Weibull and Rayleigh models when applied to Table 1.

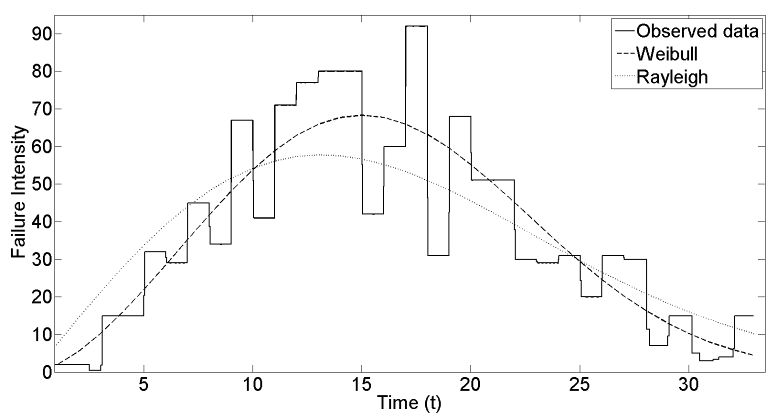


Figure 22: Failure intensity

Table 4: Comparison of Weibull and Rayleigh model

Model name	LL	AIC
Rayleigh	-182.116	368.233
Weibull	-144.206	294.412

5.3 Parameter Estimation

In the SweET tool, we apply the reduced log-likelihood expectation conditional maximization (RLL-ECM) [10] algorithm to identify the maximum likelihood estimates (MLE)of Weibull model parameters. This section describes how to obtain update rules for model parameters by applying the RLL-ECM algorithm.

The mean value function (MVF) of the Weibull model is

$$m(t) = a(1 - e^{-bt^c}) \tag{4}$$

where b and c are the scale and shape parameters respectively.

Given failure count data

$$\langle \mathbf{T}, \mathbf{K} \rangle = \langle (t_1, k_1), (t_2, k_2), \ldots, (t_n, k_n) \rangle,$$

The log-likelihood function of the Weibull SRGM is

$$\begin{aligned} LL(a,b,c|\mathbf{T},\mathbf{K}) &= \sum_{i=1}^{n} k_i \log(a) - a\left(1 - e^{-bt_n^c}\right) \\ &\quad + \sum_{i=1}^{n} k_i \log\left(\frac{e^{-bt_{i-1}^c} - e^{-bt_i^c}}{k_i!}\right) \end{aligned} \tag{5}$$

From the log-likelihood function given in Equation (5), the maximum likelihood estimate of parameter a is

$$\hat{a} = \frac{\sum_{i=1}^{n} k_i}{1 - e^{-bt_n^c}} \tag{6}$$

Substituting Equation (6) into Equation (5) produces the reduced log-likelihood function

$$\begin{aligned} RLL(b,c|\mathbf{T},\mathbf{K}) &= \sum_{i=1}^{n} k_i \log\left(\frac{\sum_{i=1}^{n} k_i}{1 - e^{-bt_n^c}}\right) - \sum_{i=1}^{n} k_i \\ &\quad + \sum_{i=1}^{n} k_i \log\left(\frac{e^{-bt_{i-1}^c} - e^{-bt_i^c}}{k_i!}\right) \end{aligned} \tag{7}$$

Differentiating Equation (7) according to Equation (**??**), the ECM update rules for parameters b and c are

$$b'' = \sum_{i=1}^{n} k_i \frac{t_i^{c'} e^{-b'' t_i^{c'}} - t_{i-1}^{c'} e^{-b'' t_{i-1}^{c'}}}{e^{-b'' t_{i-1}^{c'}} - e^{-b'' t_i^{c'}}} - \sum_{i=1}^{n} k_i \frac{t_n^{c'} e^{-b'' t_n^{c'}}}{1 - e^{-b'' t_n^{c'}}} \tag{8}$$

and

$$
\begin{aligned}
c^{''} \quad = \quad & \sum_{i=1}^{n} k_i \frac{b^{'} \left(t_i^{c^{''}} e^{-b^{'} t_i^{c^{''}}} \log(t_i) - t_{i-1}^{c^{''}} e^{-b^{'} t_{i-1}^{c^{''}}} \log t_{i-1} \right)}{e^{-b^{'} t_{i-1}^{c^{''}}} - e^{-b^{'} t_i^{c^{''}}}} \\
& - \sum_{i=1}^{n} k_i \frac{b^{'} t_n^{c^{''}} e^{-b^{'} t_n^{c^{''}}} \log(t_n)}{1 - e^{-b^{'} t_n^{c^{''}}}} \qquad (9)
\end{aligned}
$$

Note that the CM expressions given in Equations (8) and (9) can be applied in any order. Thus, it is possible to update parameter b in odd iterations and parameter c in even iterations or reverse the order of their application so that parameters c and b are updated in odd and even iterations, respectively.

The initial estimate of the model parameters are

$$
a^{(0)} = n \qquad (10)
$$

and,

$$
b^{(0)} = \frac{n}{\sum_{i=1}^{n} t_i^c} \qquad (11)
$$

Setting $c = 1$ in Equation 11 simplifies the Weibull to the exponential model.

References

[1] E. Jan, "Software Error Estimation Program (SWEEP) - User Manual," 1993.

[2] J. Gaffney and C. Davis, "An approach to estimating software errors and availability."

[3] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes.* Boston, MA, USA: McGraw-Hill, 2014.

[4] D. Kleinbaum, L. Kupper, A. Nizam, and E. Rosenberg, *Applied Regression Analysis and Other Multivariable Methods.* Nelson Education, 2013.

[5] M. Lyu, "Handbook of Software Reliability Engineering," 1996.

[6] R. Hogg, J. McKean, and A. Craig, *Introduction to Mathematical Statistics.(6th edition).* Upper Saddle River, NJ, USA: Pearson Education, 2005.

[7] P. Zeephongsekul, C. Jayasinghe, L. Fiondella, and V. Nagaraju, "Maximum-likelihood estimation of parameters of NHPP software reliability models using expectation conditional maximization algorithm," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1571–1583, 2016.

[8] D. Kleinbaum, L. Kupper, A. Nizam, and K. Muller, *Applied Regression Analysis and Other Multivariable Methods (4th Edition).* Australia: Thomson Learning Inc. Brooks/Cole, 2008.

[9] V. Nagaraju, L. Fiondella, P. Zeephongsekul, C. Jayasinghe, and T. Wandji, "Performance optimized expectation conditional maximization algorithms for nonhomogeneous Poisson process software reliability models," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 722–734, 2017.

[10] V. Nagaraju, A. Krishna Murthy, L. Fiondella, P. Zeephongsekul, and T. Wandji, "Expectation conditional maximization algorithms for failure count non-homogeneous Poisson process software reliability models," in *22nd International Society of Science and Applied Technologies (ISSAT), CA, USA*, 2016.