Lab. Report #5 – Software Reliability Assessment

Group #: 17

Student Names: Matteo Morrone, Adeshpal Virk, Sam Farzamfar, Taimoor Abrar

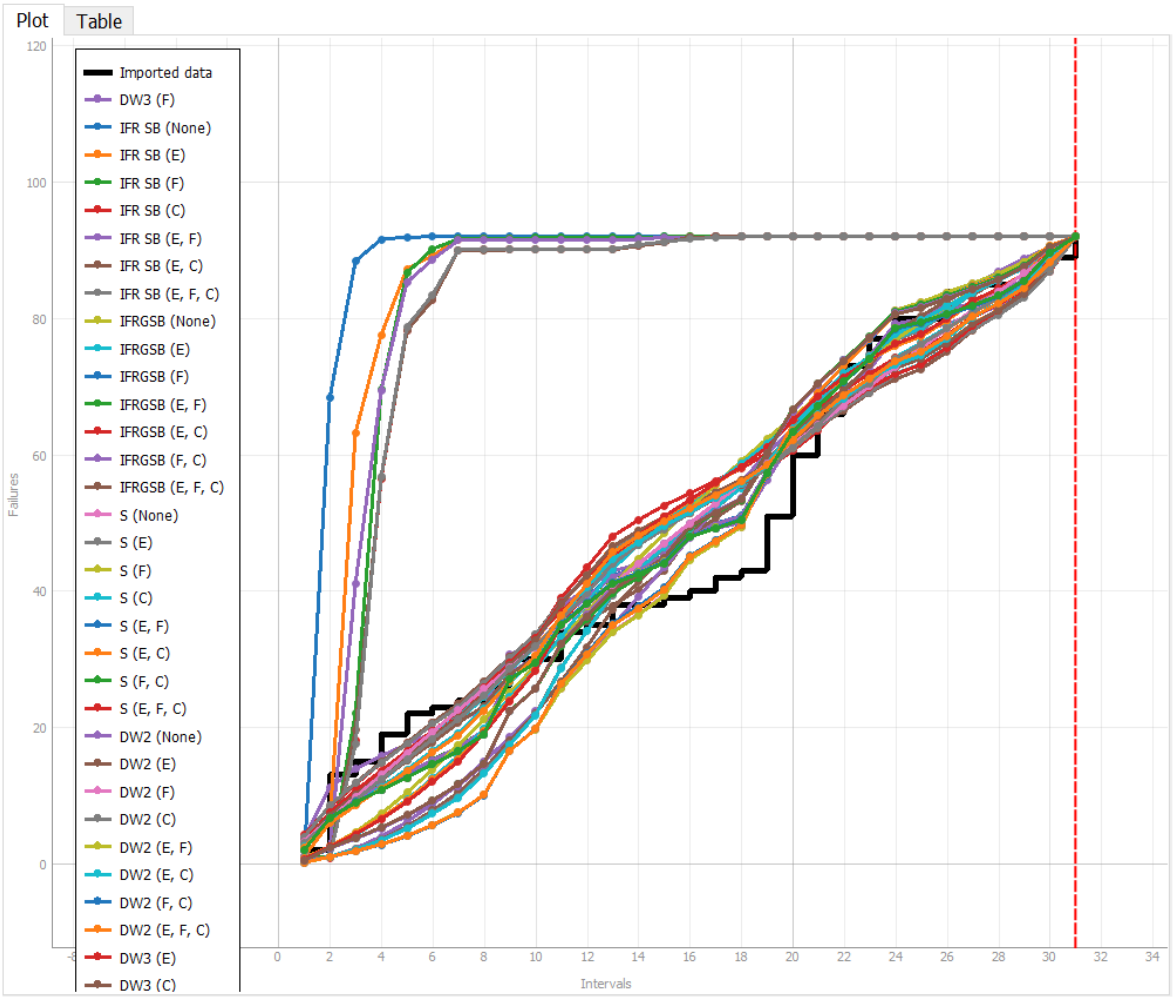
Table of Contents

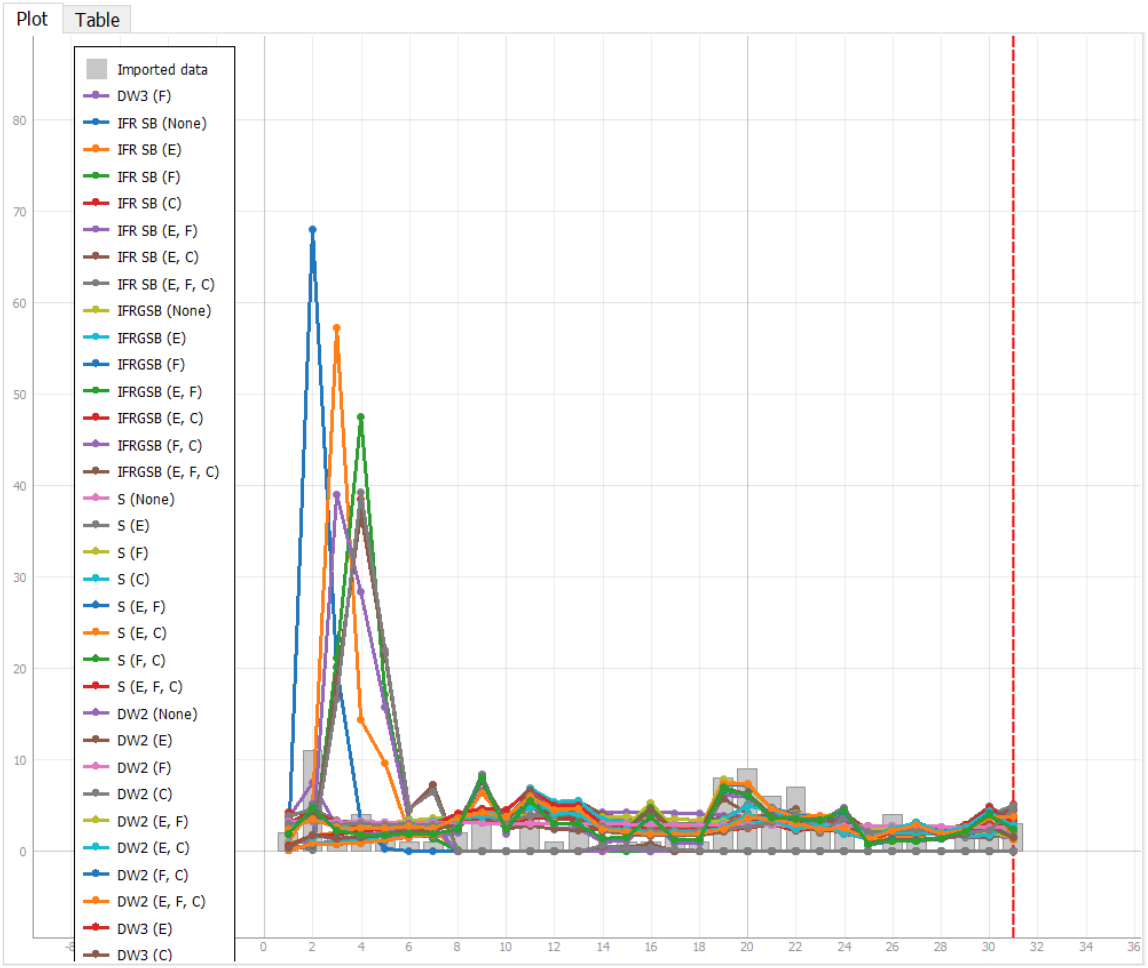
Introduction

In this lab we were tasked with using reliability assessment tools. By looking at it using reliable growth testing and using a reliability Demonstration Chart. By using these tools we can find the failure count and time interval to figure out MTTF, failure rate, and reliability.

Assessment Using Reliability Growth Testing

First we decide that we would plot every possible graph using C-SFRAT which gave us the following plots

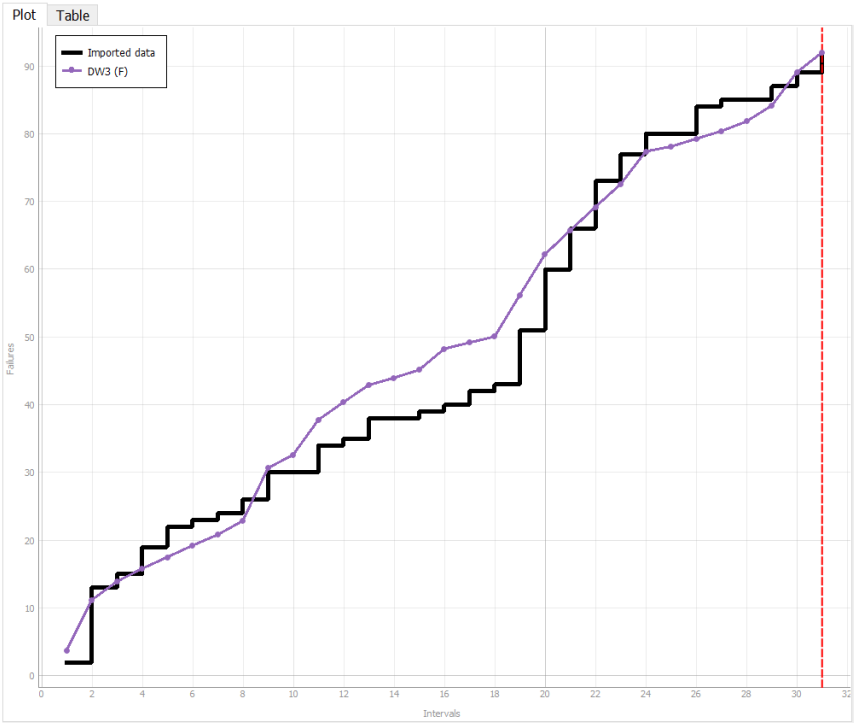


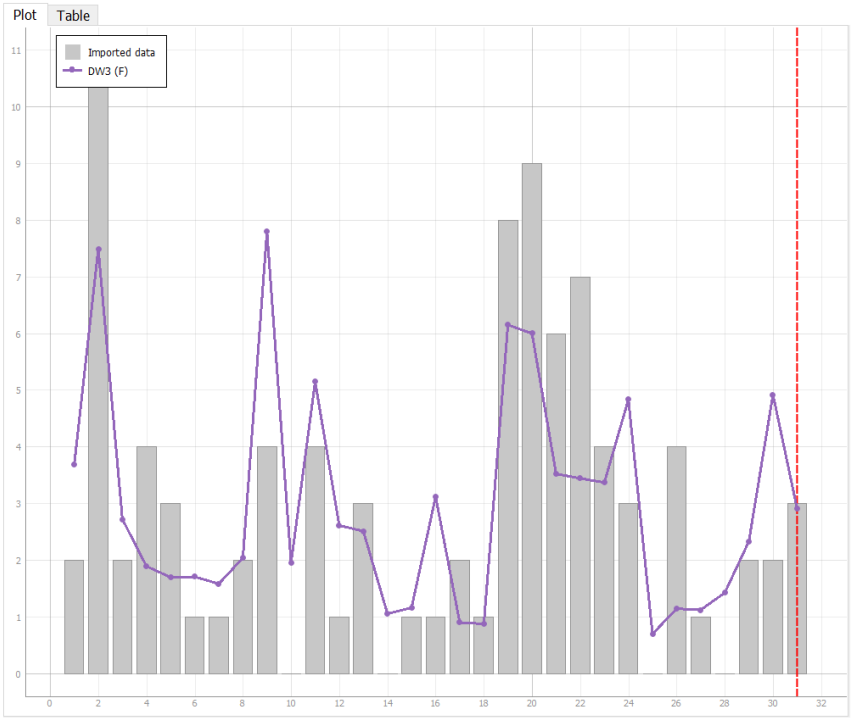

Plot
Table
Imported data
DW3 (F)
IFR SB (None)
IFR SB (E)
IFR SB (F)
IFR SB (C)
IFR SB (E, F)
IFR SB (E, C)
IFR SB (E, F, C)
IFRGSB (None)
IFRGSB (E)
IFRGSB (F)
IFRGSB (E, F)
IFRGSB (E, C)
IFRGSB (F, C)
IFRGSB (E, F, C)
S (None)
S (E)
S (F)
S (C)
S (E, F)
S (E, C)
S (F, C)
S (E, F, C)
DW2 (None)
DW2 (E)
DW2 (F)
DW2 (C)
DW2 (E, F)
DW2 (E, C)
DW2 (F, C)
DW2 (E, F, C)
DW3 (E)
DW3 (C)
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36

Then we looked at the following table to compare our graphs which told us that the DW3 graph with covariate F was our best plot.

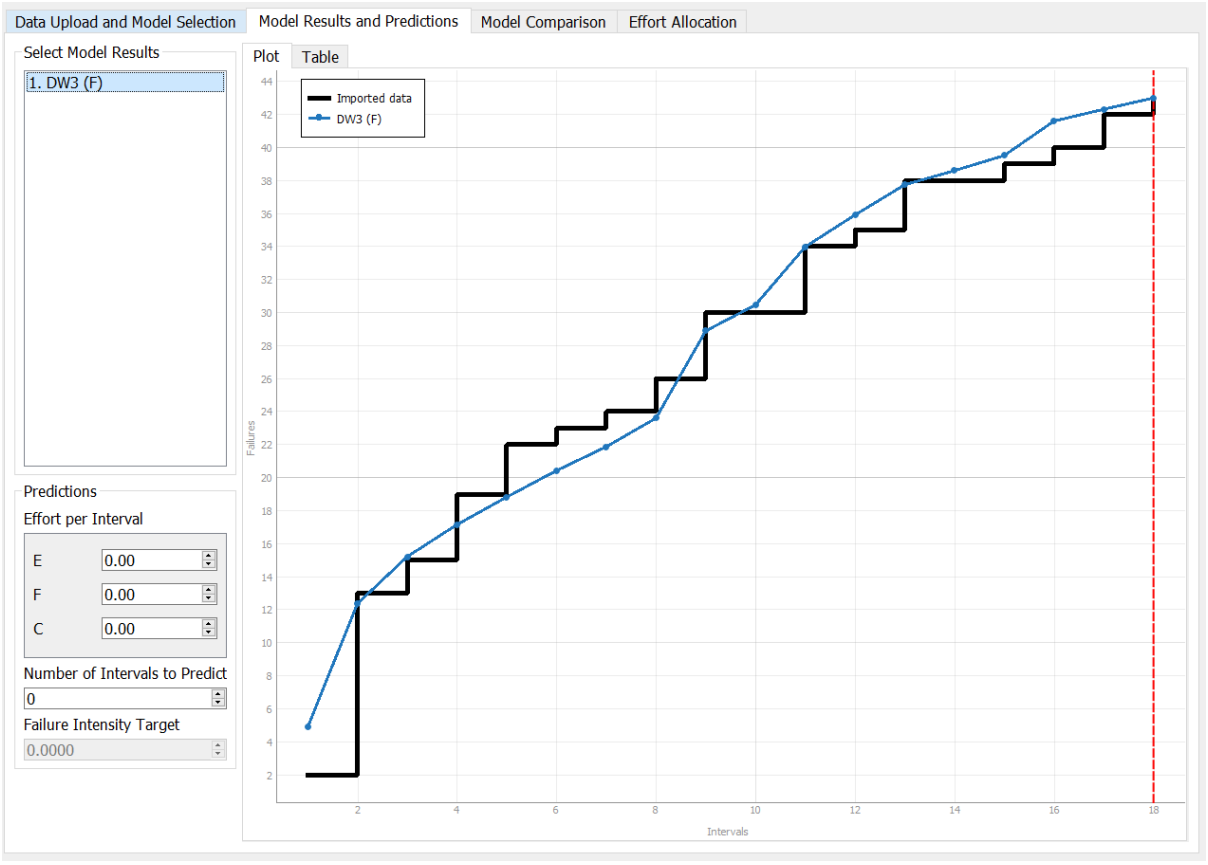
	Model Name	Covariates	Log-Likelihood	AIC	BIC	SSE	PSSE	Critic (Mean)
16	S	E	-72.667	153.334	159.070	1123.852	210.900	0.995
17	S	F	-59.662	127.323	133.059	759.170	92.282	0.999
18	S	C	-71.899	151.797	157.533	1344.723	98.408	0.994
19	S	E, F	-59.661	129.322	136.492	754.722	81.241	0.998
20	S	E, C	-71.204	152.408	159.578	1409.778	563.231	0.994
21	S	F, C	-59.653	129.306	136.476	745.463	90.396	0.998
22	S	E, F, C	-59.653	131.306	139.910	744.495	487.570	0.998
23	DW2	None	-92.571	189.141	192.009	2031.486	154.807	0.987
24	DW2	E	-91.175	188.350	192.652	2138.682	219.452	0.987
25	DW2	F	-83.554	173.107	177.409	2301.161	11.611	0.988
26	DW2	C	-89.074	184.149	188.451	2208.983	7.567	0.987
27	DW2	E, F	-83.551	175.103	180.839	2301.402	4.758	0.988
28	DW2	E, C	-88.985	185.971	191.707	2246.488	32.082	0.987
29	DW2	F, C	-83.360	174.720	180.456	2286.410	6.316	0.988
30	DW2	E, F, C	-83.325	176.649	183.819	2279.096	37.071	0.988
31	DW3	E	-72.088	152.175	157.911	1087.949	7.717	0.995
32	DW3	F	-57.100	122.199	127.935	528.046	16.021	1.000
33	DW3	C	-71.823	151.647	157.383	1307.653	71.783	0.994
34	DW3	E, C	-70.972	151.944	159.114	1349.865	47.321	0.994
35	GM	None	-74.982	153.963	156.831	917.068	87.747	0.995
36	GM	E	-72.700	151.400	155.702	1114.328	811.235	0.995
37	GM	F	-59.662	125.323	129.625	759.655	87.903	0.999
38	GM	C	-71.946	149.892	154.194	1336.197	62.367	0.994
39	GM	E, F	-59.661	127.322	133.058	755.156	102.777	0.999
40	GM	E, C	-71.237	150.474	156.210	1402.576	1320.009	0.994
41	GM	F, C	-59.653	127.306	133.042	745.938	110.422	0.999
42	GM	E, F, C	-59.653	129.306	136.476	744.943	99.513	0.998

With DW3 and covariate F we got the following best fit graphs.

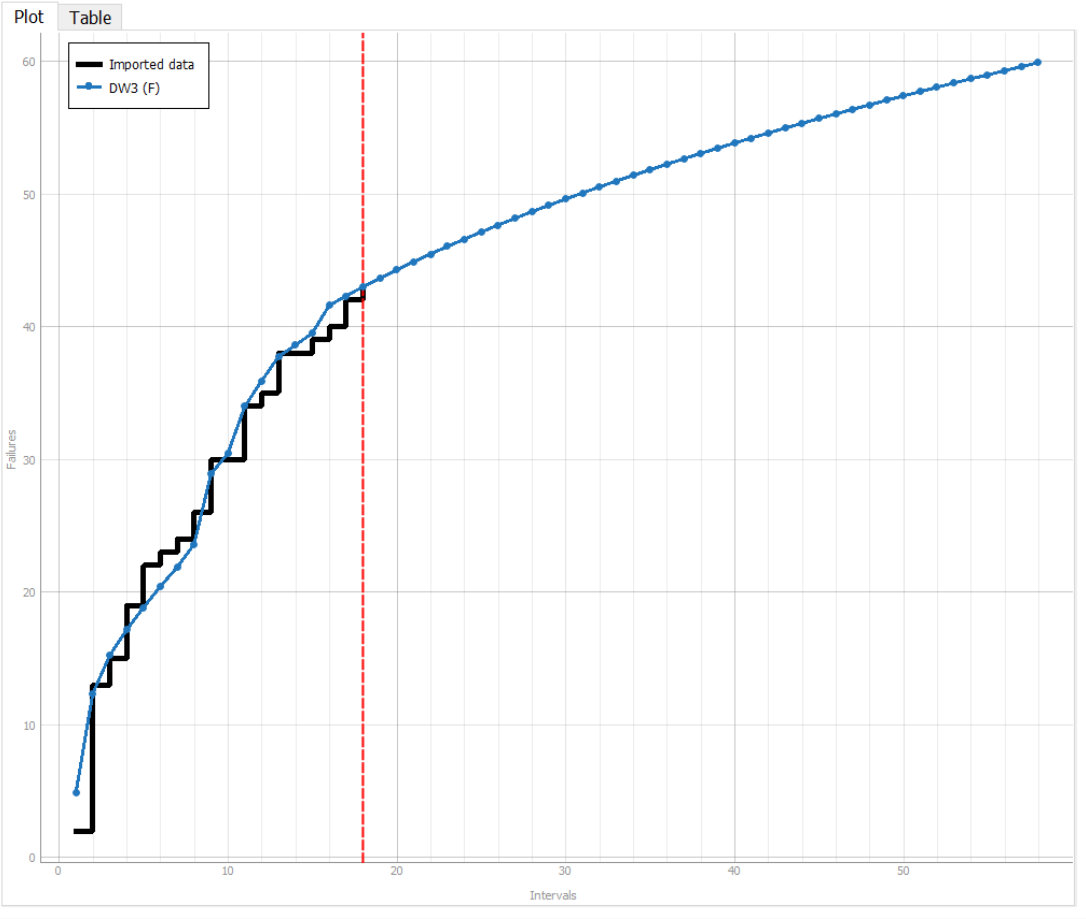




After looking at the above graph and calculating the Laplace we decided that our data should be limited to 18 intervals.



Then we predicted 40 Intervals and saw that it is steadily declining.



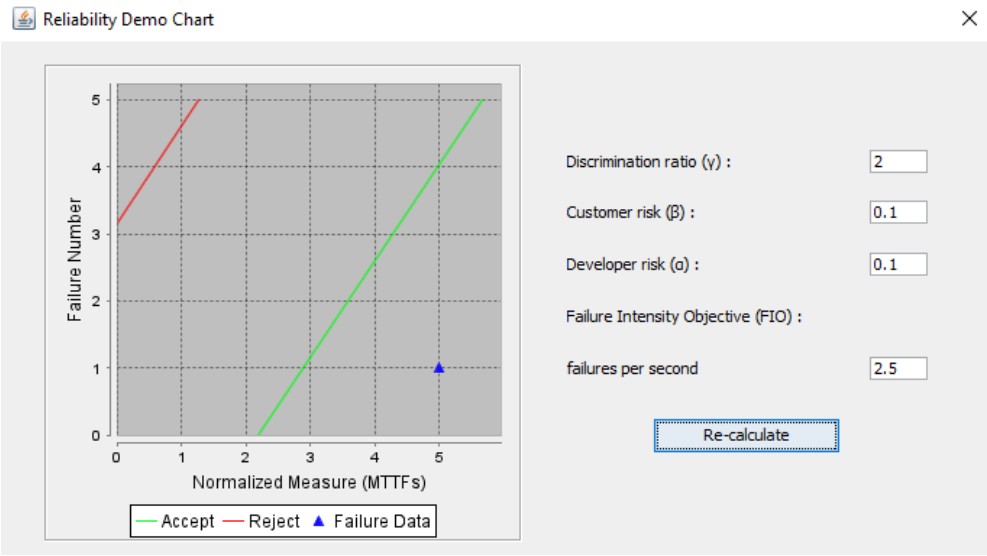
Calculating MTTF will allow us to get failures per time interval.

MTTF = (Failures at interval 18) / (Interval 18)

= (43.00) / (18) = 2.39

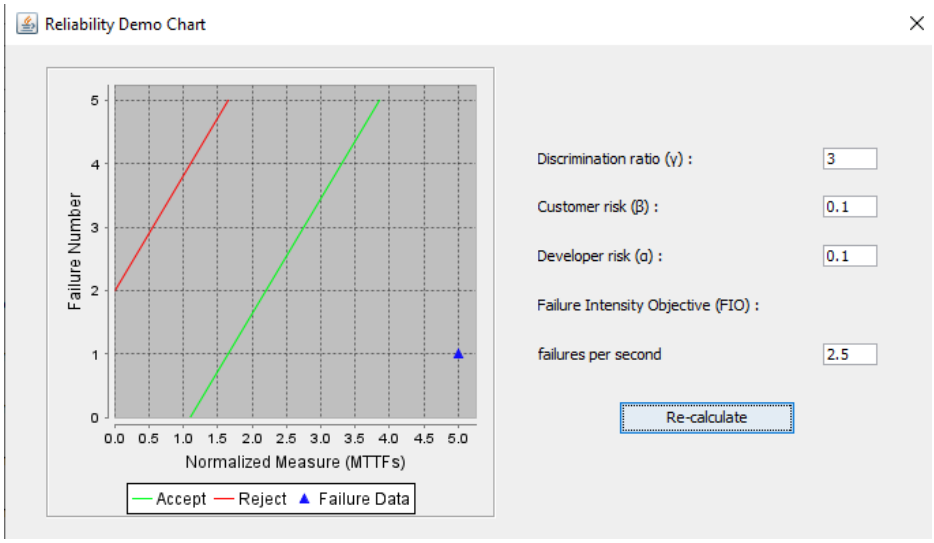
Assessment Using Reliability

Demonstration Chart

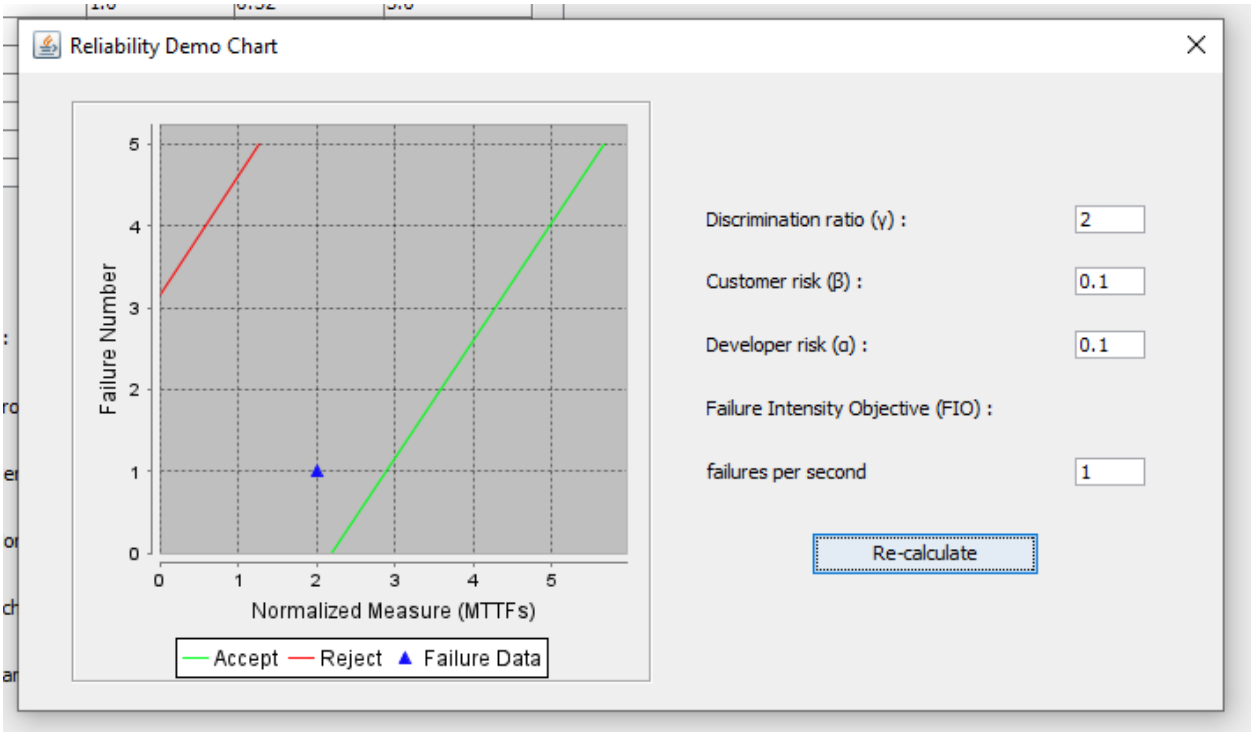


If we have a failure tolerance of 2.5 failures per second then our failure data is accepted. With a Discrimination ratio of 2, Customer risk of 0.1 and Developer risk of 0.1.

We can see that when we change the doscrimination ratio:



Our program becomes more accepting with a larder Discrimination ratio and less accepting with a smaller discrimination ratio.



We can also see that if we change our acceptance range to 1 failure per second then our data results in ambiguity and we need to do further research. This shift of the graph to the left is a result of requiring better test data to determine if the program should be accepted or rejected.

Comparison of Results

After looking at the results of both methods it can be concluded that the program is reliable. This is due to the fact that the C-STRAT program provided us with the trend in failures per second and failures at any point in the program's run time. Finding the MTTF can be found with further inference of the graphs.

Discussion on Similarity and Differences of the Two Techniques

The primary difference that can be found in the two methods is what the techniques are based on. In Reliability Growth Testing takes into account failure count and MTTF.
In the Reliability Demonstration Chart it takes into account inter failure times and MTTF.
Both graphs create a visual for the data included over a specific range of time. In the Reliability Demonstration Chart the failure data is analyzed with the accepting and rejecting conditions.

How the team work/effort was divided and managed

Teamwork is a fundamental part of the software testing process. This class has enabled us to hone our interpersonal skills and in turn allow us to collaborate and in turn split the work evenly. The lab comprised of 2 components the first being reliability growth testing and the second being the reliability demonstration chart. Our team decided that the work would be split evenly and we split both the sections amongst two people each. After each section was completed by the designated pair, the alternate section was peer reviewed and tested. At the end our group had a meeting where we went over each component of the lab step by step to ensure everything was correct and the work was divided evenly.

Difficulties encountered, challenges overcome, and lessons learned

Similar to the last lab, our first difficulty some of our team members encountered was with the software itself. The download itself took upwards of two hours on some machines as our computers refused to believe that it was not malicious. We could not get STRAT to work in the slightest due to a lack of instruction so our group had to end up resorting to the C-SFRAT software. After switching softwares to remedy this issue, we found it rather difficult to work in the second software as well, also due to a lack of instruction. The data had to be manually manipulated in order to be used as well. After a significant amount of time we were finally able to get enough of the software working to complete our lab assignment.

Comments/feedback on the lab itself

The lab itself was very challenging with a clear lack of explanation in the documentation. The documentation provided very minimal instruction and only at a very basic level, throwing us to the metaphorical wolves for the bulk of the assignment. However, the lab was interesting as it gave us an exciting glimpse into some tools which presumably will be used in the field. The introduction of tools such as C-SFRAT that we could potentially be using in our careers in the future was quite beneficial. Though there was a helpful introduction preceding the document explaining useful terms and concepts, the lab seemed to have much less instruction than the previous 4 labs, and difficulties with the software itself were rather difficult to troubleshoot.