SENG 438- Software Testing, Reliability, and Quality

Lab. Report #5 – Software Reliability Assessment

Group Number:	19
Student Names:	
Ryan	Sommerville
Eric	Renno
Quinn	Ledingham
Kaumil	Patel

Introduction:

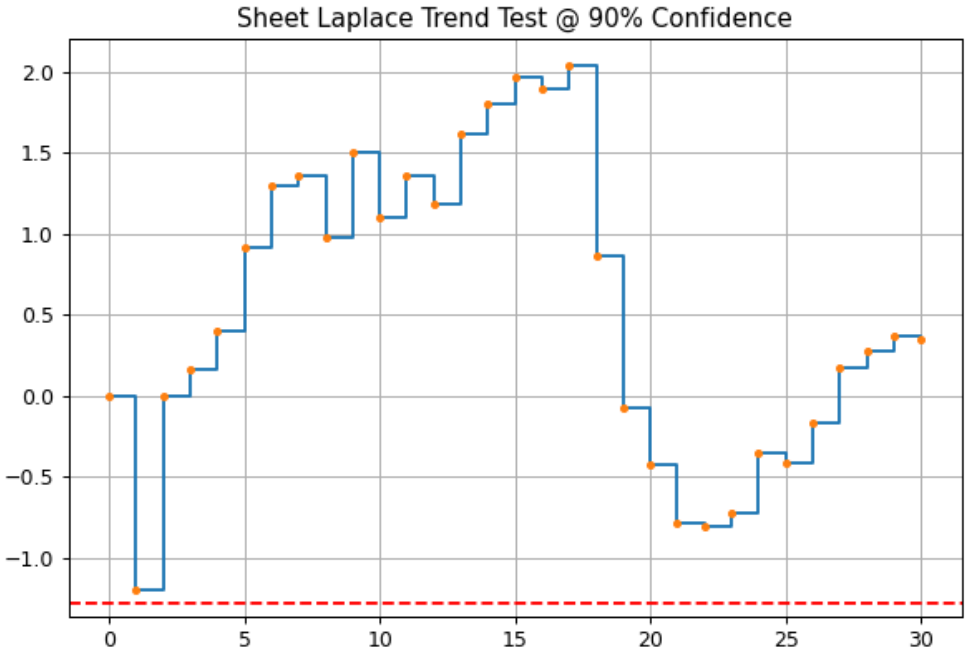
It is important for producers and users of a system to have confidence in a product. That is why measuring the Reliability of a system is so important. In order for a product to be considered reliable, there are many tools to analyze the reliability of a system. The two we used in this lab report are Reliability Growth Testing and the Reliability Demonstration Chart.

Assessment Using Reliability Growth Testing:

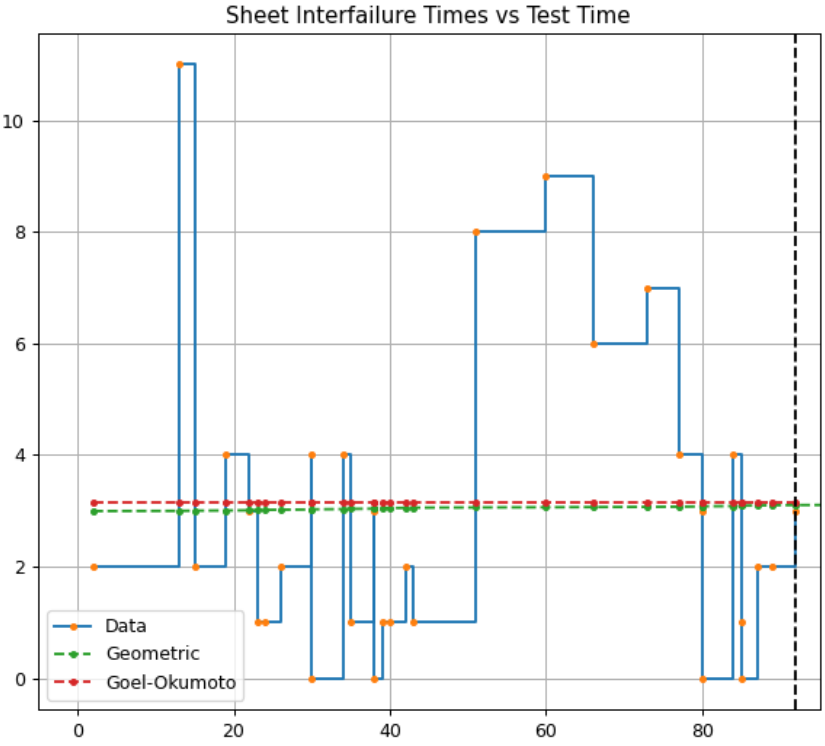
Tool: SFRAT (python)

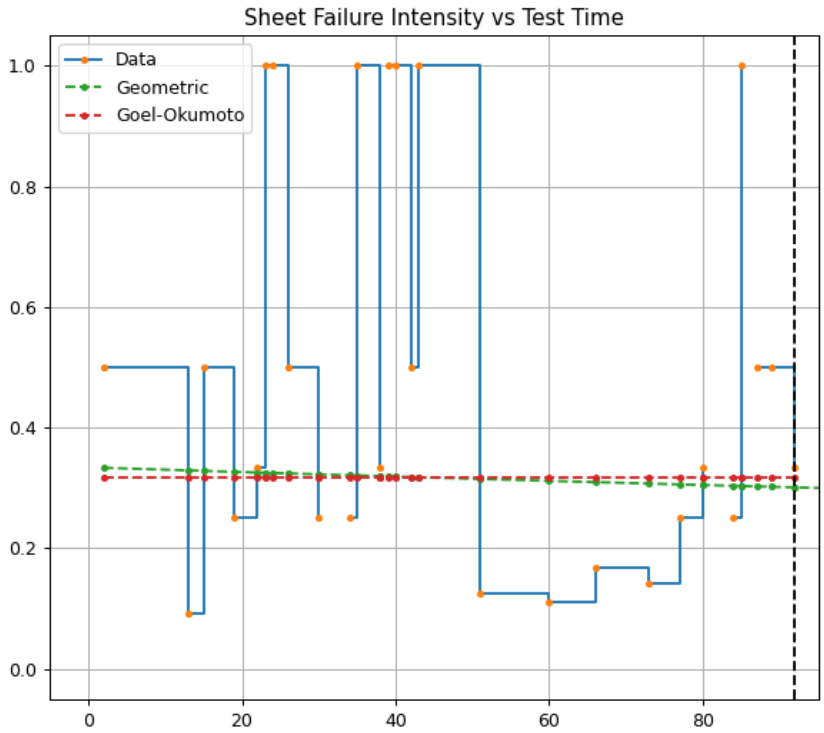
Model	AIC	PSSE 90% Data
Weibull	135.363	11.991
Inflexion S-shaped	140.442	11.573
Geometric	133.719	9.133
Delayed S-shaped	134.592	14.392
Goel-Okumoto	133.55	9.009
Jelinski-Moranda	nan	230

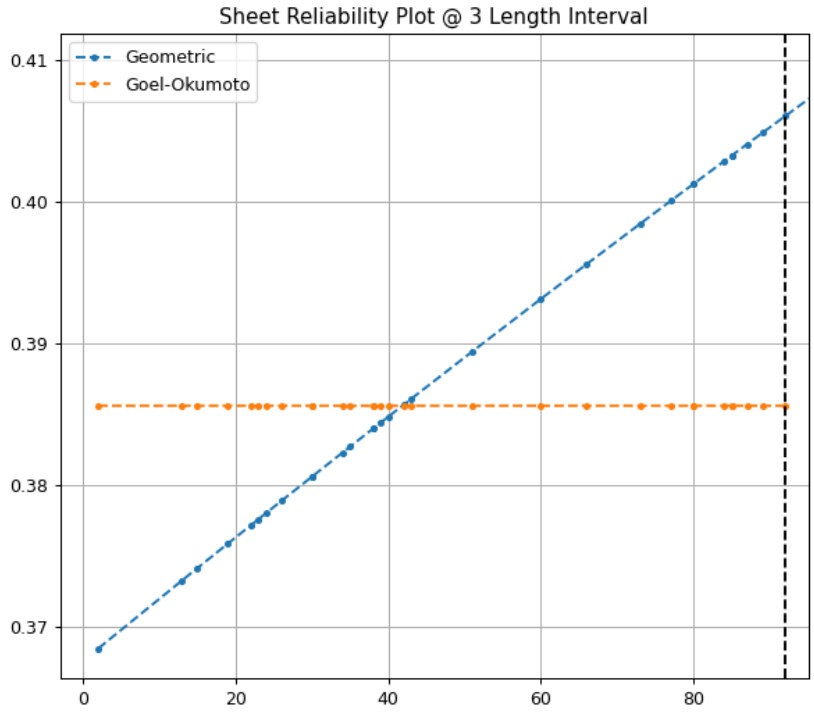
Lower AIC values indicate a better-fit model. Geometric and Goel-Okumoto have the two lowest values.



Since the trend is being analyzed using failure count data positive values mean reliability decrease and negative values mean reliability increase. The reliability model should only be used on data where the overall reliability is increasing so that would mean intervals where the values are negative like 0 - 3 or 20 - 27.







The program only allows intervals starting at 0 so it does the interval 0 - 3

Discussion on decision making given a target failure rate
After determining a target failure rate it can be used to make decisions during the software development process. How to track bugs in pre-release will be informed by the

target failure rate. If the target failure rate is low then tracking them closer during development will be required to stop them from causing more failures than the target. If the target is higher because that is what is expected the bugs might not need to be constantly tracked as much. It will also inform how to test the software. If the software is still producing more failures than the target rate that could mean that more tests need to be added to keep the number of failures being introduced from being so high. When a product is going to be released it is desirable that the software be reliable which means that the failure rate is lower or equal than the target failure rate. So the target failure rate can influence the decision to release a product because if the number of failures is above the target failure rate than it is reasonable to expect that the customers will not be happy with the product.

Discussion on advantages and disadvantages of reliabiliy growth analysis

The advantages are that the development can get a idea of how reliable their software has become and how reliable it will be for customers. It shows if less failures are occurring now than before so the development knows that their software got more reliable. The disadvantages are that assumptions have to be made to form the models and it is just a prediction of the softwares reliability. It is assuming that failure patterns during development are good indicators of the software's reliability when used by the customer which could not be true in some circumstances. If it is not true the prediction will not be correct so the analysis did not provide useful model.

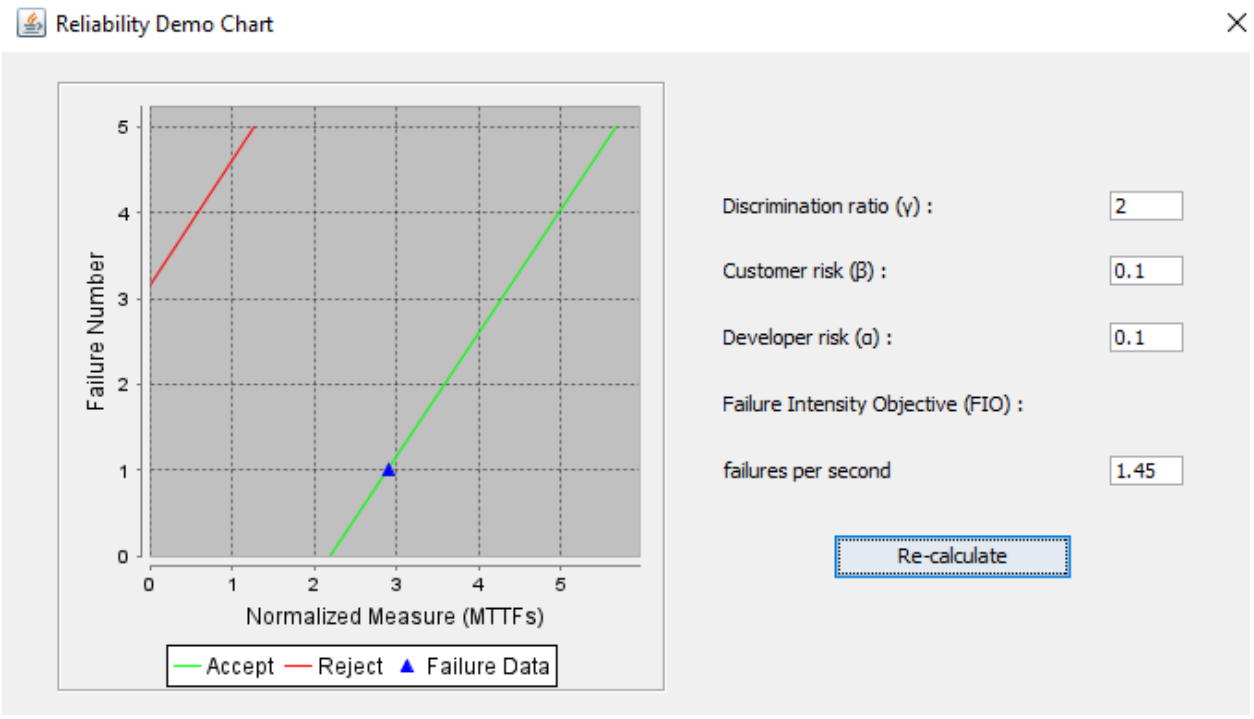
Assessment Using Reliability Demonstration Chart

RDC Data:

Confidence Levels:

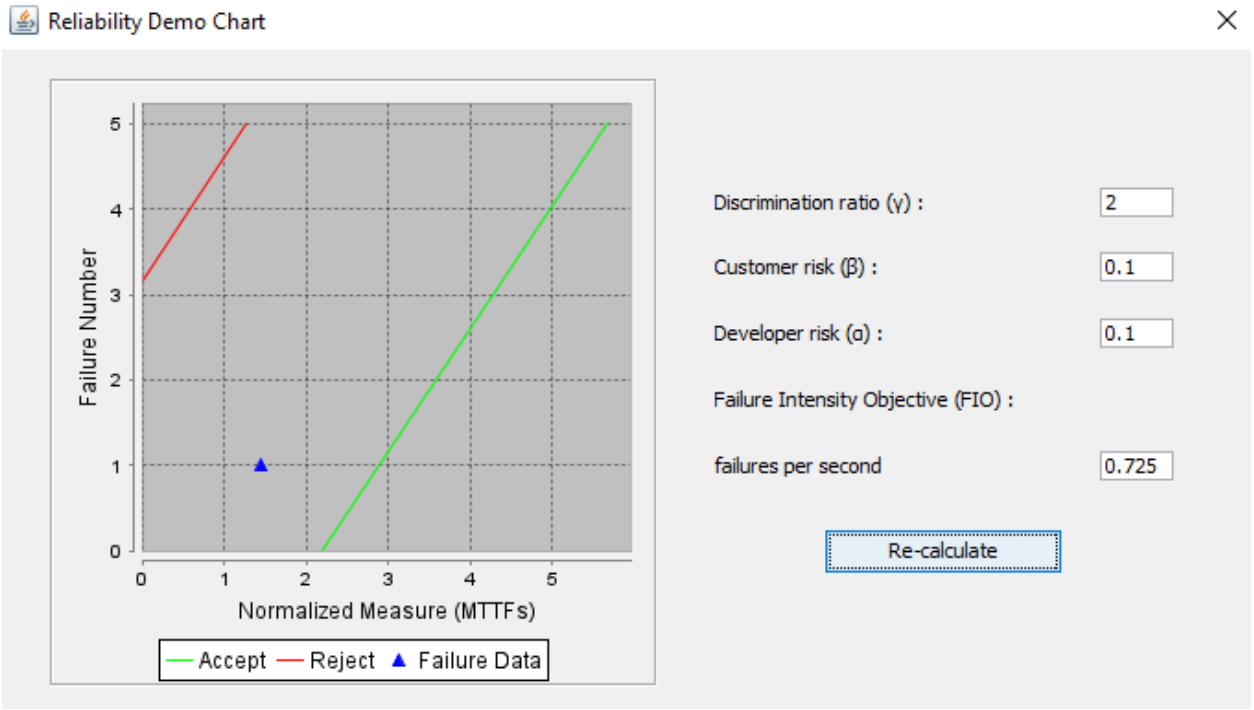
Customer Risk: 10%
Producer Risk: 10%
Discrimination Ratio: 2

Minimum MTTF:

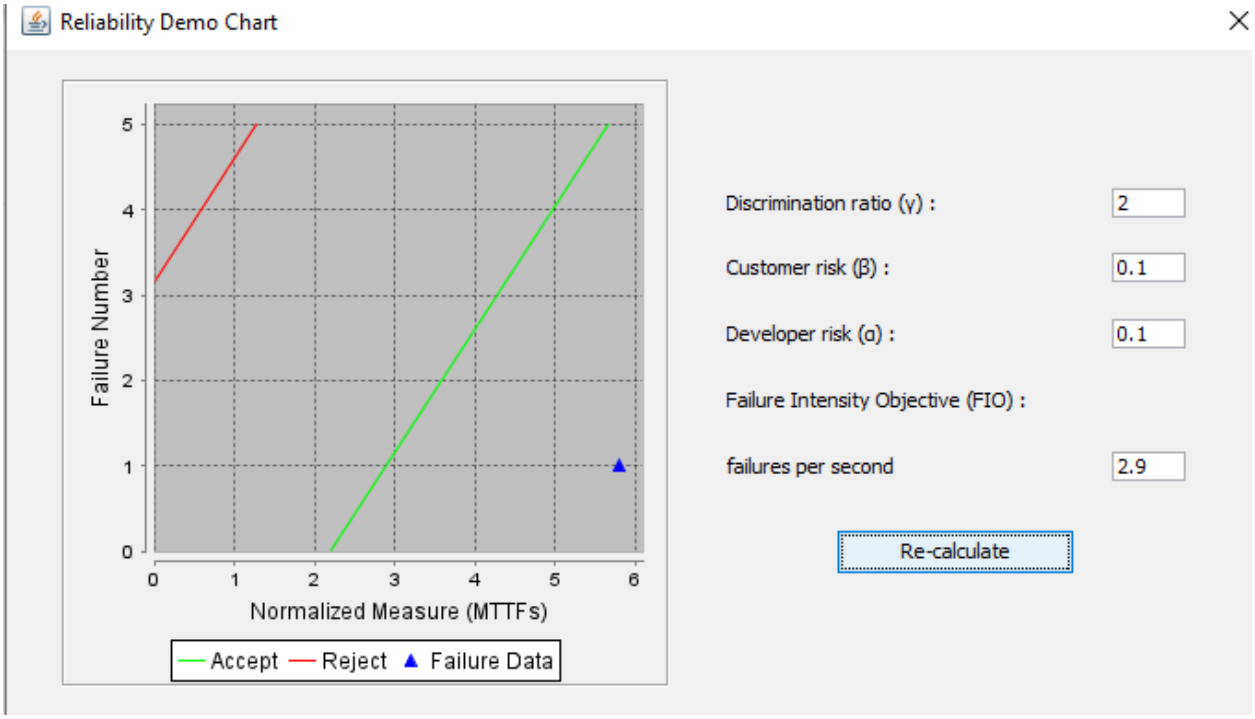


Failures per Second = 1.45. Therefore, min MTTF is 1/1.45 = 0.69 seconds.

Twice MTTF:
MTTF = 1.38 seconds. Failures per Second = 0.725



Half min MTTF:
MTTF = 0.34 seconds. Failures per second = 2.9



The minMTTF was found by adjusting the failures per second until the data was on the success line, and then finding the minMTTF by finding the reciprocal of the failures per second.

The analysis that RDC gives is an easy way to decide whether the system is reliable enough. Simply by inputting numbers and checking whether the result is in the success area or the continue area, decisions can be made. However, there are other difficulties. Mainly, it can be difficult to find reasonable numbers for the inputs. The discrimination ratio, customer risk, and developer risk are difficult to obtain concrete numbers for, and this can result in the end result seeming relatively subjective.

Comparison of Results

The dependability of a system under investigation cannot be quantified using a reliability demonstration chart. It merely emphasises the changing trend and how it may affect the system's dependability. The results of the reliability demonstration chart only depict reject, continue or accept and for this system and its failure data does not provide a indepth understanding of the failures. RDC allows use to understand a systems reblity

and its overall trend but does not provied any other metrics to study thus it can only provides a basic overview of the system and its failures.

In the sense that it allows testers to examine and analyse multiple factors such as time between failures, failure severity, and reliability graphs, reliability growth testing is considerably superior. The failure data is subjected to several models, with the best models being utilised to analyse the system's dependability. We may also acquire a better knowledge of where things are going wrong using this strategy. This method also allows us to get better results by using different trend tests such as running arithmetic average and laplace test to test if the failure data show signs of reliability growth to further study. Reliability growth testing provides better tools, metrics and results for the user to study and use inorder to make the system better.

Discussion on Similarity and Differences of the Two Techniques

Although RDC is easy to use and convenient to consider whether a program is reliable enough, it doesn't give much information besides that. Moreover, the inputs to the chart are not well defined and can seem subjective. On the other hand, Reliability Growth Testing gives far more data to analyze and understand how a system behaves, but at the same time, this means that the process will likely take longer and making a decision isn't as easy.

How the team work/effort was divided and managed

Everyone:

- Finding best models.
- Report

Ryan:

- RDC graphs
- Report

Difficulties encountered, challenges overcome, and lessons learned

It took a long time to find software that gave us the inputs we needed. Additionally, we had to figure out how to use the software ourselves since there was very little on the subject in the labs or the slides for the course.

Comments/feedback on the lab itself

It would be easier on the students in the lab if there the lab specified which software to use for which task ahead of time. Much of our time was spent simply finding the right software, which doesn't teach us anything. Additionally, it would be beneficial to have explanations on how to use the software in the lab since it prevents us from wasting time wondering whether we are using the software right.