Lab 4 - Software Reliability   
  
ETS200  
Felix Lindell ain09fli  
Fredrik Stål ada08fs1

## Theory

## Method

## Estimation tables

Text blablabla

|  |  |  |
| --- | --- | --- |
| **godata** | *Nfuture* | *Ntot* |
| *Subjective estimation* | 409 | 415 |
| *GO estimation* | 460 | 548 |
| *Actual values* | 459 | >=459\* |

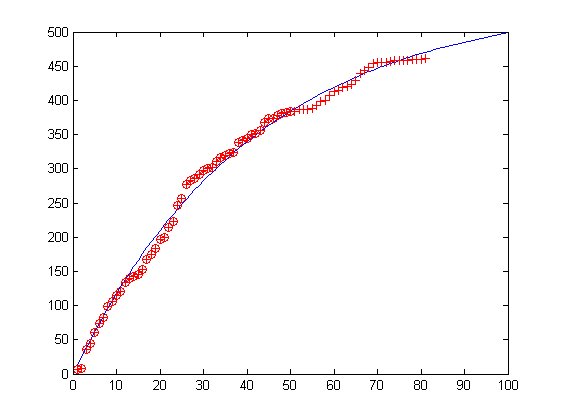
\*a=527

|  |  |  |
| --- | --- | --- |
| **jmdata** | *Nfuture* | *Ntot* |
| *Subjective estimation* | 185 | 220 |
| *JM estimation* | N/A | 226 |
| *Actual values* | 190 | >=207\* |

\*N=254

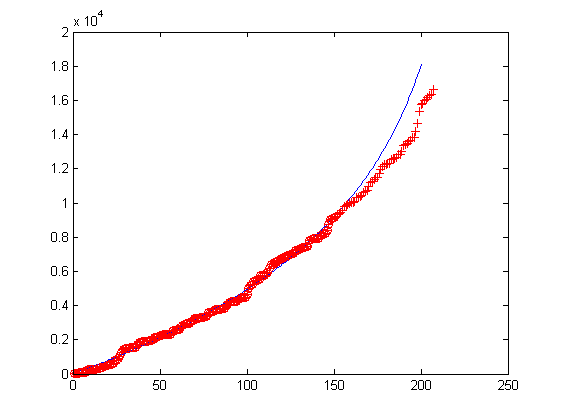
## Graphs

Inledande text



Figur 1. Godata

Kommentarer godata graf



Figur 2. Jmdata

Kommentarer jmdata graf

## Discussion

In this exercise the GO estimation model turned out to work very accurately. The subjective method performed the worst. This is most likely due to our inexperience with software reviewing and the type of project the data comes from. For the jmdata the JM estimation did not reflect the actual values. The reason for this could be factors such as small amount of data. Perhaps the method is not actually appropriate for this particular project.

We do not know how common it is to use estimations like the GO and the JM estimation in the software development industry. We believe that a person with great experience in software reviewing for a company, with access to data from previous projects could use the subjective estimation method with more accuracy.

If the estimations show a large quantity of defects remaining, it would perhaps not be wise to continue implementing additional features. Instead the focus should be on finding and correcting as many defects as possible up to a certain point. The point where the number of remaining defects is “acceptable” could be set either by the customer or the project manager etc. As the estimation for the amount of resources to find an additional defect start to increase, it might be good to instead assign the majority of resources to other parts of the development.