Software Reliability models for the first stage of Software Projects

Néstor R. Barraza[*] and Mariano Tugnarelli[**]

Universidad Nacional de Tres de Febrero, Caseros, Argentina
nbarraza@untref.edu.ar

Resumen A software reliability analysis for the first stage of software projects is presented. At this very first stage of testing we expect an increasing failure rate, where the usual software reliability growth models based on non homogeneous Poisson processes like the Goel-Okumoto or Musa-Okumoto can not be applied. However, our analysis involves some models that combine reliability growth with increasing failure rates like the logistic and delayed S-shaped models. Our analysis also includes a new model based on contagion as in the increasing failure rate as in the reliability growth stages. We point out that increasing failure rate stages are important to be modeled since corrective actions can be taken soon and also that this characteristics highlights under modern development methodologies which development is performed simultaneously as testing, like in Agile and TDD (Test driven development). Results of the application of those models to real datasets is shown.

Keywords: Software Reliability, Increasing failure rate, Contagion

1. Introduction

Since Software has become a quite important component in many technological projects, models and procedures assuring its quality are also very important. Then, the software reliability speciality rises models in order to predict some metrics like the mean time to failure or the probability of failure in a given environment. This prediction involves for example the release time or the reliability of the final product. A good prediction could be useful to save a lot of money for the Company. When the SR started decades ago, the software development methodology commonly applied was the waterfall, with a well differentiated development and testing stages. In this case, if coding is added just in order to fix failures, we can expect a decreasing number of failures in the software, it means, a reliability growth. Then, the first developed SR models were those based on non homogeneous Poisson processes, like the Goel-Okumoto and Musa-Okumoto models, see for example [8]. Over the past decades, new software methodologies were developed, Agile, TDD (Test Driven Development), Scrum, etc. The main

[*] The author is also with the School of Engineering. UBA.

[**] The author is also with Grupo Esfera S.A.

characteristic introduced is that contrarily to the waterfall methodology, testing is performed simultaneously as coding. We can expect then, an increasing failure rate at the start of testing, and surely for a log period of time, when the software is mature enough as to present reliability growth characteristics. Unfortunately, since software factories are reluctant of delivering their reports, there are no datasets available as to test those cases. We are now working on this issue and results will be shown in future works. The aim of this work is to analyze the performance of models for this first stage, it means, models able to fit increasing failure rate curves. Since most of this models present an increasing failure rate portion at first and a flat cumulative number of failures curve at the end, they can also be classified as reliability growth models, such as the delayed S-shaped and logistic models among others, see [12], [10] and [5]. There are also models proposed to fit just the increasing failure rate stage, [1], [4], [2]. Despite the mentioned lack of datasetsets, there are several reports in the literature that albeit were not taken under modern testing and development methodologies, they present the mentioned characteristic of having an increasing failure rate at the start of the report. In this work, we apply the mentioned models to those datasets at the first stage and evaluate their performance. Our results are useful to determine which model to apply at the first stage of large software projects, mainly those developed under modern software development and testing methodologies.

This work is organized as follows: Some proper SR models for the first stage of software development are presented in section 2, applications of those models in real datasets are developed in section 3, some concluding remarks are discussed in section 4, conclusions are presented in 5.

2. Proper Software Reliability models for the first stage

Any stochastic SR model gives a probability of failures as a function of time, and the mean number of failures is obtained as:

$$\mu(t) = \sum_{r=0}^{\infty} r P_r(t) \tag{1}$$

where $P_r(t)$ is the probability of having r failures detected by the time t.

For models based on non homogeneous Poisson processes (NHPP), the failure rate $\lambda(t)$ depends just over time and the mean number of failures is given by:

$$\mu(t) = \int_0^t \lambda(t) dt \tag{2}$$

The other models considered here are based on contagion, where the probability of failures is the solution of a Pure Birth process equation [6]:

$$P'_r(t) = -\lambda_r(t)P_r(t) + \lambda_{r-1}(t)P_{r-1}(t) \tag{3}$$

In this process, the mean number of failures is obtained by summing up over the Eq. 3, getting this way the following differential equation for the mean:

$$\mu'(t) = -\sum_{r=1}^{\infty} r\ \lambda_r(t) P_r(t) + \sum_{r=1}^{\infty} r\ \lambda_{r-1}(t) P_{r-1}(t) \tag{4}$$

It must be remarked that NHPP are also Pure Birth processes, then, in this case, Eq. 2 comes from 3 when $\lambda_r(t) = \lambda(t)$.

Models considered in this work are listed in tables 1 and 2. For the second group, we list both, the failure rate $\lambda_r(t)$, and the mean number of failures:

Tabla 1. Increasing/decreasing failure rate NHPP based Software Reliability models

Model	$\mu(t,0)$
Delayed S-shaped [12]	$a(1-(1+b\ t)\mathrm{e}^{-b\ t})$
Logistic [7]	$\dfrac{a}{1+e^{-b(t-c)}}$

Tabla 2. Contagion based Software Reliability models

Model	Failure rate	$\mu(t,0)$
Polya [1]	a $\dfrac{b+r}{1+a\ t}$	$1+a\ t$
Modified Polya ([4], [2])	a $\dfrac{1+b\ r}{1+a\ t}$	$(1+a\ t)^b - 1$

Due to the mean number of failures depends linearly over time, the Polya model can not be applied to many real cases. However, it served as inspiration for other contagion models, then, it is listed in table 2 just for reference and will not be taken into account in the experiments. It must be noted that the Modified Polya process is able to fit either purely increasing failure rate stages or reliability growth stages, provided the parameter b is either greater or lower than one.

3. Experiments

The application of the mentioned models on several datasets presented in the literature is analyzed next. The models are fitted using the Least Square method. In order to measure the goodness of fit we use the Predictive ratio risk (PRR), see for example [9] for an explanation of this metric.

3.1. Naval Technical Data System

These data corresponds to the development of software for the real-time multi-computer complex of the US Naval Fleet Computer Programming Center of the US Naval Tactical Data Systems (NTDS) and were collected years ago (1970) as referenced in [9]. Despite the long time ago these data were reported, they were widely used in the literature and presents the increasing failure rate stage useful to test the models proposed in this work.

Curves obtained by fitting the delayed S-shaped, Logistic and Modified Polya models are shown in fig. 1. The parameter values and the goodness of fit are listed in table 3.

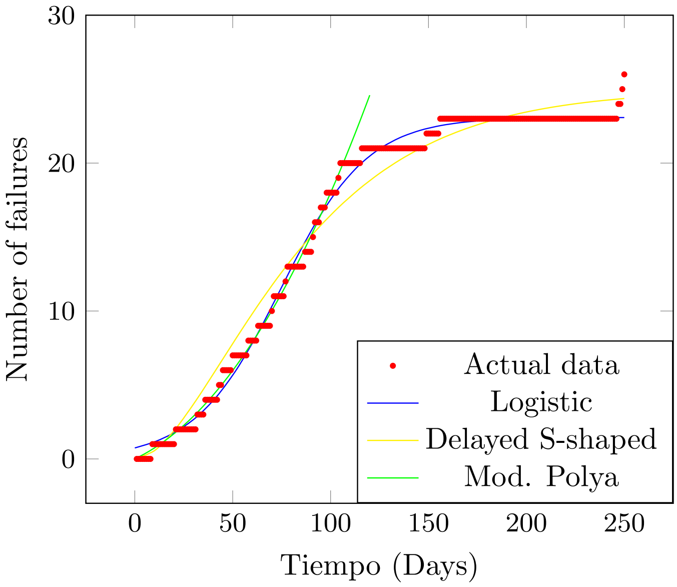

Figura 1. NTDS data.

Despite the PRR metric is useful for comparison of models applied over the same range of data, it can also serves as an indicative measure for an only model.

We can see from table 3 that the modified Polya model fits quite well the increasing failure rate corresponding to the first 110 days. The logistic model fits the best the increasing/decreasing failure rate combined with a reliability

Tabla 3. NTDS data. Goodness of fit.

Period (days)	Model	PRR
1 to 110	$(1+0{,}0295414t)^{2{,}14153}-1$	7.113
1 to 250	$24{,}9457\,(1-e^{-0{,}0226362t}(1+0{,}0226362t))$	33.6686
1 to 250	$\dfrac{23{,}087}{1+e^{-0{,}0455646(-74{,}7119+t)}}$	15.2366

growth, a result in agreement with a previous work by other authors on a different dataset: [5].

3.2. Monitoring and Real time system

This dataset was first presented in [11]. It corresponds to a monitoring and real time system. The project consists of 200 modules totaling approximately 200 kloc, and written in a high level language. The cumulative number of failures is shown in fig. 2. We can see that this report exhibits several shapes and was analyzed as a multistage project in [3]. We focus on the increasing failure rate stage elapsed from the beginning to day number 38. The models fitted for this stage and their goodness of fit are listed in table 4. Curves are depicted in fig. 3.

It must be noted that the modified Polya model fits two periods in the first stage, the pure increasing failure rate corresponding to the first 18 days and the following period from day number 13 to day number 38 where the project reaches a reliability growth, getting then a b parameter lower than one.

Tabla 4. Real Time System data. Goodness of fit.

Period (days)	Model	PRR
1 to 18	$(1+0{,}608233\,t)^{2{,}17109}-1$	41[1]
1 to 38	$327{,}745\,(1+e^{-0{,}101881\,t}\,(1+0{,}101881\,t))$	88.8518
1 to 38	$\dfrac{254{,}853}{1+e^{-0{,}283989\,(x-13{,}58234)}}$	26
13 to 38	$165{,}925\,(1-e^{-0{,}148636\,(t-13)})+108$	14.17
13 to 38	$(1+9695{,}88\,(t-13))^{0{,}4177}+108$	0.75

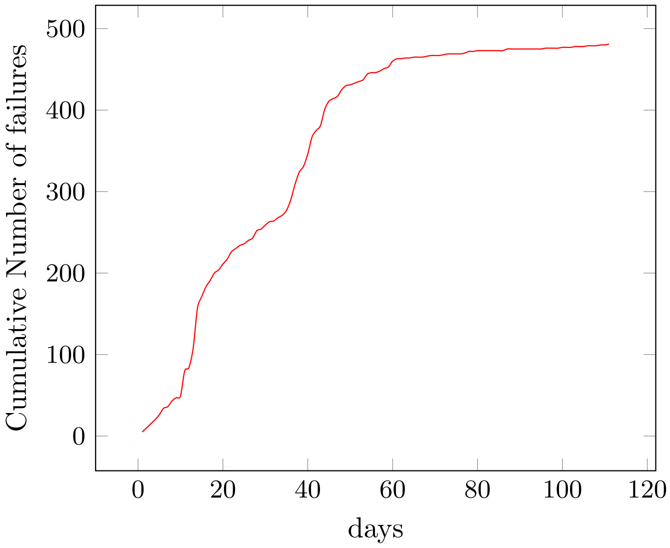

Figura 2. Real Time data.

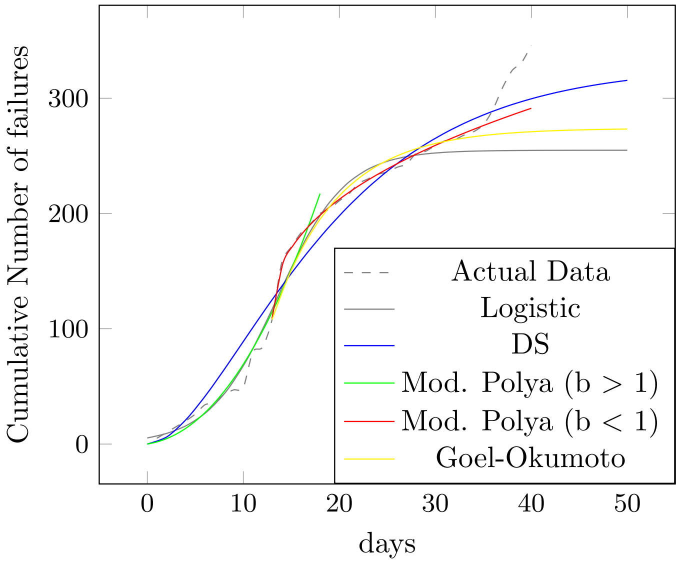

Figura 3. Real Time data. Fitted models

We can see from table 3 that again the logistic model attains the best fit in the increasing-decreasing-reliability growth stage. The reliability growth stage between days number 13 and 38 is better fitted by the Modified Polya model in its reliability growth mode, it means, being its b parameter lower than 1. This is interesting since generally, reliability growth stages are fitted the best by the Goel-Okumoto model, however, since this portion does not attain a flat curve but a slow decreasing failure rate, the modified Polya process fits better.

3.3. Client-Server System

This software is also a multistage project having a similar metric than that analyzed in the previous subsection. It corresponds to a client-server system developed between 2001 through 2005 and comprises approximately 250 kloc of C language, involving modules for the host, terminals and communications. The cumulative number of failures is depicted in fig. 4.

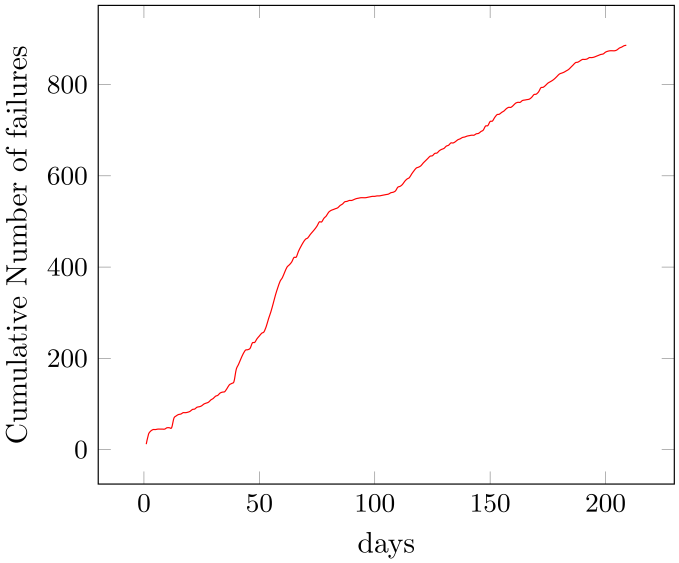

Figura 4. Client-Server system.

From fig. 4 we can distinguish two big stages bounded at day number 110. The first one presents an increasing failure rate process ending in reliability growth. The second one presents a curve with a constant failure rate, it is because the software was released pretty soon and was not mature enough so failure reports involve both, those reported from testing and from customers.

The fitted models are shown in fig. 5 and listed in table 5.

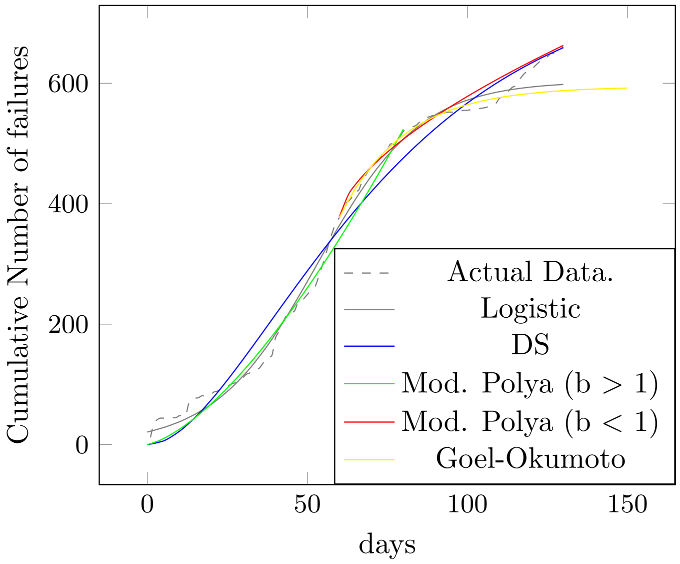

Figura 5. Client-Server data. Fitted models

Tabla 5. Client Server data. Goodness of fit.

Period (days)	Model	PRR
1 to 80	$(1+0{,}770584\ t)^{1{,}51363}-1$	1398.1^2
1 to 120	$775{,}704\ (1+e^{-0{,}0259183\ t}\ (1+0{,}0259183\ t))$	3807.93
1 to 120	$\dfrac{602{,}969}{1+e^{-0{,}0624053\ (x-53{,}2771)}}$	211.634
60 to 90	$217{,}536\ (1-e^{-0{,}0493589\ (t-60)})+377$	1.36
60 to 98	$377+(1+115{,}712(t-60))^{0{,}628382}$	5.98

Values listed in table 5 show again that the logistic model fits the best the increasing-decreasing-reliability growth stage, the Goel-Okumoto model fits the best the reliability growth portion of data, though the modified Polya model in its reliability growth mode (b parameter lower than 1) fits close to the GO model.

3.4. Agile project

This dataset corresponds to a Real Time System developed under Agiles methodologies. The actual dataset and fitted models are depicted in Fig. 6. The PRR metrics is listed in table 6. The actual data show an increasing failure rate mainly due to that the project was still in the development and debugging phase. Failure data correspond to production reports since the product was early delivered in order to be tested by the customer.

We can also see from Fig. 6 that the prediction of both models are quite different, the DS model predicts an almost immediate reliability growth and the Modified Polya model continues with an increasing failure rate.

The goodness of fit of the Modified Polya model is slightly better than the DS model.

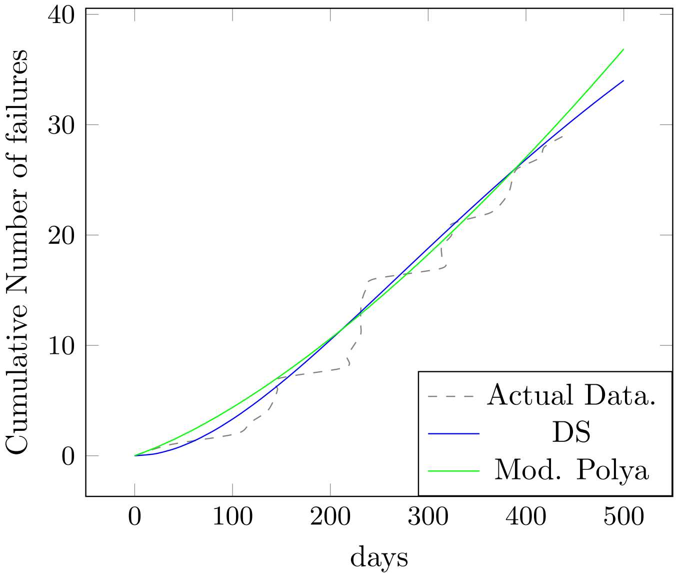

Figura 6. Agile proyect. Fitted models

Tabla 6. Agile Project. Goodness of fit.

Model	PRR
$61{,}3964(1 - e^{-0{,}00371498\,t}(1 + 0{,}00371498t))$	465.918
$(1 + 0{,}0209466t)^{1{,}48928} - 1$	436.028

4. Concluding remarks

We enumerate next the conclusions of the experiments:

1. The logistic model fits the best the increasing-decreasing-reliability growth stage.
2. The Goel-Okumoto model attains the better fit for the Reliability growth stages
3. The Modified Polya model fits quite well the purely increasing failure rate portion of data, achieving to predict more than the 50 % of a quite short training set involving just a few days.
4. The Modified Polya model with a b parameter lower than one fits a reliability growth portion of data better than the Goel-Okumoto model in cases were the flat portion of the curve is not immediately attained.

These conclusions help to choose the best model to be applied in many stages of the development and testing phases.

5. Conclusion

The increasing failure rate Software Reliability modeling was analyzed. The importance of this first stage of projects, specially when development and testing is performed under modern Software Engineering methodologies was remarked. We found that the Logistic Model fits the best the increasing-decreasing-reliability growth stage, in agreement with a previous work by other authors on another dataset. We confirmed once more that the Goel-Okumoto model fits the best the pure reliability growth stages. The modified Polya model introduced in the literature fits quite well the solely increasing failure rate stage, and when used as a reliability growth model it can fit better than the Goel-Okumoto model, specially when the reliability growth flat curve is not soon attained by a decreasing failure rate stage, as it was shown. Results of the application of those models on three software projects were shown and discussed.

Acknowledgments

This work was supported by Universidad Nacional de Tres de Febrero under grant no. 32/473 A.

Referencias

1. Barraza, N.R.: Software reliability modeled on contagion. In: 2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 49–50 (Oct 2016)
2. Barraza, N.R.: A new software reliability model based on contagion (2018), manuscript submitted for publication
3. Barraza, N.R.: Software reliability analysis of multistage projects (2018), manuscript submitted for publication
4. Barraza, N.R.: Mining bugzilla datasets with new increasing failure rate software reliability models. In: XLIII Latin American Computing Conference, CLEI 2017, Cordoba, Argentina, September 4-8 (2017), `http://www.clei2017-46jaiio.sadio.org.ar/sites/default/files/Mem/SLISW/slisw-04.pdf`
5. Chi, J., Honda, K., Washizaki, H., Fukazawa, Y., Munakata, K., Morita, S., Uehara, T., Yamamoto, R.: Defect analysis and prediction by applying the multistage software reliability growth model. In: 8th International Workshop on Empirical Software Engineering in Practice, IWESEP 2017, Tokyo, Japan, March 13, 2017. pp. 7–11. IEEE (2017), `https://doi.org/10.1109/IWESEP.2017.16`
6. Feller, W.: An Introduction to Probability Theory and Its Applications, Vol. 1. Wiley, 3rd edn. (Jan 1968)
7. Gokhale, S.S., Trivedi, K.S.: Log-logistic software reliability growth model. In: Proceedings Third IEEE International High-Assurance Systems Engineering Symposium (Cat. No.98EX231). pp. 34–41 (Nov 1998)
8. Lyu, M.R. (ed.): Handbook of Software Reliability Engineering. McGraw-Hill, Inc., Hightstown, NJ, USA (1996)
9. Pham, H.: System Software Reliability. Springer Publishing Company, Incorporated, 1st edn. (2010)
10. Sideratos, I., Platis, A., Koutras, V., Ampazis, N.: Reliability analysis of a two-stage goel-okumoto and yamada s-shaped model. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland, Advances in Intelligent Systems and Computing, vol. 286, pp. 393–402. Springer International Publishing (2014), `http://dx.doi.org/10.1007/978-3-319-07013-1\_38`
11. Tohma, Y., Yamano, H., Ohba, M., Jacoby, R.: The estimation of parameters of the hypergeometric distribution and its application to the software reliability growth model. IEEE Trans. Softw. Eng. 17(5), 483–489 (May 1991), `http://dx.doi.org/10.1109/32.90450`
12. Yamada, S., Ohba, M., Osaki, S.: S-shaped reliability growth modeling for software error detection. IEEE Transactions on Reliability R-32(5), 475–484 (Dec 1983)