

The use of reliability growth models in project management

Eduardo Miranda
Ericsson Research Canada

Abstract

This article describes the use of reliability growth models for planning resources, monitoring progress and performing risk analysis of testing activities.

1 Introduction

Every software product contains errors and every development project includes a number of testing activities designed to detect them. Now, as a Project Manager (PM) or Test Manager (TM) for one of these projects, imagine the advantages of knowing how many errors exist, the rate at which they will be found, and how many of them are left undiscovered at any given time. You could then:

- Plan how many resources and when they will be needed,
- Decide when to stop testing;
- Evaluate the quality of the software being produced; and
- Decide when to release it to your customers.

Imagine that, based on the daily results of your testing activities, you could forecast what is going to happen. Then, rather than waiting for the next milestone, you could be taking today the steps necessary to meet your deadline.

Determining these numbers, however, is not an easy task. Insight and experience are good, but not enough, as many human traits get in the way of a good estimation. At Ericsson Research Canada, one of the paths we are exploring to remedy this situation, is the use of Reliability Growth Models (RGMs) to predict the number of latent errors.

RGMs are not new, but as their name indicates, they have been used mostly for reliability calculations requiring assumptions, not only about the distribution of errors, but also about the conditions under which the software will be executed. Calculating the number of errors left does not, making the models easier to understand and apply.

2 Why it is so difficult to estimate the number of errors

The number of errors left in a software artifact at a given time, is very tricky to estimate as some basic human traits conspire to affect the accuracy of the expert's forecasts.

First, as described by Doner[1], the mind deals with temporal processes by projecting some noticeable feature of the present in a more or less lineal and monotonic fashion. This is not well suited to guess at the future yield of a process, which is not lineal and in many cases, not monotonic either.

Second, the counter intuitive behaviour of finding a large number of errors. After finding a large number of errors we tend to think we have found them all, but as Kan explains [2], given certain process efficiency, this is only indicative that many more should be expected.

Finally, we have a natural desire to please and to avoid confrontations, which translates into optimistic reports and wishing for the best rather than bringing bad news.

This in no way advocates disregarding the experts' judgement, but rather using it to confirm or reject forecasts made by more objective means.

3 Errors, Failures and Predictions

An error, a bug or a fault, is something wrong with the code; a failure is the manifestation of a fault when the software is being executed. For a failure to occur two things are needed: an error and a trigger.

The number of latent errors predicted by the models is an estimate of the true number of errors left, which is unknown, computed from the number of failures observed during testing.

The quality of the prediction depends on the thoroughness of the testing. If the test cases utilized fail to trigger failures, for example not testing for invalid values, those errors will not be accounted for and the predictions will fall short of the real number.

It is critical to the successful implementation of these models to understand that the results of the predictions are not absolute, but relative to a given testing context, and that they should be used as an indicator within that context alone.

4 Choosing the “Right” model

At Ericsson Research we have found that RGMs can be classified into two broad categories: those that match our process behaviour and those that do not.

Among those who match our process behaviour, we haven't found big differences in prediction accuracy. Remember that the real problem is not to find the equation that best fit a couple of data points coming from past projects, but to forecast the results of ongoing testing activities along a known defect discovery pattern in an economical and practical way.

The key question we are trying to answer, is not whether the model predicts 200 or 220 Trouble Reports(TR), but whether the plans are based on the assumption of having 200 TRs, when the data coming from the trenches indicates that we will be pounded by 500 or even 1,000 TRs.

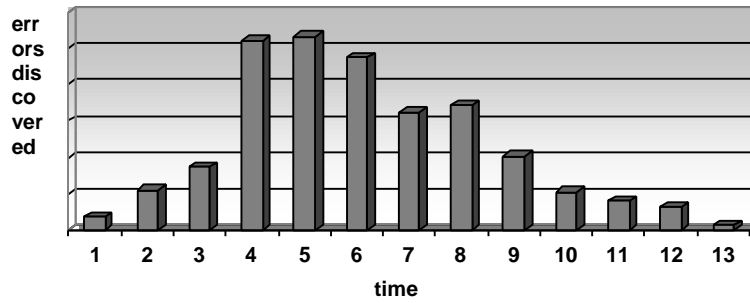
Figure 1 shows several patterns representative of the more than 20 projects and testing activities that we studied at Ericsson.

5 The number of errors discovered at a given time is a random variable

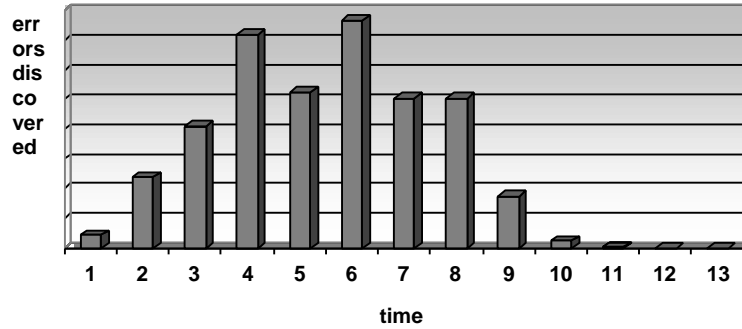
RGMs describe the trend or pattern to which the yield of a “healthy” testing process conforms, but very few times other than by chance, we will discover the exact number of errors the model predicted. Furthermore, until some time has passed, there is the possibility that the whole curve may be wrong by a significant amount.

To address these concerns we have done two things:

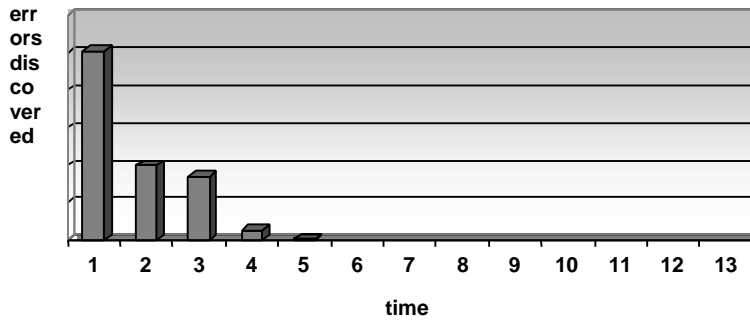
- First, instead of providing a prediction as a single value we calculate a range based on the standard error of the projection
- Second, each curve is associated with a margin of error which gives a rough idea of the probability of that curve being the true one.



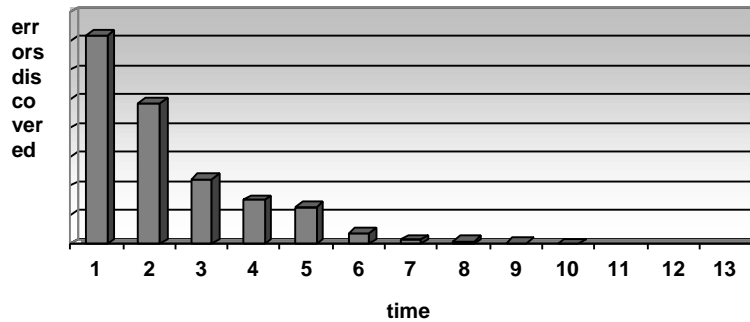
(a) Project C1 – Pattern of error discovery during development



(b) Project C2 – Pattern of error discovery during development



(c) Project FMP3 – Pattern of error discovery following product release



(d) Project FMP4 – Pattern of error discovery following product release

Figure 1 Patterns of error discovery at

Figure 2, shows the Forecast Chart produced by the Error Projection Model (EPM) tool developed at Ericsson. The way to read the chart is: assuming that the curve is right, margin

of error 40%, then there is a 68% probability that the number of errors discovered at interval 7, will be between 59 and 10 with a mean value of 34.

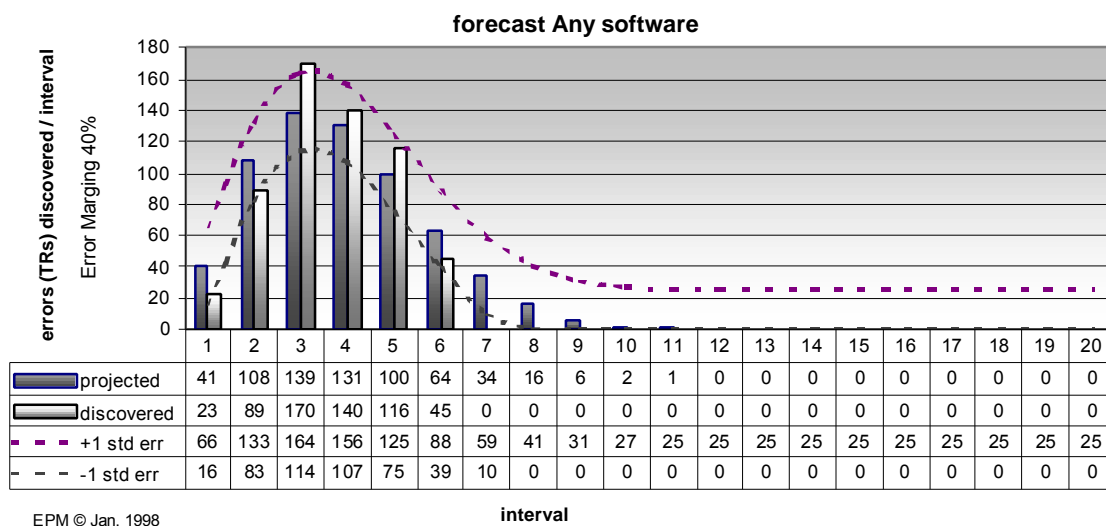


Figure 2 EPM's Forecast Chart

The margin of error is an empirical indicator designed to remind PMs and TMs of the variability of the whole curve. Calculated as a function of the number of observations, and although consistent with the observed accuracy of the RGMs used, the error margin has no statistical validity. For example, with only two observations, the margin of error is arbitrarily set at 70 % in order to warn PMs and TMs of the risk taken by basing a decision on a very small number of observations. Figure 3 shows the settings of the margin of error as a function of the percentage of testing completed.

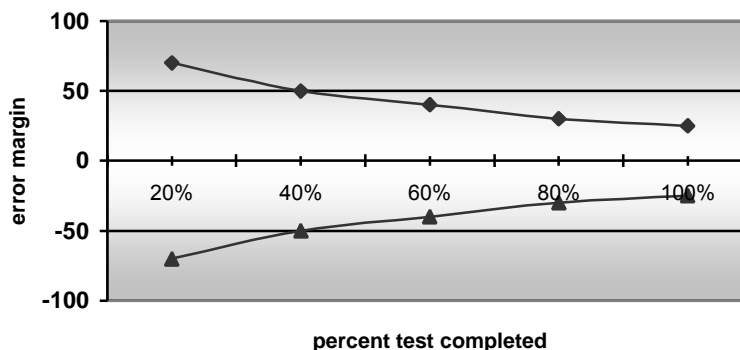


Figure 3 EPM's margin of error settings

6 Using the models to make decisions

At Ericsson Research, PMs and TMs are starting to use RGMs to help them make decisions on the basis of quantitative information.

6.1 Planning Decisions

In this case, the PM or TM starts by defining how much time is available for testing, the expected error content and the acceptable numbers of errors at the end of the testing period.

The available time is defined based on the overall project plan. The total error content is specified by providing three values: *Best, Most Likely and Worst Case* usually based on the experience from previous projects. The number of acceptable errors is decided as a function of the impact: economic, customer satisfaction, etc. that the presence of errors in the software could have. Once these values have been specified, the model is used to distribute the total error content over the allocated period as shown by Figure 4.

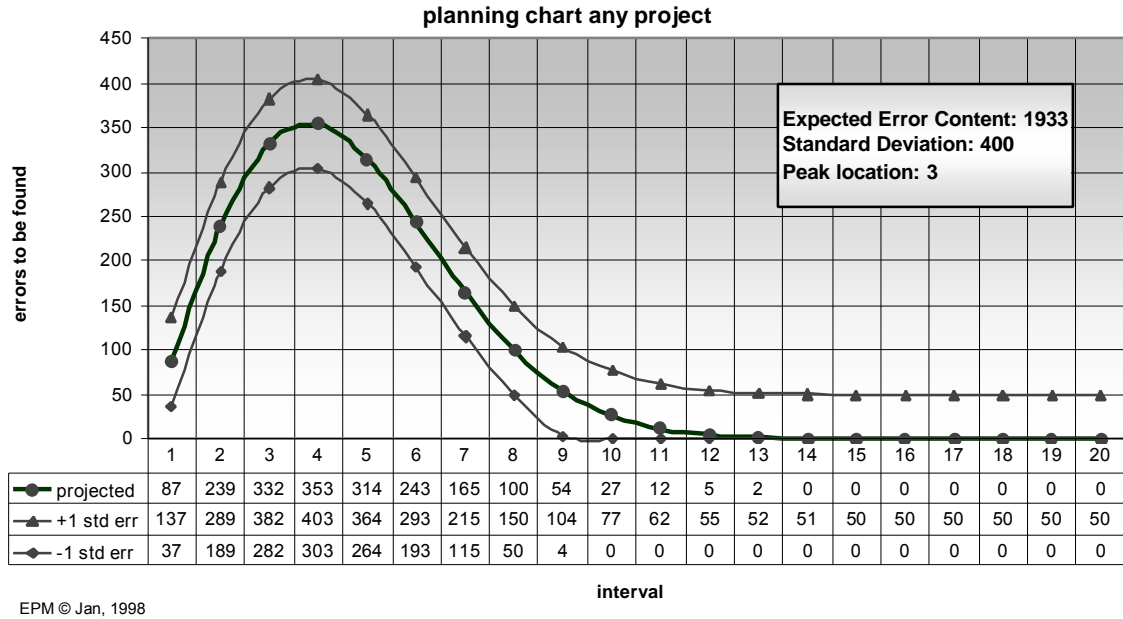


Figure 4 EPM's Planning Chart

Once this is done, the PM should decide whether or not this is feasible. If it is not, the PM shall begin a new round of planning, changing one or both of the variables under his or her direct control: the time allowed for testing or the amount of errors acceptable at release¹.

6.2 Risk Mitigation Decisions

Estimates To Complete (ETC) tell the PM how much testing is left before a certain level of latent errors is reached. By removing the guesswork from ETC preparation, RGMs provide a credible early warning for the possible evolution of the situation, allowing PMs to take decisions other than delaying the project. Figure 5 illustrates the concept.

As an example, assume a project whose goal is to complete testing in a six week period, releasing the product to the market with less than hundred latent errors. As Figure 5 shows, should the project continue at the current rate of progress, the target error content will be achieved only after interval nine, three weeks later than originally planned. Notice that this prediction is being made at week three into the testing phase, that is half way through the testing. Compare this to arriving at the same conclusion by week 5.

¹It is possible to change the expected error content as well, by launching process improvement initiatives like using better tools or adopting new, more capable processes.

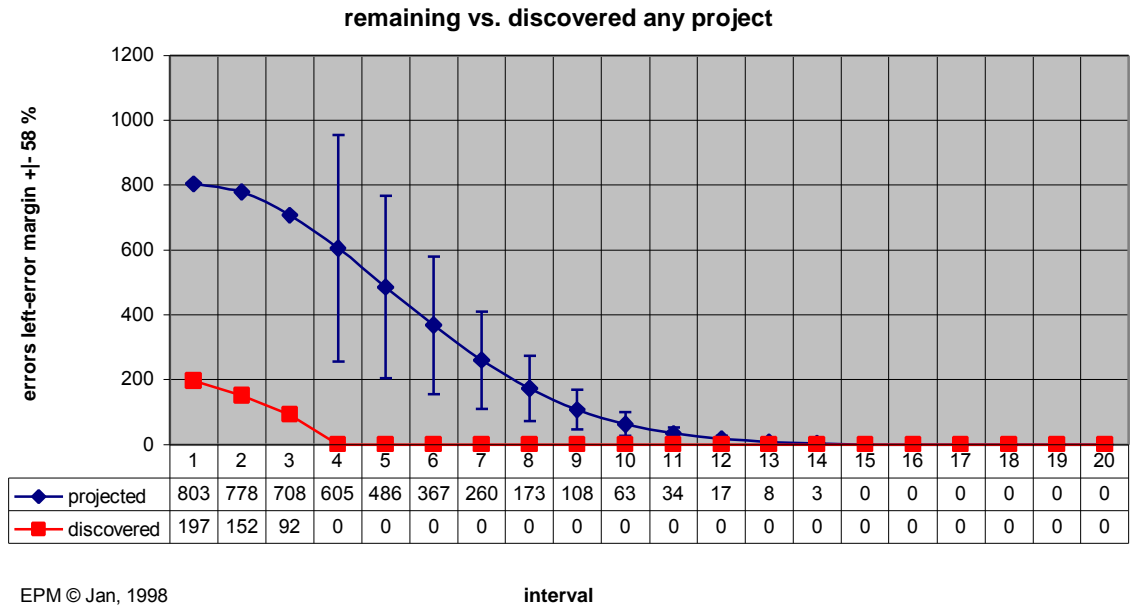


Figure 5 EPM's Errors Left Chart

The “What If” analysis allows PMs and TMs to assess the impact of making a wrong assumptions in the estimation process. What happen if the peak of the error discovery pattern occurs one or two weeks later ? What if the peak has already occurred ? The number of errors projected, and certainly, the PM’s reaction will be quite different. The “What If” analysis, shown in figure 6, helps the PMs and TMs mitigate risks by quantifying various scenarios.

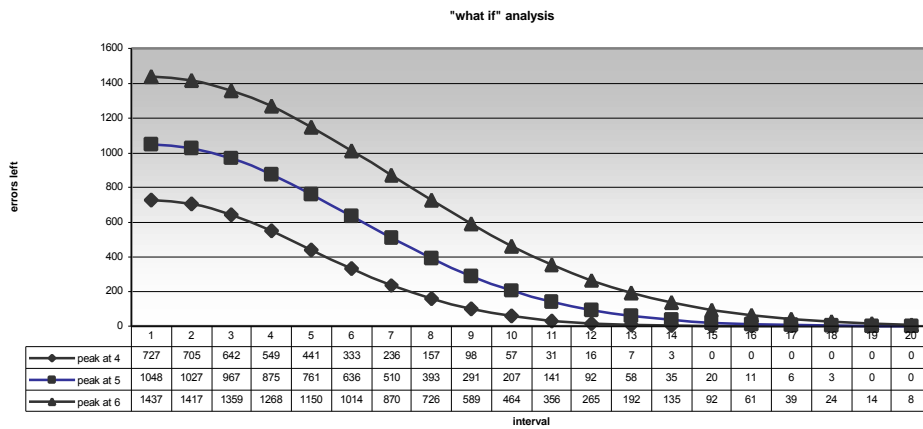


Figure 6 EPM "What If" Chart

6.3 Promotion/Release Decisions

A critical decision, which has hunted PMs for years, is when to stop testing and promote the application to the next phase of testing or release it to the market.

There are different criteria on which to base these decisions. The one favoured in this article, mandates that testing stops only when the number of errors left undiscovered is consistent with our reliability goals.

From an economic perspective, testing should stop as soon as the marginal cost of detecting and fixing an error at the current phase becomes higher than the cost of finding and fixing it at the next one, including the operational phase.

RGMs provide the quantitative basis necessary to apply these last two criteria. Figure 7, depicts a simple process, where a forecast of the number of errors left is used as a tollgate for deciding, whether or not to stop testing.

6.4 Quality Decisions & Process Improvement

Contrasting the actual against the planned provides another opportunity to detect problems. If the comparison shows that errors are not being discovered as fast or as many as originally anticipated, is it because the quality of our software development process has improved or is testing sloppy? And if more errors than anticipated are discovered, is it because the quality of our software has deteriorated or our testing process become more efficient? Is the testing process exhibiting a defined pattern or is it showing an erratic behaviour?

The models do not contain enough information to help explain the causes of a problem; moreover, they are not even able to tell us if there is a problem. The only thing the models will do is to make a trend explicit, but it is up to the user of the RGMs to interpret its meaning. That is why, is so important to understand the models and to interpret their output in the context of expert judgement and in concert with complementary metrics.

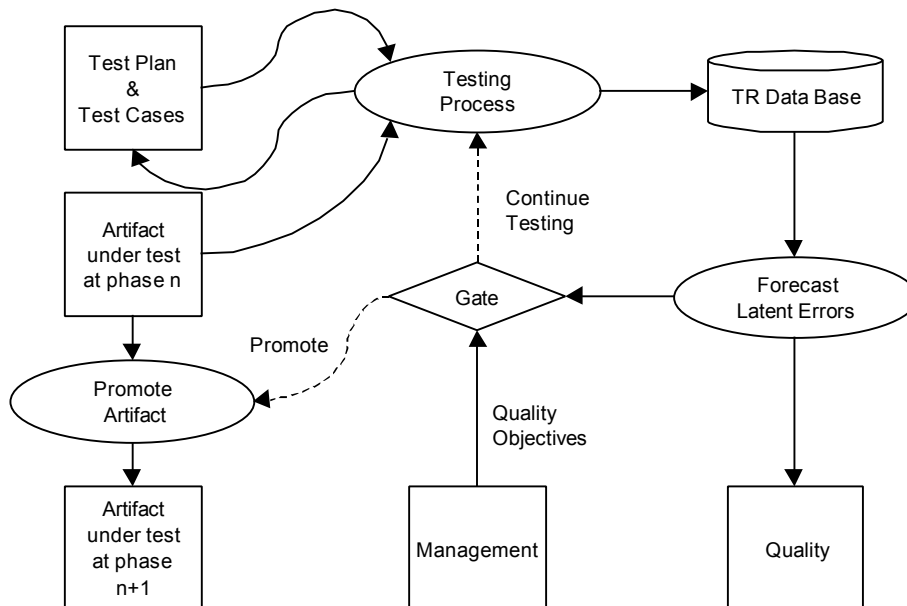


Figure 7 Toll Gate process based on EPM's predictions

7 Conclusion

At Ericsson Research Canada, RGMs are proving to produce better estimations than the experts and decisions made based on outputs from the EPM tools are slowly gaining acceptance throughout all levels of engineering and management.

The adoption process is not instantaneous, and sometimes, the lack of understanding of the strengths and limitations of the approach puts at risk all the progress made so far, so it is important to monitor and guide the diffusion process very carefully.

More accurate models can obviously help, but for project management applications, what is needed is complementary models. Models that we can use to estimate the number of test cases we need to develop, how to estimate the margin of error, etc.

References:

- [1] The Logic of Failure, D. Doner, Addison-Wesley, 1996
- [2] Metrics and Models in Software Quality Engineering, S. Kan, 1995, Addison-Wesley
- [3] Predicting Software Reliability, A. Woods, IEEE Computer November 1996, IEEE
- [4] On Predicting Software Related Performance of Large-Scale Systems, J. Gaffney, CMG XV, San Francisco 1984