A Generalized Software Reliability Model Considering Uncertainty and Dynamics in Development

Kiyoshi Honda, Hironori Washizaki, Yoshiaki Fukazawa

Waseda University, 3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
khonda@ruri.waseda.jp, {washizaki, fukazawa}@waseda.jp

Abstract. Development environments have changed drastically in recent years. The development periods are shorter than ever and the number of team has increased. These changes have led to difficulties in controlling the development activities and predicting the end of developments. In order to assess recent software developments, we propose a generalized software reliability model based on a stochastic process, and simulate developments that include uncertainties and dynamics, such as unpredictable requirements changes, shortening of the development period, and decrease in the number of members. We also compare our simulation results to those of other software reliability models. Using the values of uncertainties and dynamics obtained from our model, we can evaluate the developments in a quantitative manner.

Keywords: Software Reliability Model, Prediction of bugs, Stochastic Process

1 Introduction

The logistic curve and Gompertz curve[1] are well-known software reliability growth curves. However, these curves cannot account for the dynamics of software development. Developments are affected by various elements of the development environment, such as the skills of the development team and changing requirements. Examples of the types of software reliability models include " Times Between Failures Models " and " Failure Count Models. " [2] We use the " Failure Count Model, " which is based on counting failures and using probability methods. This type of models is represented by the Goel-Okumoto NHPP Model and the Musa Execution Time Model. Most models of this type cannot account for the dynamics of development, such as drastic changes in the members of a development team or significant reductions of the development time. However, our approach can handle these dynamic elements and simulate developments more accurately.

Recent studies by Tamura[3], Yamada[4] and Zhang[5] have attempted to describe the dynamics of developments using stochastic differential equations. These studies only use linear stochastic differential equations, but our approach uses non-linear stochastic differential equations, leading to more elaborate equations that can model situations more realistically. Our model can quantify uncertainties that are influenced by random factors such as the skills of teams and development environments. The quantification of uncertainties is important for predicting the end of developments more accurately and for optimizing the development teams or environments.

2 Generalized Software Reliability Model

For our software reliability model, we extend a non-linear differential equation that describes fault content as a logistic curve to an Ito type stochastic differential equation. We start with the following equation, which is called the logistic differential equation.

$$\frac{dN(t)}{dt} = N(t)(a + bN(t)) \tag{1}$$

The $N(t)$ is the number of detected faults at t, a defines the growth rate and the b is the carrying capacity[1].

We extend equation (1) to a stochastic differential equation because actual developments do not correctly obey equation (1) due to numerous uncertainties and dynamic changes. We consider such dynamic elements to be time-dependent and to contain uncertainty, and express them using a. The time-dependence of a can be used to describe situations such as skill improvements of development members and increases of growth rate. The uncertainty of a can describe parameters such as the variability of development members. We analyze the growth of software with a focus on the test phase by simulating the number of tested cases. We assume software development to have the following properties.

1. The total number of bugs is constant.
2. The number of bugs that can be found is variable depending on time.
3. The number of bugs that can be found contains uncertainty, which can be simulated with Gaussian white noise.

Considering these properties, we extend equation (1) to an Ito type stochastic differential equation with $a(t) = \alpha(t) + \sigma dw(t)$ as shown below.

$$dN(t) = (\alpha(t) + \sigma^2/2 + \beta N(t))N(t)dt + \gamma(t) \tag{2}$$

$N(t)$ is the number of tested cases at t, $\alpha(t) + \sigma^2/2 + \sigma dw(t)$ is the differential of the number of tested cases per unit time, $\gamma(t) = N(t)\sigma dw(t)$ is the uncertainty term, σ is the dispersion, β is the carrying capacity term which is non-linear. This equation has two significant terms, α and dw; α affects the end point of development, and dw affects the growth curve through uncertainties, especially $dw(t)$ relates $N(t)$, this means uncertainties depend on the number of tested cases. We compare the $\gamma(t) = N(t)\sigma dw(t)$ with other two types, $\gamma(t) = \sigma dw(t)$, not related with $N(t)$, and $\gamma(t) = 1/N(t)\sigma dw(t)$, related with inverse of $N(t)$. We vary these two terms, $\alpha(t)$ and the coefficient of $dw(t)$, and simulate models using equation (2). We summarize the types of $\alpha(t)$ and of the coefficient of $dw(t)$ and the corresponding situations in Table 1. Using our model, it is necessary to choose the types in Table 1 and calculate the parameters by using past data.

3 Simulation and Discussion

Three of the cases in Table 1 are modeled and plotted in Fig. 1. The difference between these three models is the parameter $\alpha(t)$. Based on **Model 1**, we defined

Table 1. $\alpha(t)$ is the number of tested cases per unit time. $dw(t)$ is the uncertainty term.

	$\gamma(t) = N(t)\sigma dw(t)$	$\gamma(t) = \sigma dw(t)$	$\gamma(t) = 1/N(t)\sigma dw(t)$
$\alpha_1(t) = a_1(\text{const.})$	The number of tested cases per unit time is constant, and the uncertainty increase near to the end. This model is similar to a logistic curve. (**Model 1**)	The number of tested cases per unit time is constant, and the uncertainty is constant at any given time.	The number of tested cases per unit time is constant, and the uncertainty is greater at the start of the project than at the end (e.g. the team matures over time).
$\alpha_2(t) = a_2(t < t_1)$ $\alpha_2(t) = a_3(t \geq t_1)$	The number of tested cases per unit time changes at t_1, and the uncertainty increases near to the end (e.g. new members join the project at time t_1). (**Model 2**)	The number of tested cases per unit time changes at t_1, and the uncertainty is constant at any given time.	The number of tested cases per unit time changes at t_1, and the uncertainty is greater at the start of the project than at the end.
$\alpha_3(t) \propto t$	Both the number of tested cases per unit time and the uncertainty increase near to the end (e.g. increasing manpower with time). (**Model 3**)	The number of tested cases per unit time increases, and the uncertainty is constant at any given time.	The number of tested cases per unit time increases, and the uncertainty is greater at the start of project than at the end.

that $a_2 = a_1$, $a_3 = 2a_1$ and $t_1 = t_{max}/2$ in **Model 2**, and $\alpha_3(t) = a_1 t$ in **Model 3**. The situation corresponding to **Model 2** is that at time t_1 the number of members of the development team doubles. The situation corresponding to **Model 3** is that the members' skills improve over time, effectively doubling the manpower by the time t_{max}. The purpose of the simulations is to confirm that our approach can assess software reliability under dynamic changes and uncertainties in development, and that it can adapt to the three models above and produce appropriate results. We use a Monte Carlo method to examine these models.

In **Model 1**, the number of tested cases per unit time is constant, and the uncertainty increases near to the end. As we predicted, the simulation result for **Model 1** fits the logistic curve. This result cannot be obtained simply by using other stochastic models that do not include a non-linear term.

In **Model 2**, the number of tested cases per unit time changes at t_1, and the uncertainty increases near to the end. In agreement with our predictions, the resulting curve sharply rises at t_1 and then converges quickly. Other models cannot describe such a time-dependent curve involving a non-linear term.

In **Model 3**, both the number of tested cases per unit time and the uncertainty increase near to the end. We expected the resulting curve to show a steeper increase than **Model 1**, but that was not the case. The reason for this is that the non-linear term pulls the curve down because of the increasing growth rate.

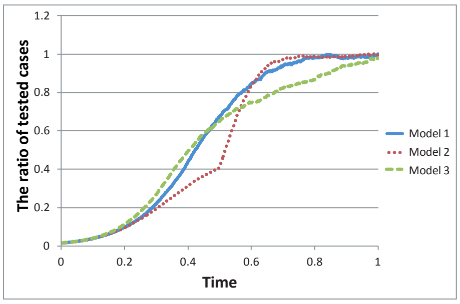

Fig. 1. The ratio of the total number of tested cases at time t to the total number of tested cases for the entire project is plotted. The x-axis represents time in arbitrary units, where 1 corresponds to t_{max} and 0.5 tot_1. In **Model 1**, the number of tested cases per unit time is constant. In **Model 2**, the number of tested cases per unit time changes at t_1. In **Model 3**, the number of tested cases per unit time increases.

4 Conclusion and Feature work

Using our model, we were able to simulate developments containing uncertainties and dynamic elements. We obtained the time-dependent logistic curve and growth curve, which was not possible using other models. Our model can be used to predict the end of projects where team members drastically change during development.

For future work, we will propose ways to quantitatively evaluate teams or team members taking uncertainties into account, and to optimize the teams to suit particular projects using our model. By using the past data, we can calculate the uncertainties of our model and predict the end of the project.

References

1. Yamada, S.,et al. : S-Shaped Reliability Growth Modeling for Software Error Detection. Reliability, IEEE Transactions on R-32, Issue: 5 475 - 484 (1983)
2. GOEL, A. :Software Reliability Models: Assumptions, Limitations, and Applicability. IEEE TOSE, VOL. SE-11, NO. 12 (1985)
3. Tamura, Y., et al. :A exible stochastic differential equation model in distributed development environment. EJOR 168, 143152,(2006)
4. Yamada, S., et al. :Software Reliability Measurement and Assessment with Stochastic Differential Equations. IEICE TFECCS E77-A(1), 109-116, (1994)
5. Zhang, N., et al. :A Stochastic Software Reliability Growth Model with Learning and Change-point.CDC' (2010)