Statistical Maintenance Time Estimation Based on Stochastic Differential Equation Models in OSS Development Project

Hironobu Sone[1], Yoshinobu Tamura[2], Shigeru Yamada[3]

[1]Tokyo City University, 1-28-1 Tamazutsumi, Setagaya-ku, Tokyo, 158-8557, Japan

[2]Tottori University, 4-101 Minami, Koyama, Tottori-shi, 680-8552, Japan

[1]g1881827@tcu.ac.jp, [2]tamuray@tcu.ac.jp, [3]yamada@tottori-u.ac.jp

Abstract

At present, the method of earned value management is often applied to the actual software projects in various IT companies. Also, open source software (OSS) are used under the various situations, because the OSS are useful for many users to make a cost reduction, standardization of systems, and quick delivery. Many OSS are developed under the peculiar development style known as bazaar method. According to the bazaar method, many faults are detected and fixed by developers around the world, and the fixed result will be reflected in the next release. In this paper, we discuss an OSS effort estimation model by using a conventional stochastic differential equation model. Moreover, we propose an optimal maintenance problem based on the proposed effort estimation model. Then, we discuss the optimal maintenance problem minimizing the maintenance effort and satisfying the earned value requirement, simultaneously. In addition, we also propose a method of judging whether the optimal maintenance time is an appropriate time from the viewpoint of the transition probability distribution of the cumulative number of maintenance effort, because proper management of maintenance effort affects software quality. Furthermore, several numerical examples of optimal maintenance time problem with earned value requirement are shown by using the effort data under actual OSS project.

Keywords: Maintenance Effort, Reliability, Stochastic Differential Equation, Earned Value Management, Transition Probability Distribution, Open Source Project

1. Introduction

Source codes of open source software (OSS) are freely available for use, reuse, fix and re-distribution by the OSS users. Many OSS are known as high performance and reliability although they are free of charge. Also, many IT companies often develop OSS for commercial use, because of cost reduction, standardization of systems, and quick delivery. In particular, OSS are developed by using the bazaar method [1]. Then, the source code is implemented in public through the Internet. OSS are promoted by an unspecified number of users and developers. The bug tracking system is also known as one of the systems used to develop OSS. Many fault information such as fix status, their details, and fix priorities are recorded through the bug tracking system.

Recently, an EVM (Earned Value Management) [2] is applied to the actual software projects under various IT companies. Also, many OSS are developed and managed by using fault big data recorded on the bug tracking systems. Considering the quality management of OSS project, many OSS are developed and maintained by several developers with many OSS users. The methods for reliability assessment of proprietary software and OSS have been proposed by several researchers [3-21]. However, the researches focused on the software effort expenditures of OSS have not been proposed. In particular, it is important to appropriately control the quality according to the progress status of OSS project. Also, the appropriate control of management effort for OSS will indirectly link to the quality, reliability, and cost. Moreover, it is useful for OSS project managers to decide the version upgrade if they can estimate the optimal maintenance time.

In this paper, we propose a new OSS effort estimation model by using the conventional stochastic differential equation model based on the exponential model and the delayed S-shaped model [17] derived from the software reliability growth models [18-21] considering the characteristics of open source projects. We assume that the number of developers and users change with irregularity. We can easily discover the regularity from various factors in open source projects. Then, we can apply the mathematical models with multiple parameters. However, it is difficult to actually use these models in terms of parameter estimation. In this paper, we apply a stochastic differential equation model with noise based on the Wiener process considering the specific circumstances of open source projects. The proposed model will be able to evaluate the project quantitatively considering external factors from indirectly in open source projects. Also, this paper discusses a useful method of OSS project management based on the earned value analysis considering the irregular fluctuation of performance resulting from the characteristics of OSS development and management. Moreover, we formulate the optimal maintenance time problem based on the proposed effort estimation model. We extend it to the optimization problem with requirements specification based on the proposed model by stochastic differential equation. Then, we find the optimum maintenance time by minimizing the total expected software maintenance effort. In, addition, we have verified whether the optimum maintenance time is suitable or not in terms of the cumulative number of reported faults, because proper management of maintenance effort affects software quality. Furthermore, several numerical examples of earned value analysis and effort optimization based on the proposed method are shown by using the effort data under actual OSS project. Especially, we focus on OSS development using the bug tracking system under operational phase in Figure 1.

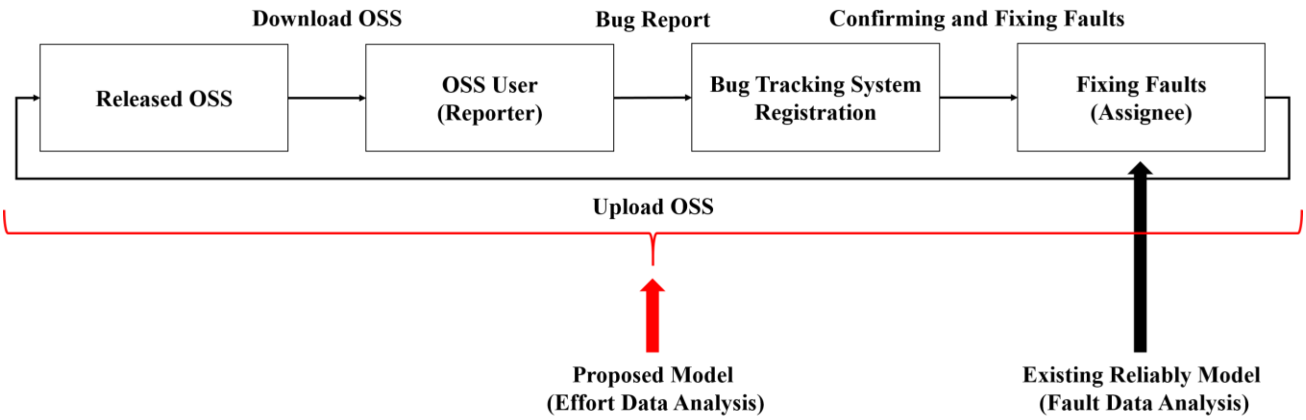

Figure 1: The OSS development using bug tracking system.

2. Wiener process models for OSS stability assessment

Considering the characteristic of the fault fixing in open source projects, the time-dependent effort expenditure phenomenon keeps an irregular state because there is variability among the levels of developer and skill through version-upgrade. In particular, OSS are developed and maintained by several developers and users.

Considering the characteristic of the operation phase in OSS projects, the time-dependent effort expense phenomenon of operation phase keeps an irregular state. Then, the time-dependent effort expenditure phenomenon of maintenance phase becomes unstable. Thus, the operation phases of many OSS projects are influenced from external factors by triggers such as the difference of skill of developers and users, time lags of development and maintenance activities. From above points, we focus on the stochastic differential equation modeling to managing the OSS project. Then, let $\Omega(t)$ be the cumulative maintenance effort up to operational time $t(t \geq 0)$ in the OSS project. Suppose that $\Omega(t)$ takes on continuous real values. Since the maintenance effort expenditures are observed during the operational phase of the OSS project, $\Omega(t)$ gradually increases as the operational procedures go on. Based on software reliability growth modeling approach [18-21], the following linear differential equation in terms of maintenance effort can be formulated:

$$\frac{d\Omega(t)}{dt} = \beta(t)\{\alpha - \Omega(t)\}, \tag{1}$$

where $\beta(t)$ is the increase rate of maintenance effort at operational time t and a non-negative function, and α means the estimated maintenance effort expenditures required until the end of operation.

Therefore, we extend Eq. (1) to the following stochastic differential equation with Brownian motion [22]:

$$\frac{d\Omega(t)}{dt} = \{\beta(t) + \sigma\nu(t)\}\{\alpha - \Omega(t)\}, \tag{2}$$

where σ is a positive constant representing a magnitude of the irregular fluctuation, and $\nu(t)$ a standardized Gaussian white noise. By using Itô's formula [23], we can obtain the solution of Eq. (2) under the initial condition $\Omega(0) = 0$ as follows:

$$\Omega(t) = \alpha[1 - \exp\{-\int_0^t \beta(s)ds - \sigma\omega(t)\}], \tag{3}$$

where $\omega(t)$ is one-dimensional Wiener process which is formally defined as an integration of the white noise $\nu(t)$ with respect to time t. Moreover, we define the increase rate of maintenance effort expenditures in case of $\beta(t)$ defined as [24]:

$$\beta(t) \doteq \frac{\frac{dF_*(t)}{dt}}{\alpha - F_*(t)}. \tag{4}$$

In this paper, we assume the following equations $F_*(t)$'s based on software reliability growth models as $F_*(t)$'s the cumulative maintenance effort function of the proposed model:

$$F_e(t) \equiv \alpha\left(1 - e^{-\beta t}\right), \tag{5}$$

$$F_s(t) \equiv \alpha\left\{1 - (1 + \beta t)e^{-\beta t}\right\}. \tag{6}$$

where $\Omega_e(t)$ means the cumulative maintenance effort expenditures for the exponential software reliability growth model with $F_e(t)$. Similarly, $\Omega_s(t)$ is the cumulative maintenance effort expenditures for the delayed S-shaped software reliability growth model with $F_s(t)$.

Therefore, the cumulative maintenance effort up to time t are obtained as follows:

$$\Omega_e(t) = \alpha[1 - \exp\{-\beta t - \sigma\omega(t)\}], \tag{7}$$

$$\Omega_s(t) = \alpha[1 - (1 + \beta t)\exp\{-\beta t - \sigma\omega(t)\}]. \tag{8}$$

In this model, we assume that the parameter σ depends on several noises by external factors from several triggers in open source projects. Then, the expected cumulative maintenance effort expenditures spent up to time t are respectively obtained as follows:

$$\mathrm{E}[\Omega_e(t)] = \alpha[1 - \exp\{-\beta t + \frac{\sigma^2}{2}t\}], \tag{9}$$

$$\mathrm{E}[\Omega_s(t)] = \alpha\left[1 - (1 + \beta t)\exp\left\{-\beta t + \frac{\sigma^2}{2}t\right\}\right]. \tag{10}$$

Similarly, we consider the sample path of maintenance effort expenditures required for OSS maintenance, e.g., the needed remaining maintenance effort expenditures from time t to the end of the project are obtained as follows:

$$\Omega_{re}(t) = \alpha\exp\{-\beta t - \sigma\omega(t)\}, \tag{11}$$

$$\Omega_{rs}(t) = \alpha(1 + \beta t)\exp\{-\beta t - \sigma\omega(t)\}. \tag{12}$$

Then, the expected maintenance effort expenditures required for OSS maintenance until the end of operation time t are respectively obtained as follows:

$$\mathrm{E}[\Omega_{re}(t)] = \alpha\exp\{-\beta t + \frac{\sigma^2}{2}t\}, \quad (13)$$

$$\mathrm{E}[\Omega_{rs}(t)] = \alpha(1+\beta t)\exp\left\{-\beta t + \frac{\sigma^2}{2}t\right\}. \quad (14)$$

3. Assessment Measures for EVM

Considering the characteristic of the operation phase of OSS, the method of EVM is used for the project management under various IT companies. Generally, the EVM is applied to commonly software development projects. However, it is difficult to directly apply the EVM to the actual OSS project, because the development cycle of OSS project is different from the traditional software development paradigm. As the characteristics of OSS, the OSS development project is managed by using the bug tracking system. This paper proposes the method of earned value analysis for OSS projects by using the data sets obtained from bug tracking system.

Considering the earned value analysis for OSS, we assume the following terms as the EVM for OSS:

- Actual Cost (AC): Cumulative maintenance effort up to operational time t considering the reporter and assignee
- Earned Value (EV): Cumulative maintenance effort up to operational time t considering the reporter
- Cost Variance (CV): Fixing effort required for OSS maintenance up to operational time t, $\mathrm{E}[\theta_e(t)]$ and $\mathrm{E}[\theta_s(t)]$
- Cost Performance Index (CPI): $\mathrm{E}[CPI_e(t)]$ and $\mathrm{E}[CPI_s(t)]$ obtained from AC and EV
- Estimate at Completion (EAC): $\mathrm{E}[EAC_e(t)]$ and $\mathrm{E}[EAC_s(t)]$ obtained from AC, EV, CPI, and BAC
- Estimate to Completion (ETC): $\mathrm{E}[ETC_e(t)]$ and $\mathrm{E}[ETC_s(t)]$ obtained from AC, EV, CPI, BAC, and EAC
- Budget at Completion (BAC): Planned Value (PV) in the end point as the specified goal of OSS project

Then, the expected fixing effort required for OSS maintenance up to operational time t in case of $\Omega_e(t)$ and $\Omega_s(t)$ can be formulated as:

$$\mathrm{E}[\theta_e(t)] = \mathrm{E}[\Omega^{r}_{re}(t)] - \mathrm{E}[\Omega^{ra}_{re}(t)], \quad (15)$$

$$\mathrm{E}[\theta_s(t)] = \mathrm{E}[\Omega^{r}_{rs}(t)] - \mathrm{E}[\Omega^{ra}_{rs}(t)]. \quad (16)$$

where $\mathrm{E}[\Omega^{ra}_{re}(t)]$ and $\mathrm{E}[\Omega^{ra}_{rs}(t)]$ are the expected maintenance effort expenditures considering the reporter and assignee until the end of operation. Also, $\mathrm{E}[\theta_e(t)]$ and $\mathrm{E}[\theta_s(t)]$ mean the maintenance effort expenditures considering the reporter until the end of operation. Similarly, the sample path of fixing effort expenditures required for OSS project maintenance up to operational time t in case of $\Omega_e(t)$ and $\Omega_s(t)$ are given by

$$\theta_e(t) = \Omega^{r}_{re}(t) - \Omega^{ra}_{re}(t), \quad (17)$$

$$\theta_s(t) = \Omega^{r}_{rs}(t) - \Omega^{ra}_{rs}(t). \quad (18)$$

The zero point of fixing effort $\mathrm{E}[\Omega_e(t)]$ and $\mathrm{E}[\Omega_s(t)]$ mean the starting point of surplus effort. On the other hand, the zero point of fixing effort $\mathrm{E}[\Omega_e(t)]$ and $\mathrm{E}[\Omega_s(t)]$ is the end point of effort expenditures required in OSS project. Therefore, the OSS project managers will be able to judge the necessity of fixing effort and stability of OSS from the starting point of surplus effort. Moreover, we can obtain the CPI by using the following equations:

$$\mathrm{E}[CPI_e(t)] = \frac{\mathrm{E}[\Omega^{r}_{re}(t)]}{\mathrm{E}[\Omega^{ra}_{re}(t)]}, \quad (19)$$

$$\mathrm{E}[CPI_s(t)] = \frac{\mathrm{E}[\Omega^{r}_{rs}(t)]}{\mathrm{E}[\Omega^{ra}_{rs}(t)]}. \quad (20)$$

Similarly, the sample path of CPI in case of Ω_e and Ω_s are given by

$$CPI_e(t) = \frac{\Omega_{re}^{r}(t)}{\Omega_{re}^{ra}(t)}, \tag{21}$$

$$CPI_s(t) = \frac{\Omega_{rs}^{r}(t)}{\Omega_{rs}^{ra}(t)}. \tag{22}$$

Furthermore, we can obtain the EAC by using the following equations:

$$\mathrm{E}[EAC_e(t)] = \mathrm{E}[\Omega_{re}^{ra}(t)] + \frac{(BAC - \mathrm{E}[\Omega_{re}^{r}(t)])}{\mathrm{E}[CPI_e(t)]}, \tag{23}$$

$$\mathrm{E}[EAC_s(t)] = \mathrm{E}[\Omega_{rs}^{ra}(t)] + \frac{(BAC - \mathrm{E}[\Omega_{rs}^{r}(t)])}{\mathrm{E}[CPI_s(t)]}. \tag{24}$$

Similarly, the sample path of EAC in case of Ω_e and Ω_s are given by

$$EAC_e(t) = \Omega_{re}^{ra}(t) + \frac{\left(BAC - \Omega_{re}^{r}(t)\right)}{CPI_e(t)}, \tag{25}$$

$$EAC_s(t) = \Omega_{rs}^{ra}(t) + \frac{\left(BAC - \Omega_{rs}^{r}(t)\right)}{CPI_s(t)}. \tag{26}$$

Finally, we can obtain the ETC by using the following equations:

$$\mathrm{E}[ETC_e(t)] = \frac{(BAC - \mathrm{E}[\Omega_{re}^{r}(t)])}{\mathrm{E}[CPI_e(t)]}, \tag{27}$$

$$\mathrm{E}[ETC_s(t)] = \frac{(BAC - \mathrm{E}[\Omega_{rs}^{r}(t)])}{\mathrm{E}[CPI_s(t)]}. \tag{28}$$

Similarly, the sample path of ETC in case of Ω_e and Ω_s are given by

$$ETC_e(t) = \frac{\left(BAC - \Omega_{re}^{r}(t)\right)}{CPI_e(t)}, \tag{29}$$

$$ETC_s(t) = \frac{\left(BAC - \Omega_{rs}^{r}(t)\right)}{CPI_s(t)}. \tag{30}$$

In particular, the CPI is very important for OSS project managers to assess the stability of OSS project.

4. Optimum Maintenance Time Problem Based on Wiener Process Models

This section discusses the optimal maintenance time problem by minimizing the maintenance effort expenditures for the operation of OSS. Then, we define the following effort rate parameters:

e_1 : the maintenance effort per effort needed to operate OSS,

e_2 : the operation effort per unit time during the operation,

e_3 : the maintenance effort per effort after the upgrade task such as major version upgrade.

Then, the expected maintenance effort expenditures in the operation of OSS can be formulated as:

$$E_1(t) = e_1 \mathrm{E}[\Omega_{*}^{ra}(t)] + e_2 t. \tag{31}$$

Also, the expected software maintenance effort expenditures after the maintenance of OSS is represented as follows:

$$E_2(t) = e_3 \mathrm{E}[ETC_*(t)]. \tag{32}$$

In particular, we consider ETC obtained from EVM as the software maintenance effort after the maintenance of OSS.

Consequently, from Eqs. (31) and(32), the total expected software maintenance effort expenditures during the specified period such as the specified version is given by

$$E(t) = E_1(t) + E_2(t). \tag{33}$$

The optimum maintenance time t^* is obtained by minimizing $E(t)$ in Eq. (33).

It is important for the OSS managers to find the optimal maintenance time considering the total expected software maintenance effort simultaneously with requirement specification such as CPI. This section focuses on the optimal maintenance problem with requirement specification. Also, we consider CPI as the requirement specification. Then, the optimal maintenance time problem with requirement specification is defined as follows:

$$\begin{aligned} &\min_t E(t), \\ &subject\ to\ CPI_*(t) \geq 1.0. \end{aligned} \tag{34}$$

5. Verification Method of Derived Optimum Maintenance Time

Although the optimal maintenance time can be derived using Eq.(34), there has been no research on whether the maintenance time is appropriate. In this paper, we judge whether the optimal maintenance time is an appropriate time from the viewpoint of the transition probability distribution of the number of faults reported. Specifically, we predict the cumulative number of reported faults reported using the equation in section 2. Then, the transition probability distribution of the cumulative number of reported faults is derived, and the time when most faults in OSS are reported is predicted. Finally, we examine the appropriateness of the optimal maintenance time by comparing the optimal maintenance time with the time when most faults in OSS are reported.

In the following formula, $\Omega(t)$ is the cumulative reported fault number up to operational time $t(t \geq 0)$ in the OSS project. Also, $\beta(t)$ is the increase rate of the number of reported faults at operational time t and a non-negative function, and α is the estimated number of reported faults required until the end of operation, and σ is a positive constant representing a magnitude of the irregular fluctuation.

Since the Wiener process $\omega(t)$ is a Gaussian process, $\log\{\alpha - \Omega(t)\}$ is also a Gaussian process. The mean of $\log\{\alpha - \Omega(t)\}$ are derived as follows:

$$\mathrm{E}[\log\{\alpha - \Omega_e(t)\}] = \log\alpha - \beta t, \tag{35}$$

$$\mathrm{E}[\log\{\alpha - \Omega_s(t)\}] = \log\alpha - \beta t + \log(1 + \beta t). \tag{36}$$

Also, The variance of $\log\{\alpha - \Omega(t)\}$ are derived as follows:

$$\mathrm{Var}[\log\{\alpha - \Omega_e(t)\}] = \sigma^2 t, \tag{37}$$

$$\mathrm{Var}[\log\{\alpha - \Omega_s(t)\}] = \sigma^2 t. \tag{38}$$

Thus, the following equation is derived:

$$\Pr[\log\{\alpha - \Omega_e(t)\} \leq x] = \Phi\left(\frac{x - \log\alpha + \beta t}{\sigma\sqrt{t}}\right), \tag{39}$$

$$\Pr[\log\{\alpha - \Omega_s(t)\} \leq x] = \Phi\left(\frac{x - \log\alpha + \beta t - \log(1 + \beta t)}{\sigma\sqrt{t}}\right), \tag{40}$$

where x is the cumulative fault fixing time at time t. Also, Φ means standard normal distribution and is defined as follows:

$$\Phi(x) = \frac{1}{\sqrt{2\sigma}} \int_{-\infty}^{x} \exp\left(-\frac{y^2}{2}\right) dy. \tag{41}$$

Considering the above points, the transition probability distributions of $\Omega_e(t)$ and $\Omega_s(t)$ are obtained as:

$$\Pr[\Omega(t) \leq n | \Omega_e(0) = 0] \quad = \quad \Phi\left(\frac{\log \frac{\alpha}{\alpha - n} - \beta t}{\sigma\sqrt{t}}\right), \tag{42}$$

$$\Pr[\Omega(t) \leq n | \Omega_s(0) = 0] \quad = \quad \Phi\left(\frac{\log \frac{\alpha}{\alpha - n} - \beta t + \log(1 + \beta t)}{\sigma\sqrt{t}}\right). \tag{43}$$

6. Application of proposed method to actual data

We discuss the applicability as a method for evaluating the stability of a project by applying actual open source project data to the proposed model.

6.1. Used data set

In this paper, we used one open source projects to derive the optimum maintenance time and assess the applicability of the time. For applying the proposed model to actual project data, we use the data of OpenStack [25] obtained from Bugzilla. OpenStack is open source software for cloud computing. This project use Bugzilla as open source bug tracking system. The information about reported faults is freely available from the bug tracking system. In particular, the effort data and fault data we obtained from Bugzilla are version 16(Pike). From the data used in this research, the cumulative number of reported faults is 2249. Also, each data is weekly unit data. Especially, we used only for fixed faults as the number of fix faults.

6.2. Application of the proposed method

We apply the maintenance effort and number of cumulative reported faults derived in Section 2 to OpenStack project data. In this paper, we have estimated the parameters by the method of maximum likelihood.

In this section, the estimation method of unknown parameters α, β, and σ in Eqs.(9) and (10) is presented. The joint probability distribution function of the process $\Omega(t)$ as

$$P(t_1, y_1; t_2, y_2; \cdots; t_K, y_K) \equiv \Pr[\Omega(t_1) \leq y_1, \cdots, \Omega(t_K) \leq y_K | \Omega(t_0) = 0]. \tag{44}$$

The probability density of Eq. (44) is denoted as

$$p(t_1, y_1; t_2, y_2; \cdots; t_K, y_K) \quad \equiv \quad \frac{\partial^K P(t_1, y_1; t_2, y_2; \cdots; t_K, y_K)}{\partial y_1 \partial y_2 \cdots \partial y_K}. \tag{45}$$

Since $E(t)$ takes on continuous values, the likelihood function, l, for the observed data $(t_k, y_k)(k = 1,2,\cdots,K)$ is constructed as follows:

$$l = p(t_1, y_1; t_2, y_2; \cdots; t_K, y_K). \tag{46}$$

For convenience in mathematical manipulations, the following logarithmic likelihood function is used:

$$L = \log l. \tag{47}$$

The maximum-likelihood estimates α^*, β^*, and σ^* are the values making L in Eq.(47) maximize. These can be obtained as the solutions of the following simultaneous likelihood equations:

$$\frac{\partial L}{\partial \alpha} = \frac{\partial L}{\partial \beta} = \frac{\partial L}{\partial \sigma} = 0. \tag{48}$$

6.3. Numerical examples for optimum maintenance time

Table 1 shows the results of parameter estimation of maintenance effort, and AIC (Akaike's Information Criterion). In terms of AIC, the exponential model fits better than the delayed S-shaped model. Also, the estimated cumulative fixing time α becomes higher than one of exponential model. In other words, we find that the exponential model is more pessimistically estimation than the delayed S-shaped model in terms of the data applied in this paper.

Table 1: Parameter estimates of maintenance effort in case of OpenStack.

		reporter and assignee (AC)		reporter (EV)	
		exponential	delayed S-shaped	exponential	delayed S-shaped
parameter	α	7.523×10^{3}	7.013×10^{3}	4.303×10^{3}	3.253×10^{3}
	β	1.807×10^{-2}	3.836×10^{-2}	1.008×10^{-2}	3.507×10^{-2}
	σ	1.781×10^{-2}	2.418×10^{-2}	6.963×10^{-3}	1.630×10^{-2}
	AIC	1265.904	1268.123	1015.074	1041.897

Figures 2 and 3 show the estimated maintenance effort required until the end of operation in case of $\Omega_e(t)$ and $\Omega_s(t)$. From Figures 2 and 3, we found that the estimated maintenance effort expenditures required at the end of operation in case of $\Omega_e(t)$ is negatively estimated than the case of $\Omega_s(t)$. Moreover, we show several numerical examples based on the optimal maintenance problems which are discussed in section 5. Figure 4 shows the estimated cumulative maintenance effort expenditures in case of exponential model, E[$E_e(t)$] and $E_e(t)$, respectively. Also, Figure 5 shows the estimated total software effort in case of S-shaped model, E[$E_s(t)$] and $E_s(t)$, respectively. In case of exponential model, we find that the optimum maintenance time is derived as $t^* = 3.408(177.8\text{ weeks})$ years in case of (BAC=1.2e+04) in Figure 4. In addition, in case of S-shaped model, the optimum maintenance time is derived as $t^* = 5.888(307.2\text{ weeks})$ years in case of (BAC=1.2e+04) in Figure5. Furthermore, the estimated cumulative effort expenditures with stability requirement in case of exponential model and delayed S-shaped model is shown in Figure 6 and 7, respectively. The time for CPI ≥ 1 was 1.363 years (71.140 weeks) for exponential model and 5.215 years (307.2 weeks) for S-shaped model. From the above results, the estimated optimal maintenance time is about $t^* = 3.408$ years (177.8 weeks) for exponential model and $t^* = 5.888$ years (307.2 weeks) for S-shaped model in case of considering CPI as stability requirement.

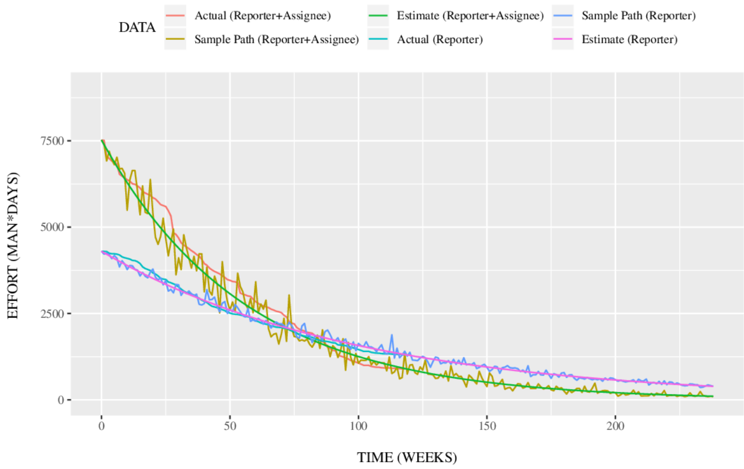

Figure 2: The remaining maintenance effort of OpenStack project using exponential model in Eqs.(11) and (13).

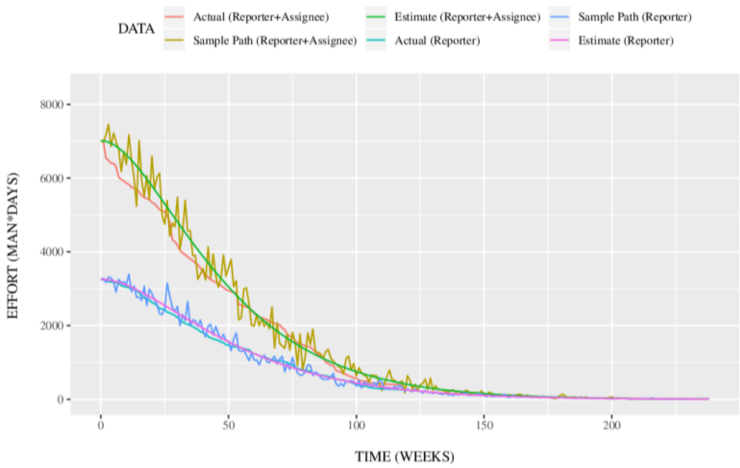

Figure 3: The remaining maintenance effort of OpenStack project using S-shaped model in Eqs.(12) and (14).

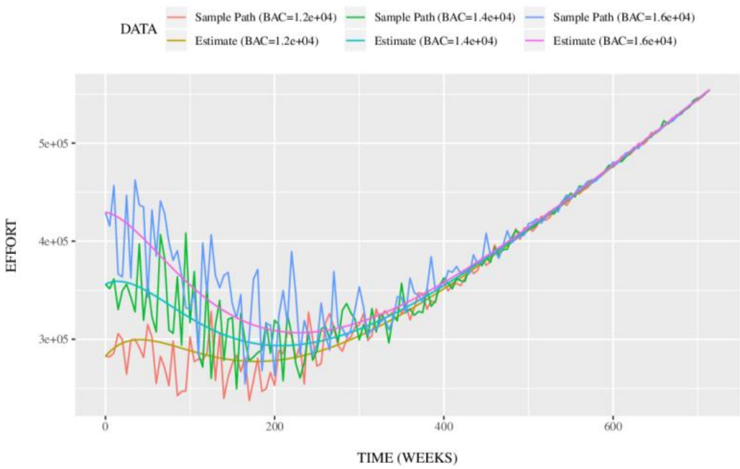

Figure 4: The estimated cumulative maintenance effort in case of E$[E_e(t)]$ and $E_e(t)$.

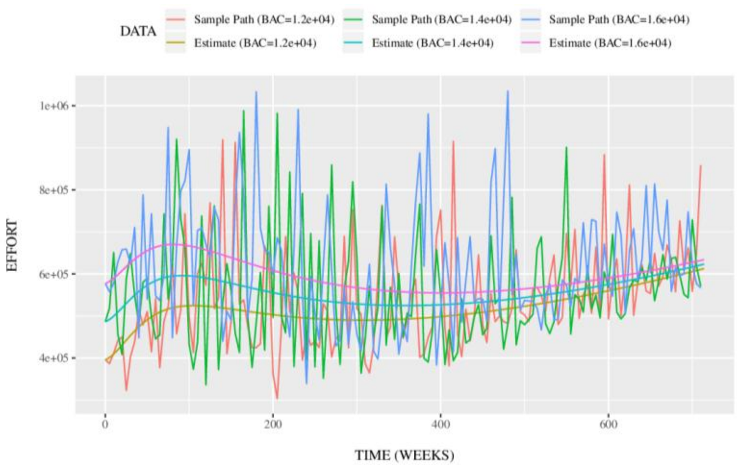

Figure 5: The estimated cumulative maintenance effort in case of E$[E_s(t)]$ and $E_s(t)$.

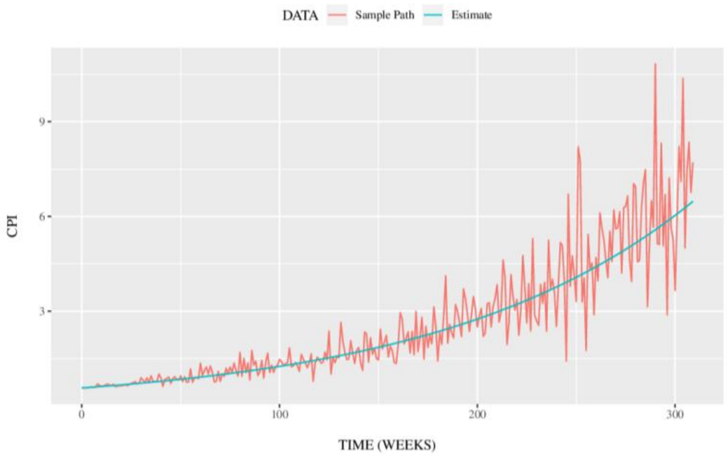

Figure 6: The estimated CPI in case of Eqs.(19) and (21).

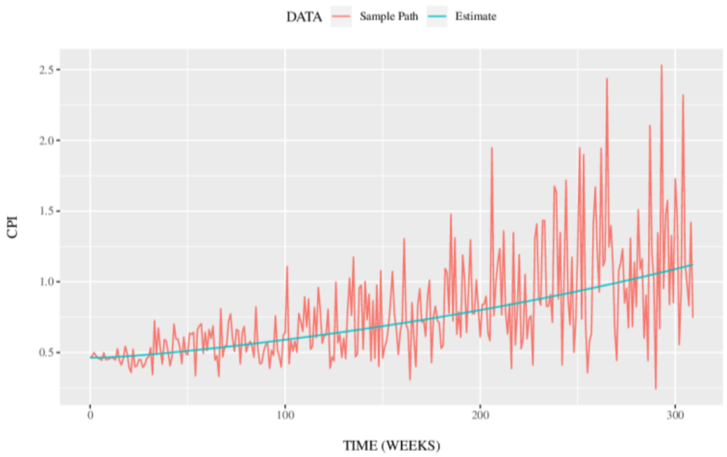

Figure 7: The estimated CPI in case of Eqs.(20) and (22).

6.4. Numerical examples considering the confidence interval of optimum maintenance time

In order to evaluate the performance of the optimum maintenance time derivation method in Section 7.3, we compare with two models for the cumulative number of faults. Table 2 shows the results of parameter estimation of cumulative number of reported faults, and AIC. In terms of AIC, the delayed S-shaped model fits better than the exponential model for model. In addition, Figures 8 and 9 show the estimated cumulative number of faults required until the end of operation in case of $\Omega_e(t)$ and $\Omega_s(t)$.

Table2: Parameter estimation in OpenStack.

		the number of reported faults of OpenStack	
		exponential	delayed S-shaped
parameter	α	2.513×10^3	2.441×10^3
	β	1.911×10^{-2}	3.550×10^{-2}
	σ	1.852×10^{-2}	1.953×10^{-2}
AIC		1011.621	994.440

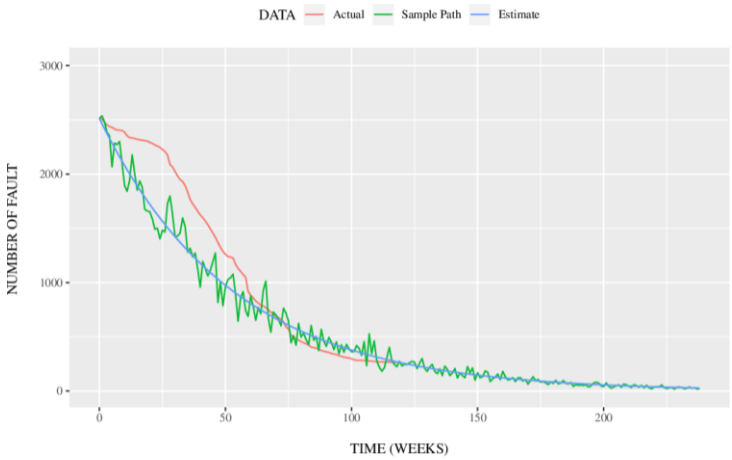

Figure 8: The number of remaining faults of OpenStack project using exponential model in Eqs.(11) and (13).

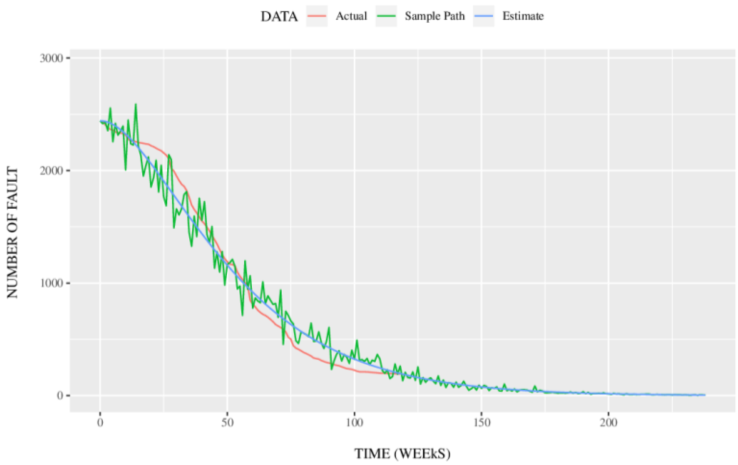

Figure 9: The number of remaining faults of OpenStack project using delayed S-shaped model in Eqs.(12) and (14).

For predicting the time when reported faults reach 50%, 70%, 90%, and 99%, transition probability distributions are shown in Figures 10 and 11 using Eqs. 42 and 43 respectively. In particular, in Figures 10 and 11, the time for the cumulative number of reported faults to reach 99% with a probability of 95% is 5.139 years (268.128

weeks) and 3.873 years (202.077 weeks), respectively. Since the estimated times are close to the optimum maintenance time derived in section 7.3, we can determine the optimum maintenance time as reasonable.

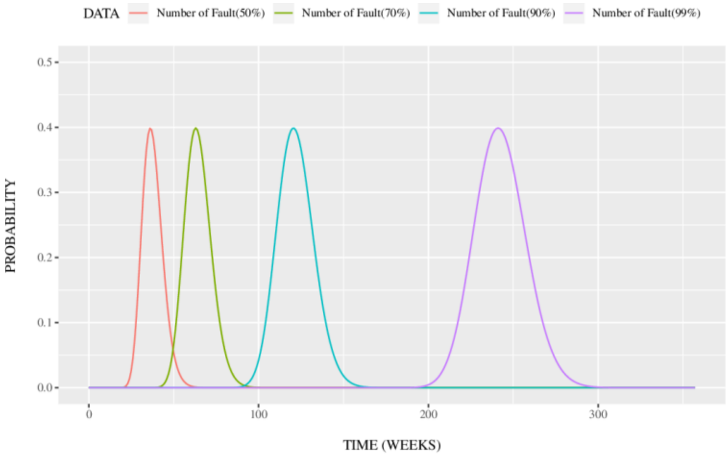

Figure 10: The transition probability distribution of the number of reported faults in case of Eq.(42).

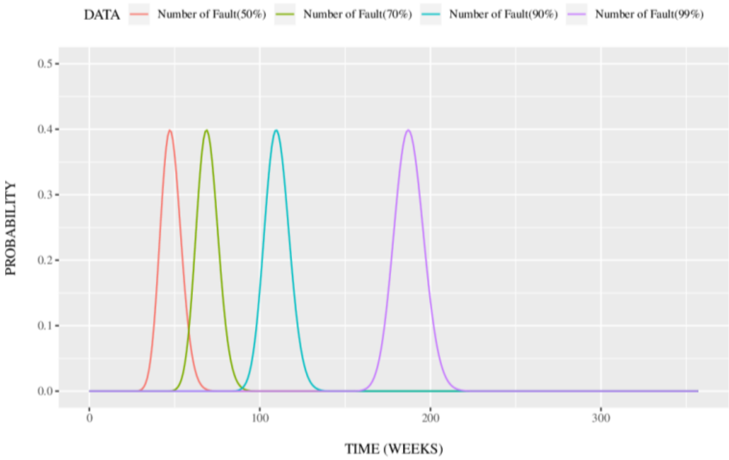

Figure 11: The transition probability distribution of the number of reported faults in case of Eq.(43).

For performance assessment of the optimum maintenance time derivation method in Section 6.3, we use the estimated total software effort using exponential model and transition probability using delayed S-shaped model, because of AIC. Figure 12 shows the result of the estimated cumulative maintenance effort expenditures using exponential model. Simultaneously, From Figure 12, the optimum maintenance time in case of three pattern respectively is roughly an appropriate prediction, because the optimum maintenance time is near the time when the transition probability of cumulative reported faults reaches 95%. In other words, the OSS maintains high quality at the optimum maintenance time.

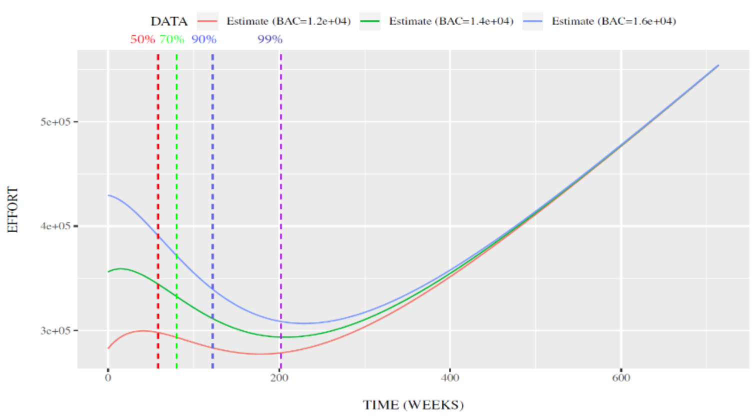

Figure 12: The estimated total software effort and the transition probability of cumulative reported faults.

From the above, it is possible to evaluate the optimum maintenance time considering the risk of system failure due to undiscovered faults by predicting the reporting status of potential faults.

7. Conclusions

The method of EV analysis considering various external factor associated with OSS project has been discussed in this paper. It will be important for the OSS project managers to control the progress of OSS project in terms of the OSS management effort. Also, the appropriate control of management effort for OSS will indirectly link to the quality, reliability, and cost reduction of OSS. In terms of OSS management effort, we have proposed the method of EV analysis based on the stochastic differential equation model. Moreover, it is important to decide the optimal length of version upgrade duration and maintenance time considering the operation status of OSS project management. The optimal maintenance time problem based on maintenance effort for OSS have been proposed in this paper. In particular, we have defined the optimal maintenance time problems considering the CPI as the stability and requirement specification.

In the past, it has been difficult to control the OSS quality and manage the project under the traditional method based on fault data for software reliability assessment. By focusing on the reporter and assignee on the fault big data recorded on the bug tracking system of OSS project, we have proposed a new approach based on the earned value and optimal maintenance time problem considering the project stability by using OSS effort data.

In particular, this paper has proposed the method of earned value analysis for OSS project considering the irregular fluctuation from the characteristics of OSS development and management. In addition, we have judged whether the optimal maintenance time is an appropriate time from the viewpoint of the transition probability distribution of the cumulative number of reported faults, because proper management of maintenance effort affects software quality. As a result, we derived the optimum maintenance time from the proposed models. The proposed method may be helpful as the assessment method of the progress of the OSS project in operation phase. Also, we have found that our method can assess the stability and effort control considering the operational environment of OSS. Moreover, the method of optimal maintenance time problem considering the OSS project stability in order to decide the optimal maintenance time has been discussed in this paper. Furthermore, the data set of actual OSS effort has been analysed to show numerical examples of the progress analysis and optimum maintenance time considering stability for an OSS project.

Data Availability

The data used in this research is obtained from Bugzilla.

Conflicts of Interest

The authors declare no conflict of interest.

Funding Statement

This research received no external funding.

References

1. Raymond, S. E.: The Cathedral and the Bazzar: Musings on Linux and Open Source by an Accidental Revolutionary", O'Reilly and Associates, Sebastopol, California, 1999.

2. Fleming, Q. E., Koppelman, J. M.: Earned Value Project Management (4th Ed.), PMI, Newton Square, U.S.A., 2010.

3. Yamada, S., Tamura, Y.: OSS Reliability Measurement and Assessment, Springer International Publishing, Switzerland, 2016.

4. Norris, J.: Mission-critical development with open source software, IEEE Software Magazine, 21(1), pp. 42-49, 2004.

5. Zhou, Y., Davis, J.: OSS reliability model: an empirical approach, Proceedings of the Fifth Workshop on OSS Engineering, pp. 67-72, 2005.

6. Sun, C., Lo, D., Wang, X,. Jiang, J., Khoo, S. A.: discriminative model approach for accurate duplicate bug report retrieval, Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10), Cape Town. South Africa. 2-8 May, pp.45-54, 2010.

7. Hooimeijer, P., Weimer, W.: Modeling bug report quality, Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering(ASE '07), Georgia. USA. 5-9 November, pp. 34-43, 2010.

8. Nurolahzade, M., Nasehi, M. S., Khandkar, H. S., Rawal, S.: The role of patch review in software evolution: an analysis of the mozilla firefox, Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops(IWPSE-Evol '09), Amsterdam. The Netherlands. 24-25 August, pp.9-18, 2009.

9. Tamura, Y., Sone, H., Yamada, S.: OSS Project Stability Assessment Support Tool Considering EVM Based on Wiener Process Models, Applied System Innovation, 2(1), pp.1-12, 2019.

10. Robles, G., Gonzälez-Barahona, M. J., Cervigön, C., Capiluppi, A., Izquierdo-Cortäzar, D.: Estimating development effort in Free/OSS projects by mining software repositories: a case study of OpenStack, Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad. India. 31 May-1 June, pp. 222-231, 2014.

11. Mishra, R., Sureka, A.: Mining Peer Code Review System for Computing Effort and Contribution Metrics for Patch Reviewers, Proceedings of the 2014 IEEE 4th Workshop on Mining Unstructured Data, Victoria. Canada. 30-30 September, pp. 11-15, 2014.

12. Rakha, S. M., Shang, W., Hassan, E. A.: Studying the needed effort for identifying duplicate issues, Empirical Software Engineering, Springer Science+Business Media: Berlin, Germany, 2015.

13. Giger, E., Pinzger, M., Gall, H.: Predicting the fix time of bugs, Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering, Cape Town. South Africa. 4-4 May, pp.52-56, 2010.

14. Bougie, G., Treude, C., German, M. D., Storey, A. M.: A comparative exploration of freeBSD bug lifetimes, Proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR 2010), Cape Town. South Africa. 2-3 May, pp.106-109, 2010.

15. Akbarinasaji, S., Caglayan, B., Bener, A.: Predicting Bug-Fixing Time: A Replication Study Using An OSS Project, Systems and Software, 136, pp.173-186, 2018.

16. Marks, L., Zou, Y., Hassan, E. A.: Studying the fix-time for bugs in large open source projects, Proceedings of the 7th International Conference on Predictive Models in Software Engineering (Promise '11), Alberta. Canada. 20-21 September, pp.11:1–11:8, 2011.

17. Yamada, S., Ohba, M., Osaki, S.: S-Shaped Reliability Growth Modeling for Software Error Detection, IEEE Transactions on Reliability, R-32(5), pp.475-484, 1983.

18. Yamada, S.: Software Reliability Modeling: Fundamentals and Applications, Springer-Verlag; Tokyo/Heidelberg, 2014.

19. Lyu, M.R. Ed.: Handbook of Software Reliability Engineering. IEEE Computer Society Press; Los Alamitos.; CA, U.S.A., 1996.

20. Musa, J.D., Iannino, A., Okumoto, K.: Software Reliability: Measurement, Prediction, Application. McGraw-Hill; New York, 1987.

21. Kapur, P.K., Pham, H., Gupta, A., Jha, P. C.: Software Reliability Assessment with OR Applications, Springer-Verlag; London, 2011.

22. Wong, E.: Stochastic Processes in Information and Systems. McGraw-Hill; New York, 1971.

23. Arnold L.: Stochastic Differential Equations-Theory and Applications. John Wiley & Sons; New York, 1971.

24. Yamada, S., Kimura, M., Tanaka, H., Osaki, S.: Software reliability measurement and assessment with stochastic differential equations, IEICE Transactions on Fundamentals, E77-A(1), 109-116, 1994.

25. The OpenStack Foundation, The OpenStack project, http://www.openstack.org/