


## Article

# OSS Project Stability Assessment Support Tool Considering EVM Based on Wiener Process Models

Yoshinobu Tamura <sup>1,\*</sup>, Hironobu Sone <sup>2,†</sup>  and Shigeru Yamada <sup>3,†</sup><sup>1</sup> Department of Industrial & Management Systems Engineering, Faculty of Knowledge Engineering, Tokyo City University, Tokyo 158-8557, Japan<sup>2</sup> Graduate School of Integrative Science and Engineering, Tokyo City University, Tokyo 158-8557, Japan; g1881827@tcu.ac.jp<sup>3</sup> Graduate School of Engineering, Tottori University, Tottori 680-8552, Japan; yamada@tottori-u.ac.jp

\* Correspondence: tamuray@tcu.ac.jp; Tel.: +81-3-5707-0104

† These authors contributed equally to this work.

Received: 6 December 2018; Accepted: 2 January 2019; Published: 4 January 2019



**Abstract:** The method of earned value management is traditionally applied to the progress assessment of actual software projects in many IT companies. Also, Open Source Software (OSS) are used under the various situations because of cost reduction, standardization, and quick delivery. In particular, many OSS are developed and managed by using fault big data recorded on the bug tracking systems. Several research papers in terms of OSS reliability assessment have been published in the past. The fault is caused by the poor handling of effort control. Therefore, we can make a stable and safety operation for OSS system if the OSS project is appropriately managed by using the software effort. The OSS effort estimation model by using a conventional stochastic differential equation model is discussed in this paper. Then, we propose the optimal maintenance problem based on the earned value requirement. In particular, we develop the OSS project stability support tool. Furthermore, several performance illustrations of the developed software tool are shown by using the effort data under actual OSS project.

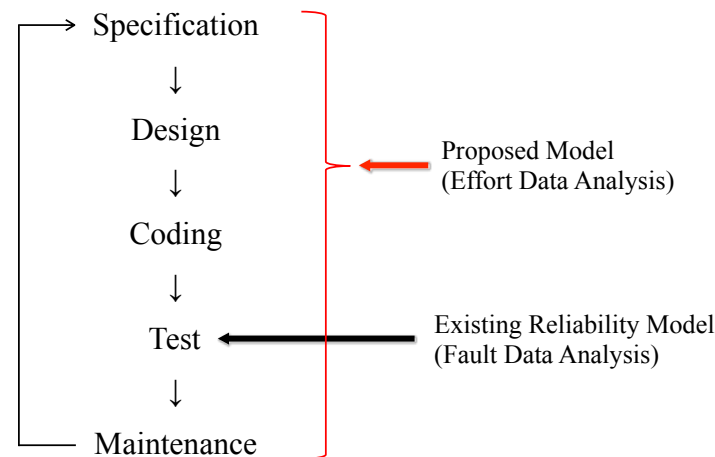
**Keywords:** earned value management; stochastic differential equation; maintenance effort; open source project

## 1. Introduction

Traditionally, the method of Earned Value Management (EVM) [1] is used in the progress assessment of actual software projects under various IT companies. Also, many Open Source Software (OSS) are managed in OSS project by using fault big data recorded on the bug tracking systems. Recently, OSS are used under the various situations, because OSS are useful for many users to make cost reduction, standardization, and quick delivery. Moreover, the methods for reliability assessment of OSS have been proposed by several researchers [2–4]. Also, our research group has proposed and discussed the method of reliability assessment for the proprietary software and OSS [5–8]. However, the researches focused on the software effort expenditures of OSS have been not proposed.

Moreover, the effective methods assisting dynamic testing management for OSS development paradigm as typified by the open source project are few in number [9–11]. Also, many OSS are used in various area such as cloud computing, mobile cloud, and server software. For example, several research papers [12–16] have been proposed in the area of cloud computing. However, these papers focus on the security, service optimization, secure control, resource allocation technique, etc. The research papers in terms of project management for clouds by using OSS are few in number.

In particular, it is important to appropriately manage the quality according to the progress status of OSS project, because the software failure is caused by the poor handling of effort control as shown in Figure 1. Figure 1 shows the difference in the data sets between the proposed method and the existing reliability assessment one. In particular, our method can totally assess the development phase by using the effort data. 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 maintenance time.



**Figure 1.** The characteristics of the proposed method.

In this paper, we discuss the OSS effort estimation model by using the stochastic differential equation model. 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 problem based on the proposed effort estimation model and 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 by using the cost performance index. In particular, we develop the OSS project stability support tool considering EVM. Furthermore, several performance illustrations of the OSS project stability support tool are shown by using the effort data in actual OSS project.

## 2. Wiener Process Models for Effort Estimation

Considering the characteristic of the operation phase in OSS projects, the time-dependent effort expenditure phenomenon of operation phase keeps an irregular state, because there is variability among the levels of developer and skill through version-upgrade, and the OSS are developed and maintained by several developers and users. 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 discuss the stochastic differential equation modeling for the OSS project. Then, let  $\Psi(t)$  be the cumulative maintenance effort up to operational time  $t$  ( $t \geq 0$ ) in the OSS project. Suppose that  $\Psi(t)$  takes on continuous real values. Since the estimated maintenance effort are observed during the operational phase of the OSS project,  $\Psi(t)$  gradually increases as the operational procedures go on. Based on the software reliability growth modeling approach [5–8], the following linear differential equation in terms of maintenance effort can be formulated:

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

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

Therefore, we extend Equation (1) to the following stochastic differential equation with Brownian motion [17]:

$$\frac{d\Psi(t)}{dt} = \{\beta(t) + \sigma v(t)\} \{\alpha - \Psi(t)\}, \quad (2)$$

where  $\sigma$  is a positive constant representing a magnitude of the irregular fluctuation, and  $v(t)$  a standardized Gaussian white noise. By using Itô's formula [18], we can obtain the solution of Equation (2) under the initial condition  $\Psi(0) = 0$  as follows:

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

where  $\omega(t)$  is one-dimensional Wiener process which is formally defined as an integration of the white noise  $v(t)$  with respect to time  $t$ . Moreover, we define the increase rate of maintenance effort in case of  $\beta(t)$  defined as [19]:

$$\int_0^t \beta(s)ds \doteq \frac{\frac{dF_*(t)}{dt}}{\alpha - F_*(t)}. \quad (4)$$

In this paper, we assume the following equations based on software reliability models  $F_*(t)$  as the cumulative maintenance effort function of the proposed model:

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

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

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

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

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

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

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

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

$$E[\Psi_s(t)] = \alpha[1 - (1 + \beta t)\exp\{-\beta t + \frac{\sigma^2}{2}t\}]. \quad (10)$$

Similarly, the sample path of fixing effort required for OSS maintenance until the end of operation time  $t$  is obtained as follows:

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

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

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

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

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

### 3. Assessment Measures for EVM

The method of EVM has been used for the project management under various IT companies. Generally, the EVM is applied to commonly software development projects [20]. 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. Considering the characteristic of the OSS operation phase, the OSS development project is managed by using the bug tracking system. The method of EVM will be helpful for OSS managers to assess QCD (Quality, Cost, Delivery) of OSS [21]. 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  $t$ ,  $E[\Theta_e(t)]$  and  $E[\Theta_s(t)]$ .
- ▶ Cost Performance Index (CPI):  $E[CPI_e(t)]$  and  $E[CPI_s(t)]$  obtained from AC and EV.
- ▶ Estimate at Completion (EAC):  $E[EAC_e(t)]$  and  $E[EAC_s(t)]$  obtained from AC, EV, CPI, and BAC.
- ▶ Estimate to Completion (ETC):  $E[ETC_e(t)]$  and  $E[ETC_s(t)]$  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  $\Psi_e(t)$  and  $\Psi_s(t)$  can be formulated as:

$$E[\Theta_e(t)] = E[\Psi_{re}^r(t)] - E[\Psi_{re}^{ra}(t)], \quad (15)$$

$$E[\Theta_s(t)] = E[\Psi_{rs}^r(t)] - E[\Psi_{rs}^{ra}(t)], \quad (16)$$

where  $E[\Psi_{re}^{ra}(t)]$  and  $E[\Psi_{rs}^{ra}(t)]$  are the maintenance effort considering the reporter and assignee until the end of operation. Also,  $E[\Psi_{re}^r(t)]$  and  $E[\Psi_{rs}^r(t)]$  mean the maintenance effort considering the reporter until the end of operation.

Similarly, the sample path of fixing effort required for OSS project maintenance up to operational time  $t$  in case of  $\Psi_e(t)$  and  $\Psi_s(t)$  are given by

$$\Theta_e(t) = \Psi_{re}^r(t) - \Psi_{re}^{ra}(t), \quad (17)$$

$$\Theta_s(t) = \Psi_{rs}^r(t) - \Psi_{rs}^{ra}(t). \quad (18)$$

The zero point of fixing effort  $E[\Theta_e(t)]$  and  $E[\Theta_s(t)]$  mean the starting point of surplus effort. On the other hand, the zero point of fixing effort  $E[\Theta_e(t)]$  and  $E[\Theta_s(t)]$  is the end point of effort 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:

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

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

Similarly, the sample path of CPI in case of  $\Psi_e(t)$  and  $\Psi_s(t)$  are given by

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

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

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

#### 4. Optimal Maintenance Time Based on Wiener Process Models Considering CPI as Requirement Specification

Several optimal software release problems based on software reliability models have been proposed by several researchers [22]. Considering the characteristics of the effort phenomena in OSS project, it is interesting for the software developers to predict and estimate the time when we should stop the operation in order to OSS version upgrade efficiently.

In the operation of OSS, we discuss the optimal maintenance problem by minimizing the maintenance effort. Then, we define the following effort rate parameters:

- $e_1$ : the importance rate of maintenance effort needed to operate OSS,
- $e_2$ : the operation effort per unit time during the operation,
- $e_3$ : the importance rate of maintenance effort after the upgrade task such as major version upgrade.

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

$$E_1(t) = e_1 E[\Psi_*^{ra}(t)] + e_2 t. \quad (23)$$

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

$$E_2(t) = e_3 \{\alpha - E[\Psi_*^{ra}(t)]\}. \quad (24)$$

Consequently, from Equations (23) and (24), the total expected software maintenance effort during the specified period such as the specified version is given by

$$E(t) = E_1(t) + E_2(t). \quad (25)$$

The optimum maintenance time  $t^*$  is obtained by minimizing  $E(t)$  in Equation (25).

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 [23]. Also, we consider CPI as the requirement specification. Then, the optimal maintenance problem with requirement specification is defined as follows:

$$\begin{aligned} & \min_t E(t) \\ & \text{subject to } CPI_*(t) \geq 1.0 \end{aligned} \quad (26)$$

Generally, the trend of stability in OSS project is unstable if the estimated value of CPI is smaller than 1.0. On the other hand, the trend of stability in OSS project becomes stable if the estimated value of CPI is larger than 1.0.

## 5. Framework of OSS Project Stability Support tool Considering EVM

The specification requirements of the OSS project stability support tool are shown as follows:

- ▶ This tool should be operated by clicking the mouse button and typing on the keyboard to input the data through GUI system. In particular, the user experience design is adopted as the important element of our tool.
- ▶ Statistical computing R for statistical computing and ggplot2 library of R package should be implemented to the source program. This tool is developed as a stand-alone of multi-platform. Also, this tool operates as Web application via the Web server.
- ▶ The method of maximum-likelihood is used as the estimation of unknown parameters in our model. Moreover, the optimum maintenance time is estimated by using the method of general-purpose optimization based on Nelder–Mead.
- ▶ This tool treats the proposed effort estimation model based on stochastic differential equation, and illustrate the cumulative maintenance effort at arbitrary time  $t$ , the CPI, and the total software maintenance effort.

It is well known the following keywords in terms of user experience design:

- ▶ Visual Design
- ▶ Information Architecture
- ▶ Information
- ▶ Structuring, Organization and Labeling
- ▶ Finding and Managing
- ▶ Interaction Design
- ▶ Usability
- ▶ Accessibility
- ▶ Human-Computer Interaction

We focus on the “Visual Design” and “Interaction Design” as the user experience design. “Visual Design” and “Interaction Design” can easily implement by using CSS3 and JavaScript programming language, because CSS3 and JavaScript includes the various effect components. We develop the OSS project stability support tool based on Visual Design” and “Interaction Design” by using the animation effects of CSS3 and JavaScript.

Moreover, the plotly R library (Plotly R Library) is used in the developed tool. In particular, the developed application software is implemented as the stand-alone application of multi-platform by using the latest technology of NW.js.

Furthermore, the procedures of EVM built into the developed tool are shown as follows:

1. This tool processes the data file in terms of the effort data in the OSS project.
2. Using the effort data obtained from the actual OSS project, we analyze the data for input data.
3. This tool estimates the unknown parameters included in our model. Then, the unknown parameters are estimated by using the method of maximum-likelihood.
4. This tool illustrates the estimated cumulative maintenance effort, fixing effort required for OSS maintenance until the end of operation, total expected software maintenance effort, and CPI as EVM measures.
5. We focus on the maintenance time based on Wiener process models considering CPI as requirement specification. In particular, the total software effort and the optimal software maintenance time minimizing the effort for our model are plotted in our tool.

## 6. Performance Illustrations Based on Developed Tool

This section focuses on the Apache Tomcat known as the OSS developed under Apache Software Foundation [24]. The project data used in this paper is obtained from the bug tracking system on the website of Apache Software Foundation. In particular, the effort data of version 7 after the year 2011 are applied in order to understand the maintenance effort in Apache Tomcat.

Figure 2 is the main screen of the developed OSS project stability support tool. Moreover, Figures 3 and 4 show the estimated cumulative maintenance effort up to time  $t$  in cases of  $\Psi_e(t)$ ,  $E[\Psi_e(t)]$ ,  $\Psi_s(t)$ , and  $E[\Psi_s(t)]$ . From Figures 3 and 4, we found that the estimated maintenance effort required at the end of operation in cases of  $\Psi_e(t)$  and  $E[\Psi_e(t)]$  are negatively estimated than the case of  $\Psi_s(t)$  and  $E[\Psi_s(t)]$ . Similarly, Figures 5 and 6 show the estimated maintenance effort required until the end of operation in cases of  $\Psi_{re}(t)$ ,  $E[\Psi_{re}(t)]$ ,  $\Psi_{rs}(t)$ , and  $E[\Psi_{rs}(t)]$ . In particular, the noise in case of S-shaped type model becomes small according to the operation procedures go on. We found that the OSS project condition is stable after 2000 days.

Furthermore, we show several numerical examples based on the optimal maintenance problems which are discussed in Section 4. Figures 7 and 8 show the estimated total expected software maintenance effort in cases  $CPI_{CPI_e}$  and  $CPI_s$  as requirement specification, respectively. For example, we find that the optimum maintenance time with stability requirement is derived as  $t^* = 11.2$  years in Figure 8. Then, we found that the estimated CPI is larger than 1.0 at the time of  $t^* = 11.2$  years. These results show that this OSS has need to make a major-upgrade after 11.2 years from release. Thus, our tool can support the stability and effort control considering the operational environment of OSS. The developed tool will be helpful as the assessment tool of the progress of the OSS project in operation phase.

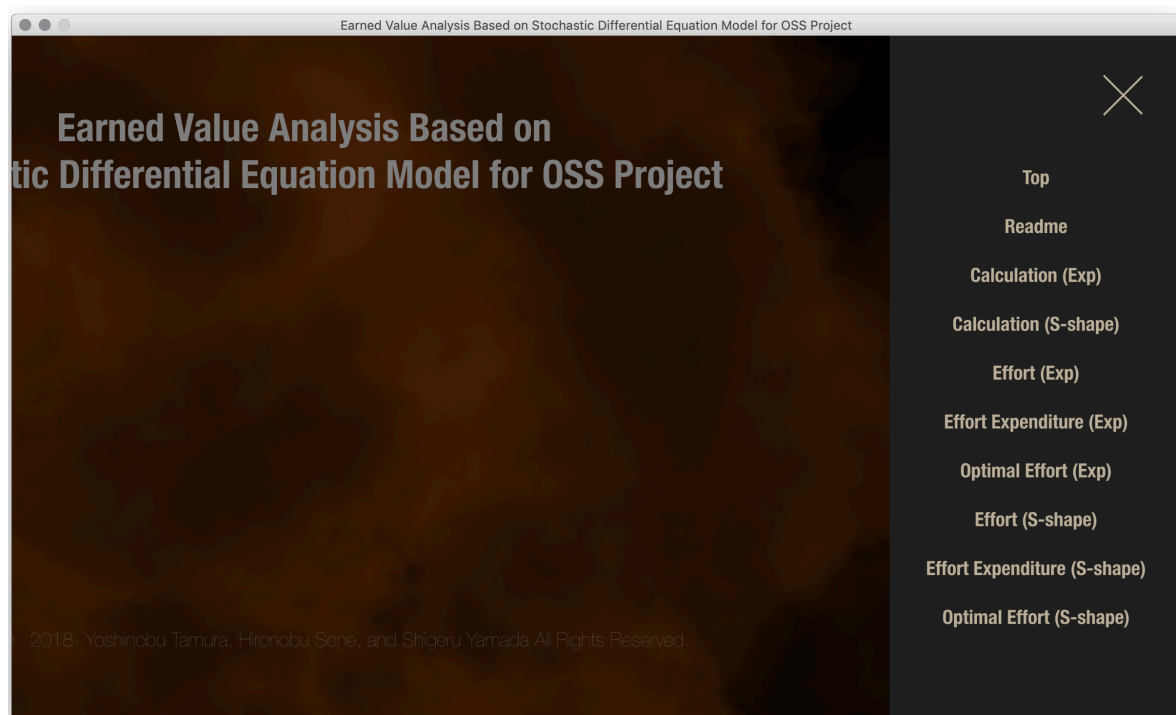
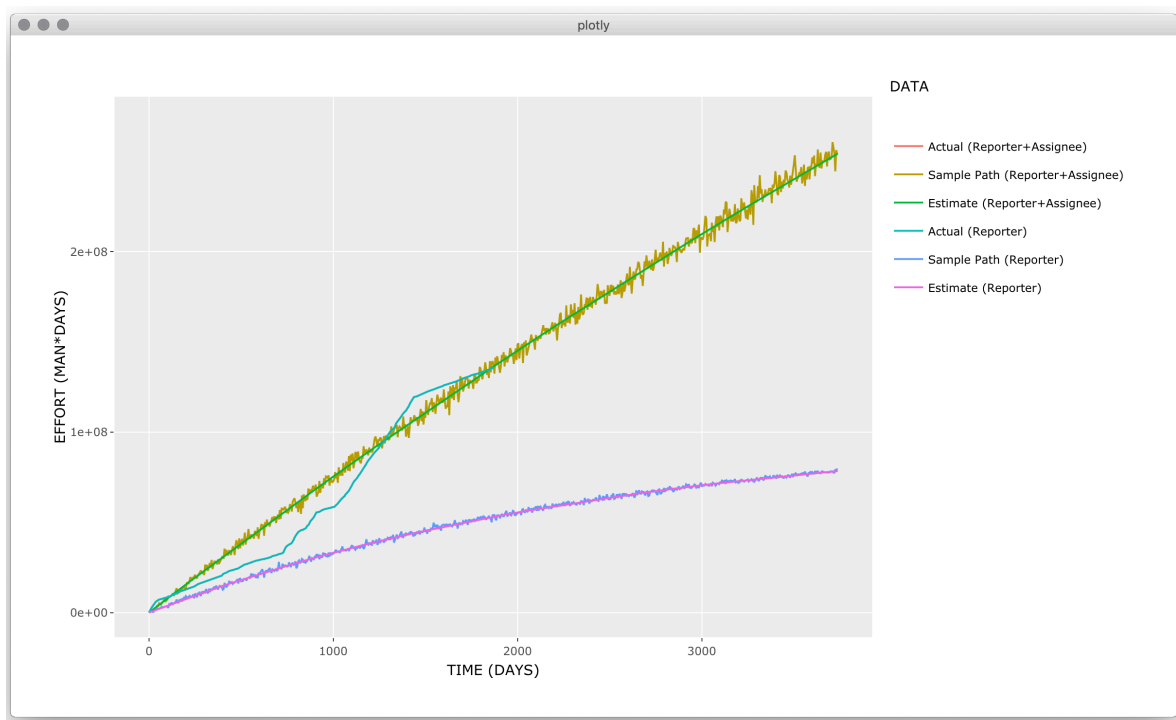
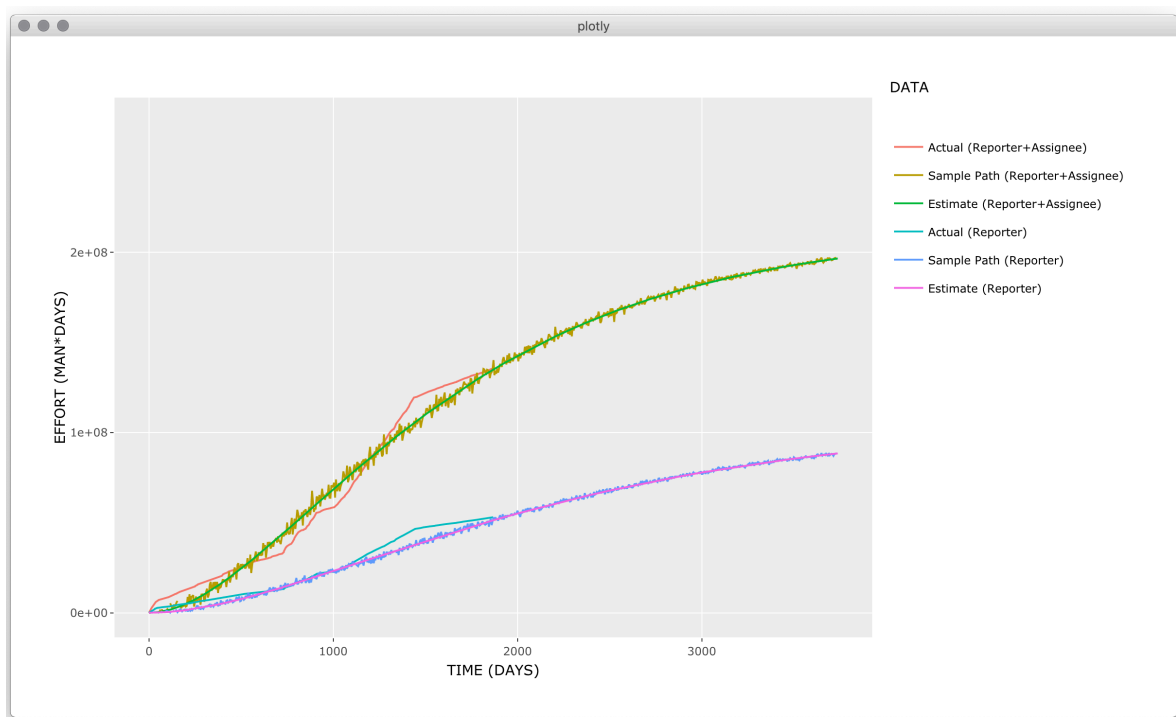


Figure 2. The main screen of the developed tool.

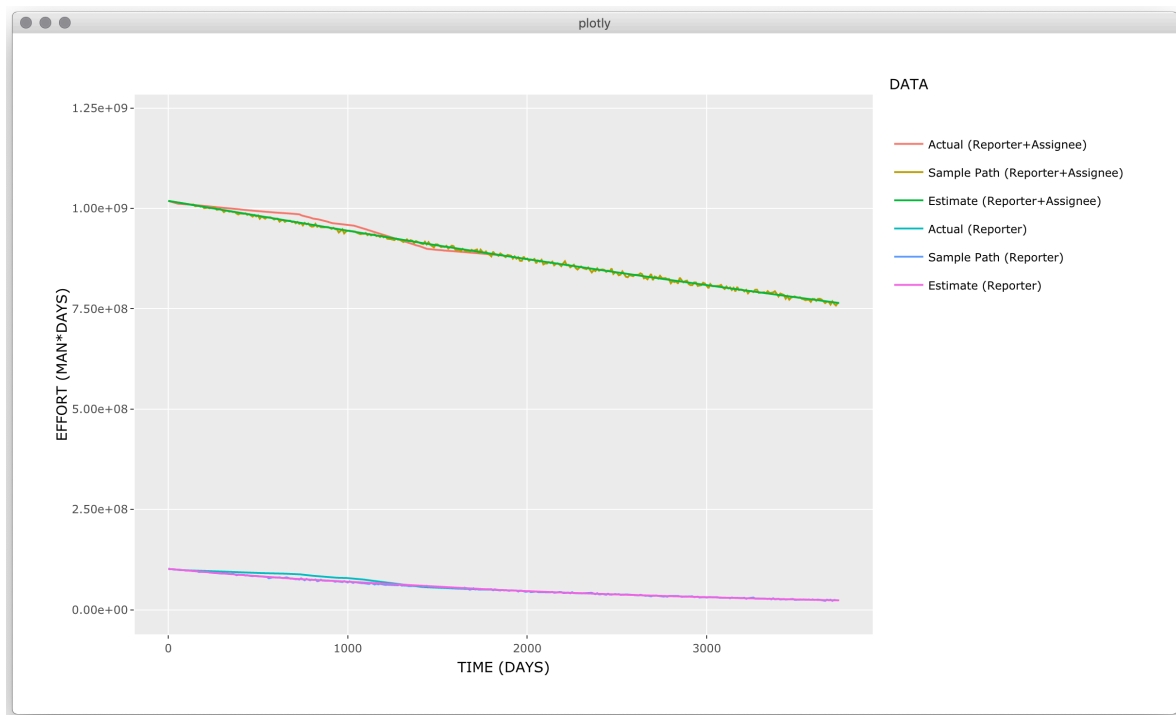


**Figure 3.** The executed cumulative maintenance effort up to time  $t$  in case of  $\Psi_e(t)$  and  $E[\Psi_e(t)]$ .

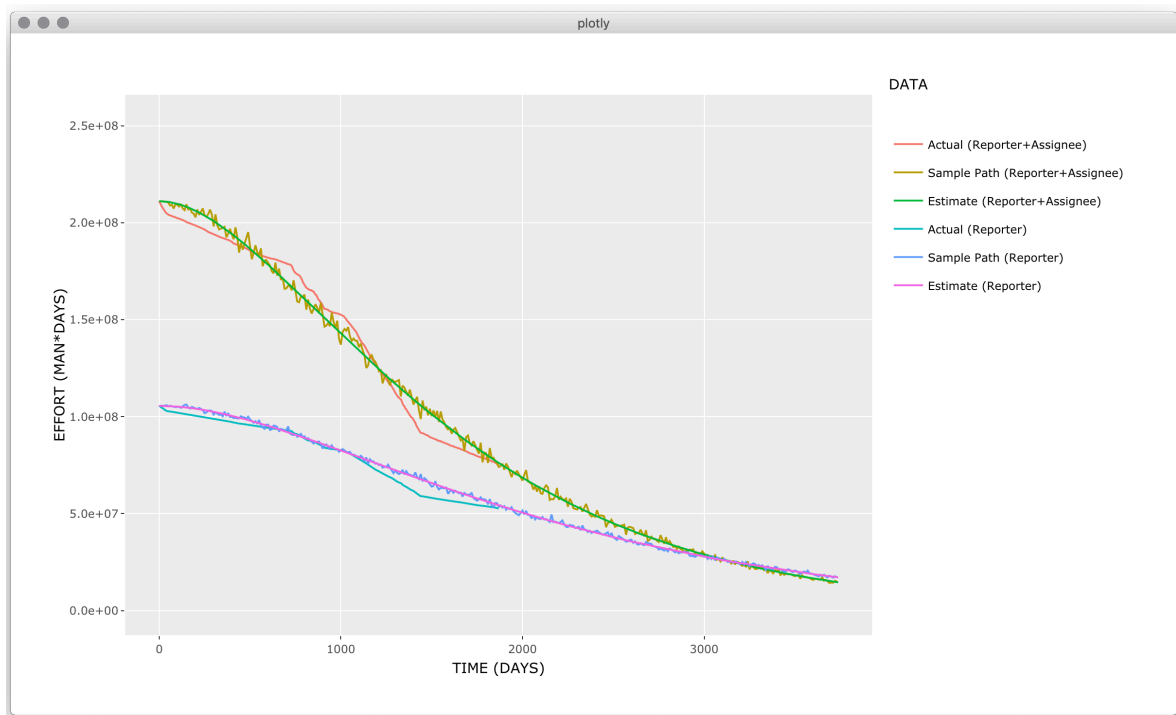


**Figure 4.** The executed cumulative maintenance effort up to time  $t$  in case of  $\Psi_s(t)$  and  $E[\Psi_s(t)]$ .

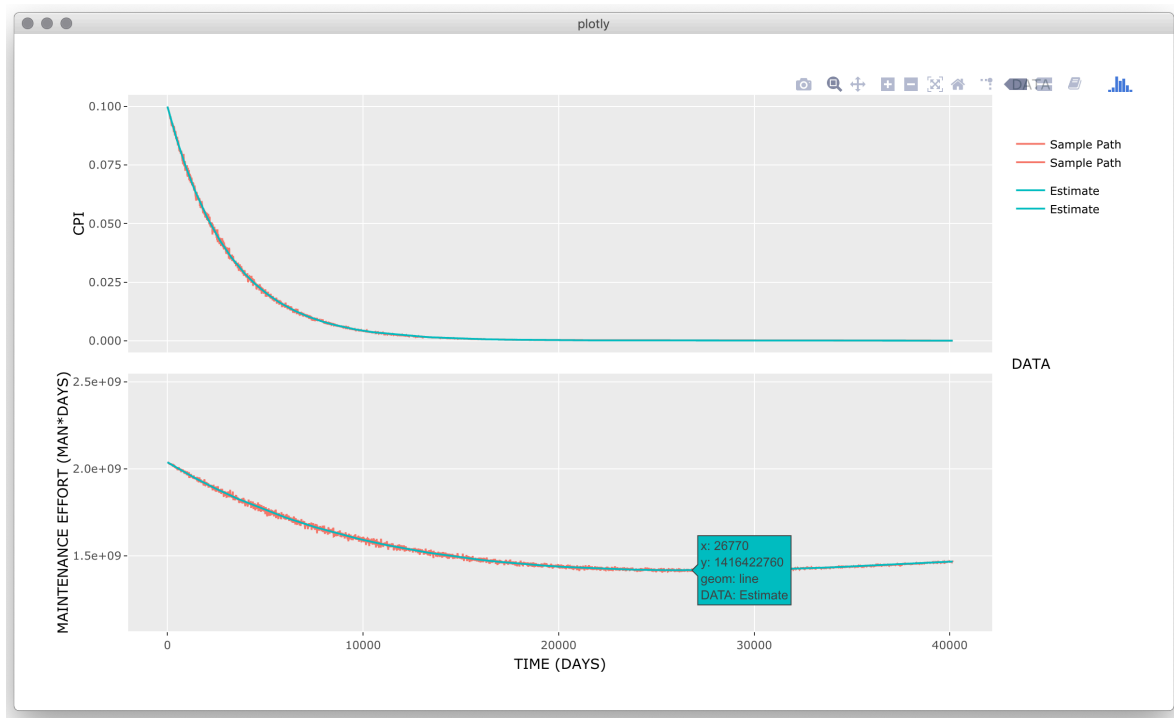




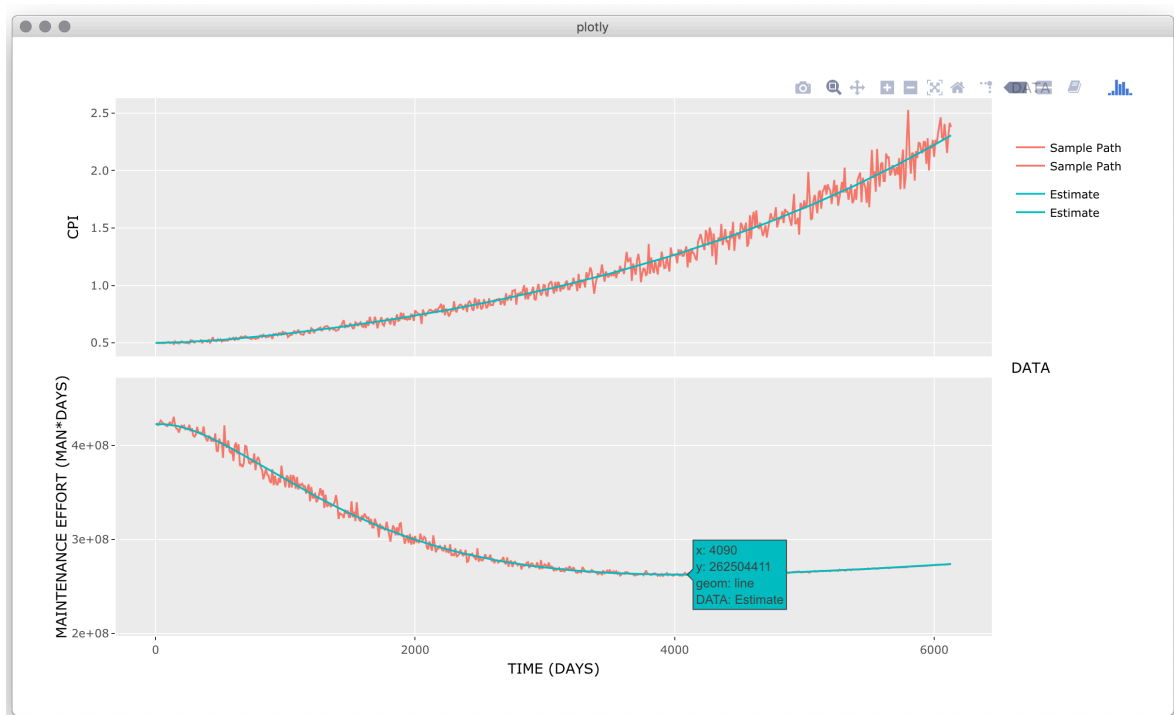
**Figure 5.** The executed fixing effort required for OSS maintenance until the end of operation time  $t$  in case of  $\Psi_{re}(t)$  and  $E[\Psi_{re}(t)]$ .



**Figure 6.** The executed fixing effort required for OSS maintenance until the end of operation time  $t$  in case of  $\Psi_{rs}(t)$  and  $E[\Psi_{rs}(t)]$ .



**Figure 7.** The executed total expected software maintenance effort considering  $CPI$ ,  $CPI_e(t)$  and  $E[CPI_e(t)]$  as requirement specification.



**Figure 8.** The executed total expected software maintenance effort considering  $CPI$ ,  $CPI_s(t)$  and  $E[CPI_s(t)]$  as requirement specification.

## 7. Conclusions

In the past, several conventional methods of OSS reliability assessment have been proposed. Usually, the fault data sets are used to assess the software reliability in the testing phase. However, the software fault is the effect generated by many fault-prone causes. In other words, the effort data

sets become a cause of fault data sets. Therefore, it is difficult for the conventional methods to totally assess the development phase by using the effort data, because the conventional methods focus on the reliability assessment based on fault data. We can control the OSS project considering the project stability and reliability by using the software effort data sets. Thereby, the proposed method will lead to assess the safety of OSS systems developed under various OSS projects. Also, the appropriate control of management effort for OSS will indirectly link to the quality, safety, reliability, and cost reduction of OSS.

In this paper, we have discussed the method of earned value analysis based on the stochastic differential equation model. Moreover, the optimal maintenance problem based on maintenance effort for OSS have been discussed. In particular, we have defined the optimal maintenance problems considering the CPI as the stability and requirement specification. Also, the software tool based on the optimal maintenance problem considering the OSS project stability in order to decide the optimal maintenance time has been developed in this paper. Furthermore, a set of actual OSS effort data has been analyzed to show several performance illustrations of the software tool in terms of the progress analysis and optimum maintenance time considering stability for an OSS project. Then, we found that our method and tool can assess the stability and effort control considering the operational environment of OSS. The developed tool will be helpful as the assessment tool of the progress of the OSS project in operation phase.

**Author Contributions:** Conceptualization, Y.T.; Data curation, H.S.; Formal analysis, Y.T.; Methodology, Y.T.; Project administration, Y.T.; Software, Y.T. and H.S.; Validation, Y.T.; Writing—review & editing, S.Y.

**Funding:** This research received no external funding.

**Acknowledgments:** This work was supported in part by the JSPS KAKENHI Grant No. 16K01242 in Japan.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Fleming, Q.E.; Koppelman, J.M. *Earned Value Project Management*, 4th ed.; PMI: Newton Square, PA, USA, 2010.
2. Yamada, S.; Tamura, Y. *OSS Reliability Measurement and Assessment*; Springer International Publishing: Cham, Switzerland, 2016.
3. Norris, J. Mission-critical development with open source software. *IEEE Softw. Mag.* **2004**, *21*, 42–49. [[CrossRef](#)]
4. Zhou, Y.; Davis, J. OSS reliability model: An empirical approach. In Proceedings of the Fifth Workshop on OSS Engineering, St. Louis, MO, USA, 17 May 2005; pp. 67–72.
5. Yamada, S. *Software Reliability Modeling: Fundamentals and Applications*; Springer: Tokyo, Japan; Heidelberg, Germany, 2014.
6. Lyu, M.R. (Ed.) *Handbook of Software Reliability Engineering*; IEEE Computer Society Press: Los Alamitos, CA, USA, 1996.
7. Musa, J.D.; Iannino, A.; Okumoto, K. *Software Reliability: Measurement, Prediction, Application*; McGraw-Hill: New York, NY, USA, 1987.
8. Kapur, P.K.; Pham, H.; Gupta, A.; Jha, P.C. *Software Reliability Assessment with OR Applications*; Springer: London, UK, 2011.
9. Li, X.; Li, Y.F.; Xie, M.; Ng, S.H. Reliability analysis and optimal version-updating for open source software. *J. Inf. Softw. Technol.* **2011**, *53*, 929–936. [[CrossRef](#)]
10. Ullah, N.; Morisio, M.; Vetro, A. A comparative analysis of software reliability growth models using defects data of closed and open source software. In Proceedings of the 35th IEEE Software Engineering Workshop, Crete, Greece, 12–13 October 2012; pp. 187–192.
11. Cotroneo, D.; Grottke, M.; Natella, R.; Pietrantuono, R.; Trivedi, K.S. Fault triggers in open-source software: An experience report. In Proceedings of the 24th IEEE International Symposium on Software Reliability Engineering, Pasadena, CA, USA, 4–7 November 2013; pp. 178–187.

12. Park, J.; Yu, H.C.; Lee, E.Y. Resource allocation techniques based on availability and movement reliability for mobile cloud computing. In *Distributed Computing and Internet Technology; Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2012; Volume 7154, pp. 263–264.
13. Suo, H.; Liu, Z.; Wan, J.; Zhou, K. Security and privacy in mobile cloud computing. In Proceedings of the 9th International Wireless Communications and Mobile Computing Conference, Cagliari, Italy, 1–5 July 2013; pp. 655–659.
14. Khalifa, A.; Eltoweissy, M. Collaborative autonomic resource management system for mobile cloud computing. In Proceedings of the Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, Valencia, Spain, 27 May–1 June 2013; pp. 115–121.
15. Gabner, R.; Schwefel, H.P.; Hummel, K.A.; Haring, G. Optimal model-based policies for component migration of mobile cloud services. In Proceedings of the 10th IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, 25–27 August 2011; pp. 195–202.
16. Park, N. Secure data access control scheme using type-based re-encryption in cloud environment. In *Semantic Methods for Knowledge Management and Communication; Studies in Computational Intelligence*; Springer: Berlin, Germany, 2011; Volume 381, pp. 319–327.
17. Wong, E. *Stochastic Processes in Information and Systems*; McGraw-Hill: New York, NY, USA, 1971.
18. Arnold, L. *Stochastic Differential Equations-Theory and Applications*; John Wiley & Sons: New York, NY, USA, 1971.
19. Yamada, S.; Kimura, M.; Tanaka, H.; Osaki, S. Software reliability measurement and assessment with stochastic differential equations. *IEICE Trans. Fundam.* **1994**, E77-A, 109–116.
20. Hanna, R.A. Earned Value Management Software Projects. In Proceedings of the 2009 Third IEEE International Conference on Space Mission Challenges for Information Technology, Washington, DC, USA, 19–23 July 2009; pp. 297–304.
21. Tamura, Y.; Sone, H.; Yamada, S. Earned value analysis and effort optimization based on Wiener process model for OSS project. In Proceedings of the 2018 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM 2018) and 2018 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE 2018), Qingdao, China, 21–24 August 2018; pp. 373–378.
22. Yamada, S.; Osaki, S. Cost-reliability optimal software release policies for software systems. *IEEE Trans. Reliab.* **1985**, R-34, 422–424. [[CrossRef](#)]
23. Yamada, S.; Osaki, S. Optimal software release policies with simultaneous cost and reliability requirements. *Eur. J. Oper. Res* **1987**, 31, 46–51. [[CrossRef](#)]
24. The Apache Software Foundation. The Apache HTTP Server Project. Available online: <http://httpd.apache.org/> (accessed on 1 January 2019).

