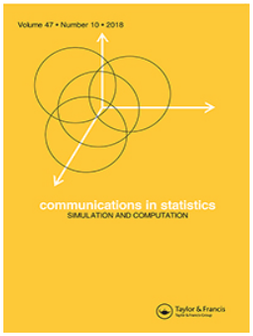

Taylor & Francis
Taylor & Francis Group

Communications in Statistics—Simulation and Computation®

ISSN: 0361-0918 (Print) 1532-4141 (Online) Journal homepage: https://www.tandfonline.com/loi/lssp20

Determining Process Death Based on Censored Activity Data

Nicholas Evangelopoulos , Anna Sidorova , Stergios Fotopoulos & Indushobha Chengalur-Smith

To cite this article: Nicholas Evangelopoulos , Anna Sidorova , Stergios Fotopoulos & Indushobha Chengalur-Smith (2008) Determining Process Death Based on Censored Activity Data, Communications in Statistics—Simulation and Computation®, 37:8, 1647-1662, DOI: 10.1080/03610910802140224

To link to this article: https://doi.org/10.1080/03610910802140224

Published online: 10 Oct 2008.

Communications in Statistics—Simulation and Computation®, 37: 1647–1662, 2009

ISSN: 0361-0918 print/1532-4141 online
DOI: 10.1080/03610910802140224

Taylor & Francis
Taylor & Francis Group

Survival Analysis

Determining Process Death Based on Censored Activity Data

NICHOLAS EVANGELOPOULOS[1], ANNA SIDOROVA[1], STERGIOS FOTOPOULOS[2], AND INDUSHOBHA CHENGALUR-SMITH[3]

[1]Department of Information Technology and Decision Sciences, University of North Texas, Denton, Texas, USA
[2]Department of Management and Operations, Washington State University, Pullman, Washington, USA
[3]Information Technology Management, School of Business, University of Albany-SUNY, Albany, New York, USA

This article addresses the problem of estimating the time of apparent death in a binary stochastic process. We show that, when only censored data are available, a fitted logistic regression model may estimate the time of death incorrectly. We improve this estimation by utilizing discrete-event simulation to produce simulated complete time series data. The proposed methodology may be applied to situations where time of death cannot be formally determined and has to be estimated based on prolonged inactivity. As an illustration, we use observed monthly activity patterns from 300 real Open Source Software development projects sampled from Sourceforge.net.

Keywords Discrete event simulations; Logistic regression; Open source software; Process activity; Survival analysis.

Mathematics Subject Classification 62N02.

1. Introduction

Survival analysis typically starts with the premise that a definite determination of whether the subjects are dead or alive is possible. A population of subjects is examined, and the ratio of dying subjects per time period, known as hazard rate, is recorded. A model can be then built to describe the hazard rate as a function of time. Such models include the Cox proportional hazards model (Cox, 1975), as well as various logistic regression modeling approaches (Efron, 1988; Silveira Chalita et al., 2006).

Received February 16, 2007; Accepted April 16, 2008
Address correspondence to Nicholas Evangelopoulos, ITDS Department, University of North Texas, P.O. Box 305249, Denton, TX 76203-5249, USA; E-mail: evangeln@unt.edu

In the context of organizational survival, however, establishing the death of any social entity can be problematic due to potential organizational transformations, mergers, etc. The late 1990's and early 2000's witnessed the boom of Open-Source Software (OSS) development coupled with the unprecedented success of several open-source software products such as the Linux operating system, Mozilla Web browser, and Apache server. This spurred significant interest in the open-source movement on the part of industry practitioners (Kevin, 2004; Lacy, 2006) as well as academic researchers (c.f. Bergquist and Ljungberg, 2001; Glass, 2003; Huntley, 2003).

Questions of survival and sustainability of open-source projects have received growing attention (Chengalur-Smith and Sidorova, 2003; Crowston et al., 2004; Norris, 2004; Paulson et al., 2004). In most cases, such research attempts to examine the influence of a variety of factors on project survival. However, in order to conduct successful empirical studies in the area, one should be able to define and measure the survival of open-source projects, or, in other words, to be able to identify the death of an open-source project in an accurate and timely manner. Since OSS projects, like many other virtual communities, are relatively free of formal obligations, the true time of death of an OSS project may be unclear to an external observer. Therefore, survival researchers are left with no option but to infer the death of open source projects from project activity patterns.

The need to associate a prolonged inactivity period with the death of the activity-generating process emerges in a variety of contexts. In health sciences, patients experiencing absence of activity ("remission") of a particular disease, such as cancer, are routinely monitored each month for signs of come-back ("relapse") of the disease (e.g., Craddock et al., 2000). A large number of electronic devices, ranging from computer screens and optical disk drives to household water supplies, employ power-saving features that are invoked after a certain period of inactivity. Dormant bank accounts have been known to be criminal targets by greedy bank employees who think nobody is going to notice the missing funds (Howerton, 1985). In 1995, the Swiss Banking Association (SBA) announced the discovery of $34 million in dormant bank accounts that may have belonged to World War II Holocaust victims (Salazar, 2000). Unlike Switzerland, where bank accounts are allowed to be inactive indefinitely, most states in the U.S. require by law that inactive account balances be turned over ("escheat") to the state after a period ranging from two to five years (depending on the state). Unclaimed property audits are one of the hottest areas in state and local tax practice centers (Fiore, 1999). But how is an account declared dormant? The determination of the associated inactivity period threshold is usually done in an ad-hoc fashion.

The purpose of this article is to develop and demonstrate a methodology that would allow establishing the death of a process based on lack of process activity, while controlling for an error related to potential activity resurgence. While the illustration of this methodological approach revolves around OSS project activity patterns as observed from SourceForge.net, the methodology may be applied to other similar activity patterns. The article is organized as follows. Section 2 presents the problem of estimating the time of death of a process using censored activity data. Section 3 characterizes the problem from a probabilistic point-of-view and proposes an algorithm that combines logistic regression and discrete event simulation in order to estimate process death when only censored data are available. Section 4 illustrates the proposed methodology using simulated data. In Sec. 5, we apply our methodology to real open-source project activity data. Finally, in Sec. 6, some concluding comments are given.

2. Process Activity and Process Death

A typical alive activity-generating physical process, such as an open-source project, exhibits periods of inactivity that may be associated with actual inactivity of the process or the inability to capture process activity using a particular data collection method. Thus, the actual observed activity pattern may depend on the type of physical process, definition of process activity, process activity data source, etc. Figure 1(a) presents complete activity data of a hypothetical process for illustration purposes, with "1" identifying activity, and "0" identifying inactivity. The term "complete data" here implies that the time of process death is known and the observation period extends beyond the time of death of the process. When such complete data are available, periods of observed activity and temporary inactivity will be followed by a period of permanent inactivity, and the duration of this observed permanent inactivity will depend on how long the process is observed after its death. Figure 1(a) shows an activity pattern for a hypothetical process that has been observed for nine periods (e.g., months) after its death. Permanent inactivity is indicated by grey color.

In this article, we examine the situation where the time of death of a process is unknown and the observation period may end before or after the death of the process, leading to potentially incomplete activity data. We will refer to this type of incomplete data as *censored process activity data*. Figure 1(b) shows activity patterns for three hypothetical processes which died at different times. Death of each process is indicated by the beginning of a permanent inactivity period (grey color). Process 1 died before the end of the observation period, whereas processes 2 and 3 died after the end of the observation period. Therefore, at time t_o the observer is presented with the task of classifying the three observed censored inactivity periods as permanent or temporary based on their duration and some chosen inactivity threshold. In the next section, we examine a probabilistic framework that characterizes our problem.

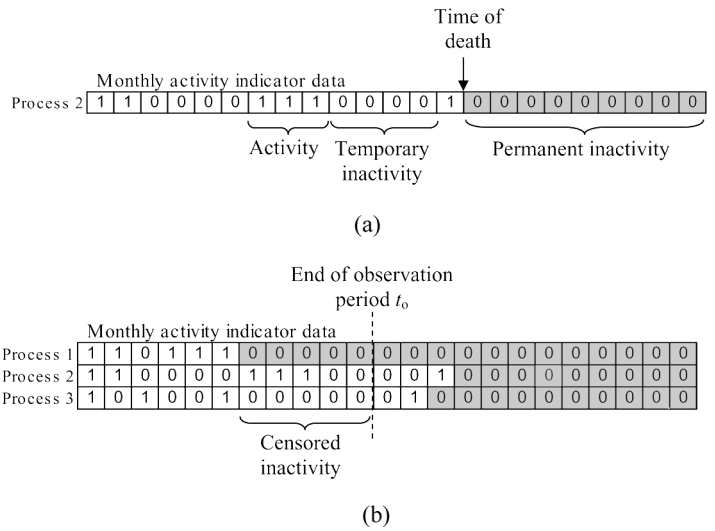

Figure 1. Illustration of process activity patterns.

3. Estimation of Probability for Resurgence Using Complete and Censored Data

3.1. *Process Activity and Future Resurgence*

In order to study activity and inactivity patterns of a certain, observable physical process with a time horizon $k \in \mathbf{N}$ representing a predetermined number of periods of interest, we start by defining the discrete random sequence $\{X_i : 0 \leq i \leq k, i, k \in \mathbf{N}\}$ as:

$$X_i = \begin{cases} 1 & \text{if the physical process is active in period } i, \\ 0 & \text{if the physical process is inactive in period } i. \end{cases}$$

For simplicity, we will assume that the sequence $\{X_i\}$ consists of identical Bernoulli trials, with a constant activity probability $P(X_i = 1) = p$ and a constant inactivity probability $P(X_i = 0) = q$. Consider the random walk $\{S_n : 0 \leq n \leq k, n, k \in \mathbf{N}\}$, defined as $S_n = X_0 + \cdots + X_n$ with $S_0 = 1$, representing the number of active periods up to time period n, starting with an active period. Since the random walk is time and space homogeneous, $p_1(x) = P(S_x = 0) = q^x$, $x \in \mathbf{N}$, is the same as $p_1(x) = P(S_x = 1 \mid S_0 = 1)$, $x \in \mathbf{N}$, and represents the probability of observing a run of x inactive periods. Without loss of generality, we will assume from now on that we start with an inactive period and thus $p_2(x) = P(S_x \geq 1) = 1 - q^x$ represents the probability of observing at least one active period among x periods.

3.2. *Process Survival*

In our discussion so far we have assumed that the underlying physical process continues to have the potential of producing activity, i.e., it remains "alive" for all time periods of interest. We will now consider the case where the process can die. A dead physical process can no longer produce activity, therefore remains permanently inactive. In order to study process survival, let us consider the discrete random sequence $\{Q_j : 0 \leq j \leq k, j, k \in \mathbf{N}\}$, representing the survival state, defined as:

$$Q_j = \begin{cases} 1 & \text{if the physical process is alive in period } j, \\ 0 & \text{if the physical process is dead in period } j. \end{cases}$$

For simplicity, we assume that $\{Q_j\}$ is generated by a Markov chain with transition probabilities as shown in the state transition diagram presented in Fig. 2. Note that $Q_j = 0$ is an absorbing state. Therefore, reaching this state will result in a long run of inactivity: for any $1 \leq j \leq x$, $P(S_x - S_{j-1} \geq 1 \mid Q_j = 0 \cap Q_0 = 1) = 0$, meaning

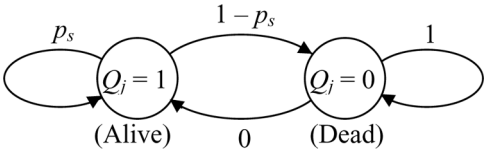

Figure 2. State transition diagram for the Markov chain $\{Q_j\}$.

that activity can no longer be produced after death. Also note that the events $\{Q_j = 0\}$ and $\{X_j = 0\}$ are dependent, since a dead process can no longer produce activity. To avoid trivial degenerate situations, set $Q_0 = 1$, i.e., begin the sequence $\{Q_j\}$ from an alive initial state. Then, for $x > 0$, $p_3(x) = P(Q_x = 1 \mid Q_0 = 1) = p_s^x$ represents the probability of observing a run of x surviving periods in our underlying physical process. Combining our Markov chain and random walk notations, for $x > 0$, $p_4(x) = P(Q_x = 1 \cap S_x = 0 \mid Q_0 = 1) = p_s^x q^x$ represents the probability of observing a run of x temporarily inactive periods, i.e., periods that even though they are all inactive, they maintain their capability to produce some activity in the future. Lemma 3.1 below calculates the probability to observe a run of inactive periods by considering the cases where the physical process is dead therefore inactive, and those where the physical process is alive and inactive.

Lemma 3.1. *Let q and p_s be the constant probabilities of inactivity and survival, respectively, as previously defined. The probability to observe a run of x inactive periods, given the process is alive at the start, is then given by*:

$$p_5(x) = \frac{(1 - p_s)(1 - p_s^x q^x)}{1 - p_s q} + p_s^x q^x. \tag{3.1}$$

Proof. In each period, in order to get inactivity, we can either have a process that has died in some previous period, or a process that survives but remains inactive. Considering all such periods, $p_5(x) = P(Q_1 = 0 \mid Q_0 = 1) + \sum_{1 \le j < x} P(Q_j = 1 \cap S_j = 0 \cap Q_{j+1} = 0 \mid Q_0 = 1) + P(Q_x = 1 \cap S_x = 0 \mid Q_0 = 1) = (1 - p_s) + p_s q(1 - p_s) + p_s^2 q^2(1 - p_s) + \cdots + p_s^x q^x$, and (3.1) follows after we observe that the last expression is a geometric series with x terms plus $p_s^x q^x$.

For active periods to occur, all the preceding periods must now also be a run of surviving periods. Therefore, let us define future survival and activity resurgence as $FSR_k = \{$following a run of inactive periods, the process survives and produces activity in at least one period by the time the kth period is reached$\}$. Lemma 3.2 below examines the probability to observe a run of inactive periods followed by future survival and activity resurgence. Note that a run of x inactive periods can occur in many ways: we can have a death in period 1, or survival with inactivity in period 1 and then death in period 2, etc. However, from all those cases, only survival with inactivity until period x can allow some future activity.

Lemma 3.2. *Let q and p_s be the constant probabilities of inactivity and survival, respectively, as previously defined. The probability to observe a run of x inactive periods (i.e., $S_1 = S_2 = \cdots = S_x = 0$) followed by future survival and activity resurgence within k periods is then given by*:

$$\begin{aligned} p(x \cap FSR_k) &= P(Q_x = 1 \cap S_x = 0 \mid Q_0 = 1) \sum_{x < j \le k} P(Q_j = 1 \cap S_j \ge 1 \mid Q_0 = 1) \\ &= p_s^{x+1} q^x (1 - q) \frac{1 - p_s^{k-x} q^{k-x}}{1 - p_s q}. \end{aligned} \tag{3.2}$$

Proof. $p(x \cap FSR_k) = p_s^{x+1} q^x (1 - q) + p_s^{x+2} q^{x+1} (1 - q) + \cdots + p_s^k q^{k-1} (1 - q)$. The proof of Lemma 3.2 now follows after observing that this is a geometric series with $k - x$ terms.

Let us now examine the probability to observe a future resurgence, given an observed run of inactive periods. This is given in Lemma 3.3 below, which is presented as a lemma without proof, since the proof follows easily from combining Lemmas 3.1 and 3.2.

Lemma 3.3. *Let q and p_s be the constant probabilities of inactivity and survival, respectively, as previously defined. The conditional probability to observe a future survival and activity resurgence within the remaining $k - x$ periods, given an observed run of x inactive periods (i.e., given $S_x = 0$), is then given by*:

$$p(FSR_k \mid x) = p(x \cap FSR_k)/p_5(x) = \frac{p_s^{x+1} q^x (1-q)(1 - p_s^{k-x} q^{k-x})}{1 - p_s + p_s^{x+1} q^x (1-q)}. \tag{3.3}$$

After the end of the first inactivity run and after some subsequent activity, a new inactivity run may begin. In this situation the time horizon gets reduced to k^*, $1 \leq k* < k$, i.e., fewer than k time periods of interest remain to be observed. Within the new inactivity run, expression (3.3) still holds, but k is now modified to k^*, and $p(FSR_{k*} \mid x) < p(FSR_k \mid x)$ as one can easily show.

3.3. *Estimation of the Probability of Resurgence*

Using observed process activity data, one could estimate $p(FSR_k \mid x)$ by fitting a logistic regression model for the log-odds ratio of FSR_k, regressed on a polynomial function of the sub-run size x:

$$p(FSR_k \mid x) = \frac{\exp(\beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p)}{1 + \exp(\beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p)}, \quad 1 \leq x < k, \;\; x \in N. \tag{3.4}$$

In order to obtain the fitted model (3.4), one would have to consider all sub-runs of size x that would be followed by resurgence at any future point, and not just the total inactivity run size that is followed by resurgence in the immediately following period.

When we have censored data, i.e., the physical process is observed for a shorter time period k^*, $1 \leq k^* < k$, such censoring will have two effects: it will (i) decrease the observed future survival and resurgence, leading to an underestimation of $p(x \cap FSR_k)$ since the number of periods that follow the inactivity run of size x is now smaller; it will also (ii) set the probability to observe inactivity runs of size $x > k^*$ to zero, since there will not be enough time to observe such long inactivity runs, leading to an underestimation of $p_5(x)$. As a result, depending on p_s, q, k, and x, $p(FSR_k \mid x)$ could be underestimated or overestimated using the logistic regression model in (3.4). In the following section we propose improving these estimates following "sudden death" and simulation approaches.

3.4. *Estimation of Resurgence Probability Based on Simulated Data*

Based on our previous discussion, consider now a homogenous population of processes where some average $p(FSR_k \mid x)$ needs to be estimated based on censored data. A first "naïve" approach to artificially increasing the observed data through simulations would be to apply sudden death to all processes. The "sudden death"

approach assumes that all processes die at the end of the observation period. The censored data are then augmented with inactivity runs of size m, representing lack of activity, since the processes are all assumed to be dead. The "sudden death" approach will tend to underestimate $p(FSR_k \mid x)$, since some potential future resurgences will be eliminated. It may, however, improve the estimate for large inactivity sub-runs, for which the event of non resurgence may not have been observed frequently enough.

As a second approach, to further improve the estimation of $p(FSR_k \mid x)$, we propose an algorithm that generates simulated activity data and then uses them to estimate the resurgence probability. The algorithm includes the following steps:

1. Use the observed activity data to extract all the activity and inactivity runs. Obtain their empirical distribution functions using the activity/inactivity run size as a random variable.

2. Fit a logistic regression model to obtain the first probability for resurgence $\hat{p}_1(FSR_k \mid x)$ for each temporary inactivity sub-run size x. Use the most recent period as an indicator of final activity and mark as "dead" a process that is inactive during that period. Select a threshold for x based on an acceptably small resurgence probability.

3. Applying the selected death threshold to the censored data, for each process, estimate the time of death. If a Markov chain model such as the one presented in Sec. 3.2 is assumed, exploiting the properties of Geometric distribution, parameter p_s (the constant survival probability) can be estimated as the inverse of the average survival period. If p_s is assumed to be diminishing as the process ages, some linear function that approximates the number of deaths per time period may be estimated instead.

4. Generate process death events either based on p_s (Markov chain assumption) or based on an estimated process death rate. Generate random run size data based on the empirical distributions for activity and inactivity runs and store them in two queues. Start each process with a random choice of activity or inactivity run and continue filling up the process with alternating activity and inactivity runs taken from the queues until the time of death of the process is reached, then fill the remaining periods with inactivity. When a run size is too long to fit in the period before the death of that process, move to the next variate in the queue, but continue considering the queue entries from the top. Once a variate has been used, delete it from the queue. Extend the process simulations over a time period that is long enough to observe permanent inactivity periods longer than the longest observed temporary inactivity for all projects. Extend the process simulations over a time period that is long enough to observe permanent inactivity periods longer than the longest observed temporary inactivity for all projects.

5. Fit a logistic regression model on the simulated data. Obtain the second estimation of resurgence probability $\hat{p}_2(FSR_k \mid x)$ and select the second death threshold.

6. If the threshold in Step 5 is different from the one used in Step 2, repeat Steps 2–5 until the results converge and produce a final approximation of $p(FSR_k \mid x)$.

In the next section we demonstrate the use of this approach using simulated process activity data.

4. Simulation Study

To illustrate how process resurgence probability estimates can be improved with the use of the two approaches described in Sec. 3.4, we will use simulated process activity data. We will first obtain estimates using complete data and then try to approximate them using censored data (see Secs. 2 and 3 for our definitions of complete and censored data.)

Let us consider a population of $M = 500$ processes, each observed at $N = 100$ time periods (months). Let $\{X_{m,n}\}$ be a binary sequence of activity/inactivity for process m, with $n = 1, \ldots, N$. Assume each *alive* sequence is generated by a Bernoulli process with success probability p_m, different for each process m. In order to achieve a wide range of different values, let us generate 500 different Bernoulli parameters coming from a Normal distribution, $p_m \sim N(0.5, 0.2^2)$, and truncate any value that is outside the [0, 1] range. Regarding process death, assume that the 500 processes die according to the following schedule:

1. Months 1–20: all processes are alive;
2. Months 21–70: processes die at a constant rate of 2%;
3. Months 71–100: all processes are dead.

In order to make the simulation more realistic, we will not kill exactly 10 processes in each one of the 50 "dying" months, but we will consider a normally distributed death count $h_t \sim N(10, 3^2)$, with the extra condition $\sum_{t=21}^{70} h_t = 500$, so that all processes die by month 70.

Using the complete simulated data, we estimated the probability of resurgence using a third-order polynomial function of the run size as the predictor of the log-odds ratio. The fitted model, determined in a stepwise selection fashion, provided good fit: The Hosmer–Lemeshow GOF test had $p = 0.220$ and the concordance index was 95.1%. The corresponding estimated probabilities, as a function of inactivity sub-run size, are shown in Fig. 3 (thick line), Fig. 4 (thick line), and

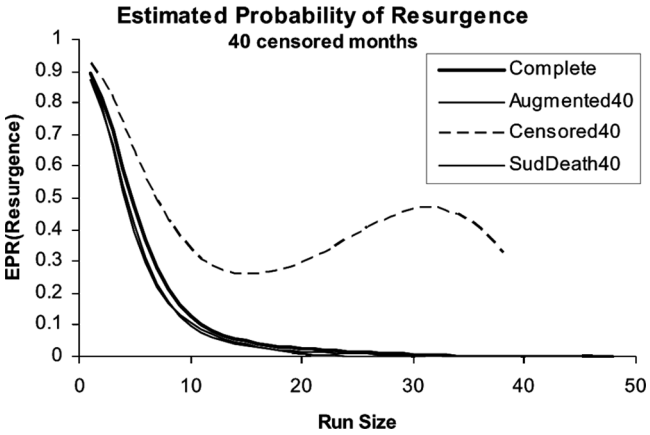

Figure 3. Probability of resurgence based on simulated data: 40-months retained.

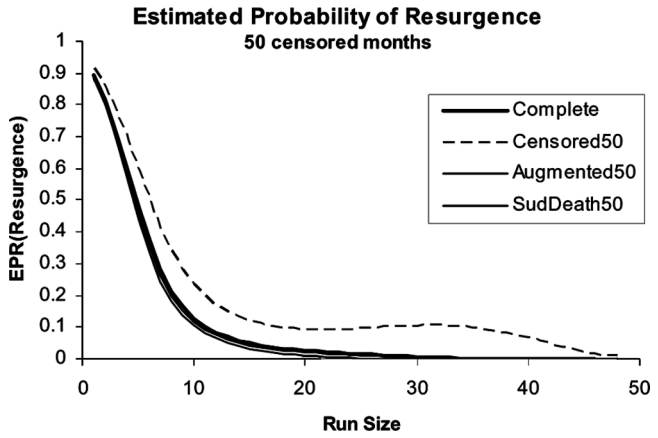

Figure 4. Probability of resurgence based on simulated data: 50-months retained.

Table 1 (column "Complete"). In order to compare our approaches to estimating the probability of resurgence when complete data are not available, we created two sets of censored data by keeping the first 40 and 50 months, respectively. Following the approach described in Sec. 3.3, we estimated $\hat{p}(FSR_k \mid x)$. Figure 3 (dashed line) and Table 1 (column "Censored 40") show the estimated probability of resurgence based on 40 months of censored data. Comparing it to the corresponding curve using complete data (Fig. 3, thick line, Table 1, column "Complete") we see that the discrepancy is significant and is increasing for large inactivity run sizes. Note that the hump shown at the right end of the dashed curve in Fig. 3 is, to some extent, an artifact of the third-order polynomial model. However, the heavy tail in this estimated curve reflects an existing upward bias on the proportion of resurging inactive runs that is a result of censoring: given some longer observation period, long non-resurging runs of inactivity associated with dead processes will have a chance to appear in the data and the proportion of resurging inactive runs, for large run sizes, will exhibit a downward correction. This is what is accomplished by our "sudden death" and "simulation" approaches. Similar results were obtained based

Table 1
Estimated probabilities of resurgence for 500 simulated processes

	Resurgence probability estimates						
Run Size	Complete	Censored 40	Sudden Death 40	Simulation 40	Censored 50	Sudden Death 50	Simulation 50
1	0.8926	0.9254	0.8710	0.8879	0.9132	0.8837	0.8921
5	0.4748	0.6484	0.4161	0.3930	0.5953	0.4537	0.4350
10	0.1293	0.3376	0.0979	0.1047	0.2332	0.1174	0.1046
15	0.0491	0.2581	0.0336	0.0396	0.1192	0.0424	0.0327
20	0.0263	0.2950	0.0170	0.0106	0.0931	0.0216	0.0102
25	0.0148	0.3878	0.0097	0.0008	0.0953	0.0117	0.0018
30	0.0062	0.4628	0.0046	0.0000	0.1039	0.0049	0.0001

on 50 months of censored data (Fig. 4, dashed line, Table 1, column "Censored 50"). We then followed our sudden death approach and estimated the probability of resurgence by fitting a similar logistic regression model (Fig. 3, one of the two thin lines, Table 1, column "SuddenDeath40"). As evident from Fig. 3 and Table 1, sudden death approach provides a significant improvement over using raw censored data. Similar results were obtained based on 50 months of censored data (Fig. 4, one of the thin lines, Table 1, column "SuddenDeath50").

We next followed our algorithm in Sec. 3.4 by first obtaining the empirical distributions of temporary inactivity runs sizes and activity run sizes, without making any parametric assumptions (Step 1). We then obtained resurgence data as illustrated in Sec. 3.1 and fit a logistic regression model (Step 2). With respect to the inactivity threshold, we should mention here that our general goal is to identify a threshold associated with an acceptably small probability. In the general case, the resurgence probability would be monotonically decreasing, but in this particular case it was not, therefore our chosen threshold was simply the point in the curve where the probability of resurgence becomes minimum. Following Step 3, we set the inactivity threshold to 16 months, where the minimum resurgence probability is attained (see Fig. 3, dashed line). Regarding death rate, for simplicity, we considered a constant death rate. In the 5-month period between the 20th and the 24th month, 55 processes (out to of 500) exhibited inactivity periods of 16 months or longer without resurgence, yielding a monthly death rate of 2.2%. We then used the empirical distributions of the inactivity and activity runs and the estimated death rate to simulate new activity data for 500 processes over 100 months (Step 4). Subsequently, we obtained new estimates of the resurgence probabilities (Step 5), shown on Fig. 3 (one of the two thin lines) and Table 1 ("Simulation 40"). As suggested by Fig. 3 and Table 1, while the probability estimates in the simulation approach present a drastic improvement over estimates based on raw censored data, the improvement over the sudden death approach, if any, is only marginal. Continuation of the algorithm (Step 6) failed to produce any noticeable improvement. Similar results were obtained using 50 months of censored data (see Fig. 4 and Table 1). The probability estimate using raw censored data (Fig. 4, dashed line, Table 1, "Censored 50"), was closer to the probability curve using complete data (Fig. 4, thick line, Table 1, "Complete") than the estimate obtained using 40 months of raw censored data. Yet, application of the simulation algorithm once again resulted in a significant improvement of probability estimates (Fig. 4, thin line, Table 1, "Simulation 50").

5. Application on Open Source Software Project Activity Data

5.1. *Data Description*

In this section we will demonstrate the use of the logistic regression approach to identify the inactivity threshold and corresponding probability of resurgence for 300 open source projects registered with SourceForge.net between 1999 and 2003. The projects were randomly selected from the 70% of the most active projects as of May 2003 based on the SourceForge activity rank score. In a follow-up data collection, activity data for these projects were collected between the month of initial project registration and February 2006.

For the purpose of this study we were interested only in developer activity, therefore a project was considered active during a particular month if there was any evidence of developer activity registered at SourceForge.net portal, including new releases, patches (submitted or posted), features (added) or requests for support (answered), and bugs (fixed). During the observation period of 76 months, the 300 sampled projects were found to have an average of 4.21 activity runs per project, with an average size of an activity run of 2.13 months. The sampled projects had an average of 3.31 temporary inactivity runs per project, an average size of temporary inactivity run of 5.45 months and an average size of censored inactivity of 27.43 months. Figure 5 shows the distribution of temporary inactivity (a) and activity (b) run sizes for all 300 projects. In order to separate projects into more homogeneous groups in terms of temporary inactivity size, affiliation with a mainstream foundry (such as PhP, PERL, Games, and JAVA) and catering to multiple audiences were used as the key discriminating factors. Four groups were formed. Group 4 (multiple audiences/mainstream foundry) had the smallest average temporary inactivity run size and the largest average activity run size. In the next sections we estimate resurgence probability for various inactivity run sizes following the approaches discussed in Sec. 3. In addition to testing our approach on the entire set of 300 OS projects, we also apply the same approach to the more homogenous set of 67 projects belonging to Group 4.

5.2. *Resurgence Probability Estimates Using Raw Censored Data*

Using censored data from the 300 OSS projects in our 76-month observation period, we fit a Logistic Regression model for Future Resurgence with three polynomial orders of *RunSize* as predictors. The data set included 12,721 rows corresponding to all sub-runs contained in our observed inactivity runs. The model exhibited very good fit (Hosmer–Lemeshow Goodness-of-Fit test having $p = 0.601$) and a concordance index of 78.3%. All predictors (i.e., all polynomial orders of *RunSize*) were found to be highly significant ($p < 0.001$). Figure 6 (dashed line) and Table 2 (column labeled "Censored300") show the estimated resurgence probabilities for different run sizes. Figure 7 (dashed line) and Table 2 (column "Censored67") show similar estimated resurgence probabilities for Group 4 projects.

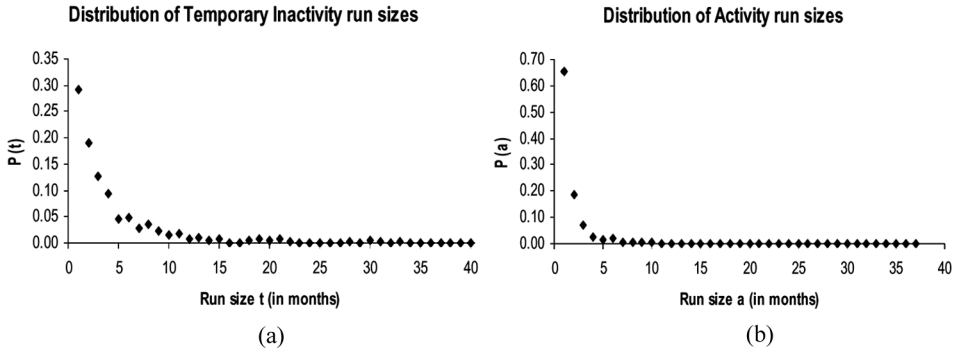

Figure 5. Distribution of temporary inactivity (a) and activity (b) run sizes for all projects.

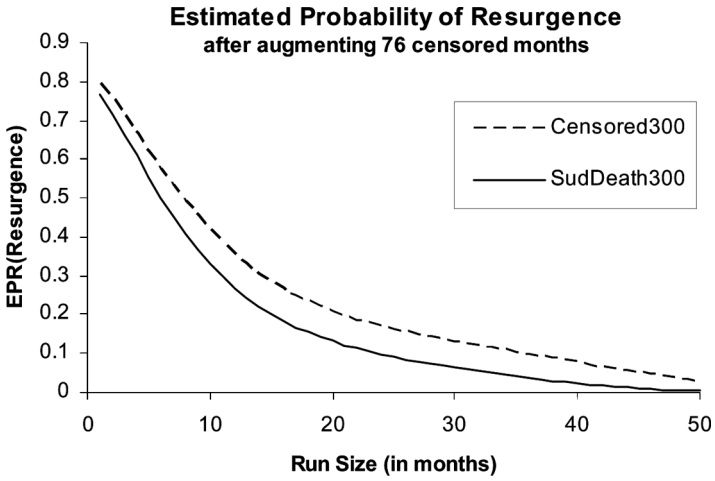

Figure 6. Expected probability of resurgence for all 300 projects.

5.3. *Resurgence Probability Estimates Using the Sudden Death Approach*

In order to improve the resurgence probability estimates using the sudden death approach, we augmented the censored data for the 300 sampled OSS projects with inactivity runs of 33 months, corresponding to the largest observed temporary inactivity. We then estimated the resurgence probability using logistic regression. The results are presented in Fig. 6 (thin continuous line) and Table 2 (column "SuddenDeath300"). Figure 7 (thin continuous line) and Table 2 (column "SuddenDeath67") show similar estimated resurgence probabilities for Group 4 projects.

Table 2
Estimated probabilities of resurgence for 300 sampled OSS projects and for the 67 projects belonging to Group 4 (multiple audiences/mainstream foundry)

	Resurgence probability estimates				
Run size	Censored 300	Sudden death 300	Censored 67	Sudden death 67	Simulated 67
1	0.7966	0.7677	0.8376	0.8015	0.8788
5	0.6210	0.5548	0.6274	0.5404	0.6398
10	0.4175	0.3301	0.3581	0.2630	0.3541
15	0.2838	0.2002	0.1979	0.1307	0.2028
20	0.2068	0.1312	0.1223	0.0762	0.1208
25	0.1611	0.0909	0.0879	0.0524	0.0627
30	0.1298	0.0629	0.0725	0.0407	0.0222
35	0.1030	0.0406	0.0665	0.0338	0.0041
40	0.0762	0.0226	0.0653	0.0282	0.0003

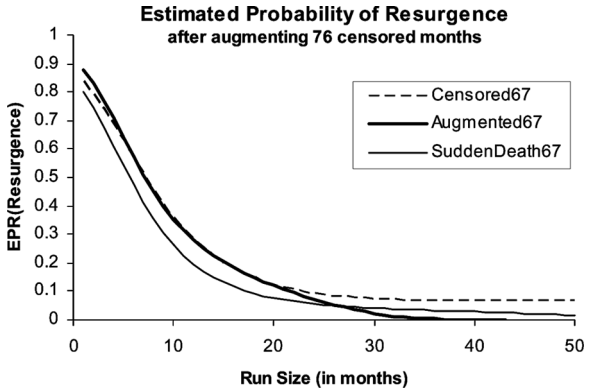

Figure 7. Expected probability of resurgence for Group 4 (67 projects).

5.4. *Applying the Simulation Algorithm*

Step 1. In the original sample of 300 OSS projects, 930 valid temporary inactivity runs and 1,186 activity runs were recorded. These runs followed the discrete probability distributions shown in Fig. 5. Out of these 300 projects, 67 belonged to Group 4 (multiple audiences/mainstream foundry). The discrete probability distributions for temporary inactivity and activity run sizes in Group 4 were similar in shape with smaller typical run sizes. Drawing from these empirical distributions, a large number (in excess of 10,000) of inactivity and activity run sizes was generated for each of the two types.

Step 2. We obtained resurgence probability estimates based on raw censored data, as described in Sec. 5.2 (see Fig. 6, dashed line, and Table 2, column "Censored300"). Using $\alpha = 0.10$ as an acceptably small probability, we selected a death threshold of 35 months when all 300 projects are considered, and a threshold of 22 months for Group 4 projects. As our observation period was not sufficiently long to produce a reliable death rate estimate based on a 35-month threshold, we will continue our analysis by focusing on Group 4. Figure 7 (dashed line) and Table 2 (column labeled "Censored67") show the estimated probability of resurgence for any given inactivity run size in Group 4.

Step 3. Based on the 22-month death threshold, projects belonging to Group 4 with censored inactivity sizes of 22 or smaller were considered likely to resurge and projects with censored inactivity sizes of 23 or larger were considered to have died at the beginning of such inactivity run. In our original sample of 67 projects belonging to Group 4, three exhibited no activity, therefore were considered dead from the beginning. Out of the remaining 64 projects, 21 projects were determined to have died within the 18-month period of May 2003 to October 2004 (the first 42 months were excluded from this calculation because during the period of November 1999 to April 2003, new projects were being added to the data set). The average monthly death rate is 1.82%. No trend was assumed among the data.

Step 4. Assuming a constant monthly project death rate of 1.82%, monthly activity data for 500 projects and 102 months was generated through discrete-event simulations. The choice of 102 months allowed for sufficient activity during the

first 15 months, before any projects started dying, and for enough time for all 500 projects to die according to their assumed death rate. It also allowed for 30 months after the death of the last project, in order to produce permanent inactivity sizes that outrun the observed temporary inactivity sizes in Group 4.

Step 5. Using simulated data, we fit a Logistic Regression model. The model exhibited acceptable fit (Hosmer–Lemeshow GOF test supported a good fit with a $p = 0.147$) and a concordance index of 90.3%. All predictors were found to be highly significant (all $p < 0.001$). Figure 7 (thick continuous line) and Table 2 (column "Simulated67") show the estimated probability of resurgence corresponding to different run sizes. Using $\alpha = 0.10$ as the cutoff probability, the death threshold is again 22 months, even though the shape of the probability function has changed. Since there was no revision in the project death threshold, there is no need for further iterative refinements.

Comparing resurgence probability estimates obtained by the logistic regression analyses in Steps 2 and 5, we see that probabilities using simulated data, for large run sizes are generally smaller. This is not surprising, considering the distribution of temporary inactivity sizes (larger temporary inactivity runs become less frequent), relatively high death rate, and the fact that censored inactivity is less accurate in approximating larger permanent inactivity runs. Generally, as we discussed in Sec. 3, differences between probabilities produced by the two estimation methods would depend on the censored observation period, as well as the process parameters.

6. Discussion and Conclusion

In this article we examined an approach to the death of open source projects based on censored activity data. The approach combines logistic regression analysis with discrete event simulation and provides an alternative to Markovian birth and death process analysis. Some limitations of our approach include its reliance on censored data to obtain simulation parameters: the results will therefore be subject to the same inaccuracies and biases as the original data. Another limitation relates to our independence assumption. Sub-runs from the same project were treated as independent observations, when in fact they should be correlated. In our particular case, examination of an ACF plot of the run sizes after accounting for project, revealed some weak within-project autocorrelation. In order to make our simulation study (Sec. 4) more realistic, we introduced Bernoulli parameters for project activity p_i that were the same within the projects, but varied slightly across projects. Still, our approach proved robust enough, to some extent, to still able to produce some improved results. An example that illustrates another limitation of our approach is the case where the entire group of 300 projects was considered. The estimated death threshold of 35 months was too long for our observation to produce a reliable corresponding death rate estimate. In this case, a longer observation period was needed.

Because the proposed methodology is data-oriented and does not have to follow the assumptions of discrete time Markov chains, it can be extended to accommodate such factors as a non constant death rate of projects, existence of different groups of projects that exhibit significantly different activity patterns and may need to be evaluated using different thresholds, and change in distributions of activity and temporary inactivity sizes over time. Further directions of research may include a

more detailed examination of how the censoring time, the death rate and activity and inactivity distributions affect the accuracy of results obtained by fitting the logistic regression using censored data.

The problem of identifying potentially dead processes is not unique to the OSS field. The Internet suffers from a similar problem, sometimes referred to as "link rot", where the web resources either disappear or are relocated over time. Some servers have established policies for the timely removal of "cobweb sites", web pages that have no active webmasters or have not been updated within a given period of time. Companies face similar challenges in identifying inactive records in a database. Therefore other potential applications of the methodology may include organizational memory management and knowledge management (managing the information life cycle and deciding how long the information should be kept or when storage should be moved outside the firm; Nilakanta et al., 2006), data management (deciding on whether or not to move less frequently accessed data to remote storage), customer relationship management (determining whether bank or credit card accounts exhibiting long inactivity periods are likely to be used again in the future), and asset management (determining when an asset may be considered abandoned property).

References

Bergquist, M., Ljungberg, J. (2001). The power of gifts: organizing social relationships in open source communities. *Information Systems Journal* 11(4):305–320.

Chengalur-Smith, S., Sidorova, A. (2003). Success and survival of open-source projects: A population ecology perspective. *Proceedings of the International Conference on Information Systems*. Seattle, WA.

Cox, D. R. (1975). Partial likelihood. *Biometrika* 62:269–276.

Craddock, C., Szydlo, R., Klein, J., Dazzi, F., Olavarria, E., Van Rhee, F., Pocock, C., Cwynarski, K., Apperley, J., Goldman, J. (2000). Estimating leukemia-free survival after allografting for chronic myeloid leukemia: a new method that takes into account patients who relapse and are restored to complete remission. *Blood* 96(1):86–90.

Crowston, K., Annabi, H., Howison, J., Masango, C. (2004). Towards a portfolio of FLOSS project success measures. *Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering*. Edinburgh.

Dempsey, J. B., Weiss, D., Jones, P., Greenberg, J. (2002). Who is an open source software developer? *Communications of the ACM* 45(2):67–72.

Efron, B. (1988). Logistic regression, survival analysis, and Kaplan–Meier curve. *Journal of the American Statistical Association* 83(402):414–425.

Fiore, N. (1999). From the tax adviser: unclaimed property audits. *Journal of Accountancy* 188(3):104.

Glass, L. R. (2003). Practical programmer: a sociopolitical look at open source. *Communications of the ACM* 46(11): 21–23.

Howerton, P. (1985). Computer crime (A Tutorial). *Proceedings of the 1985 ACM Annual Conference on The Range of Computing: mid-80's Perspective*. pp. 54–55.

Huntley, C. L. (2003). Organizational learning in open-source software projects: an analysis of debugging data. *IEEE Transactions on Engineering Management* 50(4):485–493.

Kevin, T. (2004). Linux creeps into the enterprise. *Network World* 21(21):24.

Lacy, S. (2006). California mulls open source. *BusinessWeek* (on-line) February 6, 2006.

Nilakanta, S., Miller, L., Zhu, D. (2006). Organizational memory management: technological and research issues. *Journal of Database Management* 17(1):85–94.

Norris, J. S. (2004). Mission-critical development with open source software: lessons learned. *IEEE Software* 21(1):42–49.

Paulson, J. W., Succi, G., Eberlein, A. (2004). An empirical study of open-source and closed-source software products. *IEEE Transactions on Software Engineering* 30(4):246–256.

Salazar, B. (2000). The Holocaust—Recovery of Assets from World War II: A Chronology (May 1995 to Present). CRS Report for Congress RL30262. http://www.opencrs.cdt.org/document/RL30262/, accessed Jan 13, 2007.

Silveira Chalita, L., Colosimo, E., De Souza Passos, J. (2006). Modeling grouped survival data with time-dependent covariates. *Communications in Statistics – Simulation and Computation* 35:975–981.