Empir Software Eng (2008) 13:147–184
DOI 10.1007/s10664-007-9051-7

A statistical framework for analyzing the duration of software projects

Panagiotis Sentas · Lefteris Angelis · Ioannis Stamelos

Published online: 3 October 2007

Editor: Ross Jeffery

Abstract The duration of a software project is a very important feature, closely related to its cost. Various methods and models have been proposed in order to predict not only the cost of a software project but also its duration. Since duration is essentially the random length of a time interval from a starting to a terminating event, in this paper we present a framework of statistical tools, appropriate for studying and modeling the distribution of the duration. The idea for our approach comes from the parallelism of duration to the life of an entity which is frequently studied in biostatistics by a certain statistical methodology known as survival analysis. This type of analysis offers great flexibility in modeling the duration and in computing various statistics useful for inference and estimation. As in any other statistical methodology, the approach is based on datasets of measurements on projects. However, one of the most important advantages is that we can use in our data information not only from completed projects, but also from ongoing projects. In this paper we present the general principles of the methodology for a comprehensive duration analysis and we also illustrate it with applications to known data sets. The analysis showed that duration is affected by various factors such as customer participation, use of tools, software logical complexity, user requirements volatility and staff tool skills.

Keywords Duration of a software project · Survival analysis · Software cost estimation · Project cancellation

P. Sentas · L. Angelis (✉) · I. Stamelos
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
e-mail: lef@csd.auth.gr

P. Sentas
e-mail: psentas@csd.auth.gr

I. Stamelos
e-mail: stamelos@csd.auth.gr

1 Introduction

The problem of accurately estimating the cost of a software project has been extensively studied in the software engineering literature. Since the cost, expressed either as work effort or productivity, is a random variable depending on a large number of factors or attributes of the project, a large part of the research concerns the fitting of statistical models to historical data sets with completed software projects (see for example references Boehm 1981; Shepperd and Schofield 1997; Clark et al. 1998; Kitchenham 1998; Maxwell et al. 2000; Angelis et al. 2001, and Maxwell 2002). These models are evaluated and used for predicting the costs of new software projects.

There is one more variable which plays a central role in the management of software projects and which is closely related to the cost: the time-to-market or project *duration*. In a study aiming to determine the critical features of a project and to take decisions based on them (such as budget planning, scheduling of activities or resource allocation), the duration can be considered as a dependent variable too, whose variability is affected by different project characteristics. However, project duration is a time variable and as such it has two peculiarities: First of all, it is characterized by two critical events, the initialization and the termination of the project which are marked by two relevant dates. Second, the duration can be continuously measured as a distance from any time point to the initial point. As a consequence, an organization at any time can conduct a study on the duration of projects that have been so far completed but it seems quite reasonable to utilize information from ongoing projects too. Indeed, the projects that have started in the past but have not finished by the time of the study contain significant information regarding the time they have stayed active so far.

In a traditional statistical analysis, the aforementioned features are ignored. The durations are considered as measures of only the completed projects while the durations of ongoing projects are overlooked as missing values. A typical duration analysis would involve fitting of a theoretical distribution in order to describe the probability distribution of durations and analysis of variance (ANOVA) or regression modeling in order to study the effect of different variables on duration. However, it is possible to exploit all available information, even from unfinished projects by setting the problem in the framework of a more general methodology formulated for time variables.

A similar problem, common in life sciences, is the study of *survival* of an entity or time length of a process triggered by a specific initial event and terminated by a subsequent terminal event. The general methodology, involving expansions of the traditional methods specially formulated for time data is known as *survival analysis*. Having in mind that project duration is essentially a time variable, it seems reasonable to set the duration analysis in the survival framework that has been extensively developed and studied in the statistical literature. Under this perspective, project duration can be abstractly considered as the life time of an entity (software project) from the starting event (specification) to the terminating event (delivery). It should be noted that as survival analysis has been applied to many other scientific areas, such as engineering and economics, depending on the application, other terms for survival time have been used, like *failure time* or *event history.* Analogous terms are used for the various probability functions related to survival times. In order to be consistent with the terminology of the software project management research, we adjusted the vocabulary of survival analysis to meet our needs for interpretation. In what follows, we use the general term *project duration analysis* (PDA) to refer to all statistical methods and tools from survival analysis that are used for studying the distribution of project duration and for identifying the factors that affect it.

The benefits of using the tools of PDA are twofold: first, we can construct probabilistic models for the duration utilizing all of the available information, not only from completed projects but also from projects that have not finished yet. Second, the models are expressed in a longitudinal manner, in the sense that the various probabilistic functions are represented by the evolvement of their values as time progresses. Representations of this form are useful for making "real time" inference about the duration, for example for determining the possibilities to complete a project within the following few months from now. In general, the modeling of duration is beneficial as it can be further be utilized for assessing the cost or the scheduling of a project either by cost models or by simulation runs.

Going over the main points of PDA, the duration of a software project (measured in days, months or years) is considered to be a *dependent variable* while the *diagnostic factors* (or simply *factors*) under study that may affect the duration, are the categorical attributes characterizing the projects. Duration may be affected by continuous variables as well (for example the project size), and these are called *covariates*. Factors and covariates constitute the set of *independent variables* or *predictors*. The duration of incomplete projects (*censored* observations) is defined as the time from the starting date until the date we stopped collecting data. In practice this date coincides with the current date, i.e. the date in which we wish to make inferences and predictions using all the available information up to now. The application of PDA therefore requires a specific *study starting date* and a *study termination date*. The projects participating in the study are the ones that were initiated on or after the study starting date and before the termination date.

The paper is organized so as to present first the basic principles of PDA and then to illustrate its use by applications to real data published in the literature. An obvious problem with published data is that they contain only completed projects and is therefore impossible to present the method with unfinished projects. In order to include unfinished projects in our analysis, we simulated their existence by adjusting arbitrarily the final date of the study.

The comparative analysis showed interesting results regarding the effect of various factors on duration such as customer participation, using of tools, software logical complexity, user requirements volatility and staff tool skills.

The rest of the paper is organized as follows: In Section 2 we review some related work. In Section 3, we describe the basic principles of the PDA method with several examples using real data. In Section 4 we present an application of the methods to a known dataset from the literature. In Section 5, we discuss some estimation situation where PDA can be particularly useful and we provide an extensive discussion regarding the validity of PDA as a statistical analysis tool. Finally, in Section 6 we conclude with an overview and directions for future work.

2 Related Work

Duration plays central role in project management. Especially for software projects, various methods have been proposed for the analysis of project duration. In this section we review the most indicative approaches for this problem.

The basic equations of COCOMO 81 and COCOMO II (Boehm 1981; Boehm et al. 2000) contain a formula for deriving the duration of a project from its effort. Other models in the form of equations associating duration with effort and size have been also presented, such as the Putnam model (Putnam and Myers 2003). The ISBSG Reality Checker (http://www.isbsg.org/ISBSG.nsf/weben/Reality%20Check) is a simple software tool that utilizes the data in the ISBSG repository and regression analysis to generate estimates of the work effort and elapsed time (duration) required to carry out and complete a software development project.

In (Rainer and Shepperd 1999), a longitudinal case study of a project at IBM Hursley Park is presented. The paper, based on data collected from interviews, meetings and project documents deals with duration and especially with the schedule behavior and a variety of related factors including planned versus actual progress, resource allocation and functionality delivered.

A study concerning the distributions of software project duration and effort from the ISBSG, release four dataset is presented in (Oligny et al. 2000). The authors fit normal distributions to the logarithms of both random variables and they model their relation using regression models for different development platforms.

The forecasting of software project duration is considered in (Lind and Sulek 2000) where a neural network model is proposed for modeling software project overruns. The accuracy of the model is tested on actual software project management data in comparison to a regression model. The results show that neural network modeling does better than traditional regression and can be used for managerial judgments in knowledge work.

A method for the identification and explanation of risky projects is presented in (Mizuno et al. 2001), combining results from a risk questionnaire and a multiple regression model. The purpose of the approach is to efficiently estimate the cost and the duration of software projects.

The dynamic relationship between project duration and project effort is investigated in (Barry et al. 2002). A two-stage dynamical model is developed and evaluated empirically for this purpose. The results show significant and positive relationship between duration and effort.

An alternative approach is presented in (Jain 2002) where the author suggests the use of a task allocation algorithm for distributed teams across time zones, aiming to minimize the project duration. The algorithm is applied to a 24-h software development model exploiting the globalization and the internet.

Much of the work by Rainer and Hall concerns project duration. For example in (Rainer and Hall 2003), 26 factors that potentially affect software process improvement are explored while in (Rainer and Hall 2004) the close relationship between project duration and project effort is discussed. Moreover, data from two projects are used to identify the areas in the projects where poor progress was occurring and to investigate the causes of this poor progress.

Finally, in (Ayal 2004), regression analysis and structural equation modeling are used to construct and validate an explanatory model for project duration extensions. Furthermore, the effect of scope changes and other related drivers on project delivery times are also analyzed.

As we can see from the aforementioned references, there is a quite wide range of methods applied to the analysis of duration. The diversity of techniques shows that there is a need for systematic way of handling longitudinal data in software project management. Although there is a general tendency to use statistical methodology (see Oligny et al. 2000; Lind and Sulek 2000; Mizuno et al. 2001, and Ayal 2004), the methods that have been applied so far, like the least squares regression, are the traditional ones and they are applicable to any variable.

The motivation of our work was the realization of the duration’s particular nature as time variable. Processes that are developed and observed in time have their own characteristics and peculiarities. Furthermore, there is much statistical research work available from other scientific areas, suitable for time variables with incomplete data.

In (Sentas and Angelis 2005) and (Angelis and Sentas 2005), the authors presented some basic notions and examples regarding the use of Survival Analysis for studying duration. In this paper we give a comprehensive account of all the important aspects of the theory adjusted for the project duration problem along with extensive experimentation on real data. Furthermore, we consider extensively the interpretation of the statistical notions and results and also the practical implications of the method. Although the main characteristic of the proposed methodology is that it models the probabilistic distribution of the duration in a non-parametric way, i.e. no theoretical distribution is assumed in advance, we will present some aspects of the parametric approach which can be useful in certain situations.

3 Theory of Project Duration Analysis

The statistical methods that will be described here have been introduced since long in medical and engineering research for studying the survival time of patients or the reliability of devices. During the latest decades, these methods have acquainted a wide spread in various scientific areas such as marketing, criminology, epidemiology, and even social and behavioral sciences.

The key notion of *survival time* is defined as the time to the occurrence of a predefined, terminating event. This event has a critical meaning according to the application domain (death, failure, response to treatment, etc). In our case, we will use the term *duration time* or simply *duration* to refer to the time from specification until delivery of a software project. The *duration data* can therefore include apart from the duration other variables characterising the projects. The study of duration data is focused on estimation of the probabilities of duration, comparisons between distributions of different projects and identification of prognostic factors related to duration.

In general, we can distinguish two types of approaches in studying the distribution of duration: the *parametric*, where the theoretical distribution fitting the data is assumed to be known (e.g. exponential, Weibull, lognormal, gamma) and we need to estimate its parameters and the *nonparametric* where the distribution is unknown. However, there is a peculiarity in the duration data that makes the use of conventional methods impossible, i.e. the precise duration times of some projects may not be known. That occurs when some projects in the dataset have not come across the terminal event, i.e. they are not completed at the end of the study or at the time of the analysis. These cases in a dataset are generally called *censored times* but since our data contain software projects we can refer to them as *incomplete* or *ongoing* projects. These can also occur when some projects are lost to follow-up after a period of study, i.e. the researcher loses any contact with the developers and does not know whether they have been completed or not.

3.1 Types of Censored Duration Times

In literature, there are known three types of censoring which in the case of projects are:

Type I censoring This occurs in studies (e.g. controlled experiments) which start with the simultaneous initiation of a certain number of projects. Sometimes, due to time and/or cost limitations, the researchers decide to observe the development of projects for a predefined period of time, after which the incomplete projects are forced to terminate or are abandoned. Duration times recorded for the projects that terminated during the study period are the times from the start of the experiment to their termination. The duration times of the abandoned projects are considered censored and they are recorded as at least the length of the study period. Moreover, it is possible that some projects may be abandoned or fail during the study period. Their durations, from the start of study to abandonment, are also considered censored.

Type II censoring According to this scenario, the researchers wait for a predefined percentage of projects (for example 80 or 90%) to be normally completed. Then, the incomplete projects are abandoned and their duration is considered censored and equal to the largest complete duration.

Type III or random censoring This is the most common case in software project research. The period of study is predefined and projects start at different times during that period. For those which are completed before the end of the study their precise duration times are known. Others may be lost to follow-up and, more often, others may be ongoing at the end of the study. A “lost” project has censored duration at least the time from initiation to the last contact. An incomplete project has also censored duration at least from initiation to the end of the study.

The aforementioned types of censoring are types of *right censoring*. However, it is possible to encounter types of *left censoring* and *interval censoring*. Left censoring may occur when it is known that a project was terminated before a certain time *t*, but the exact termination time is unknown. Interval censoring may occur when a project is known to have been terminated between times *a* and *b*. These types of censoring may be present in various circumstances, for example in the open-source development.

As an illustrative example, consider the case where in our data we have the 18 projects from the ISBSG7 dataset (ISBSG 2005) with data quality rating A or B, for which the development type is characterised as “Re-development” and the implementation date along with duration in months are known (non-missing). We can therefore calculate the starting date and make the plot in Fig. 1 to depict the duration of each project from the starting to

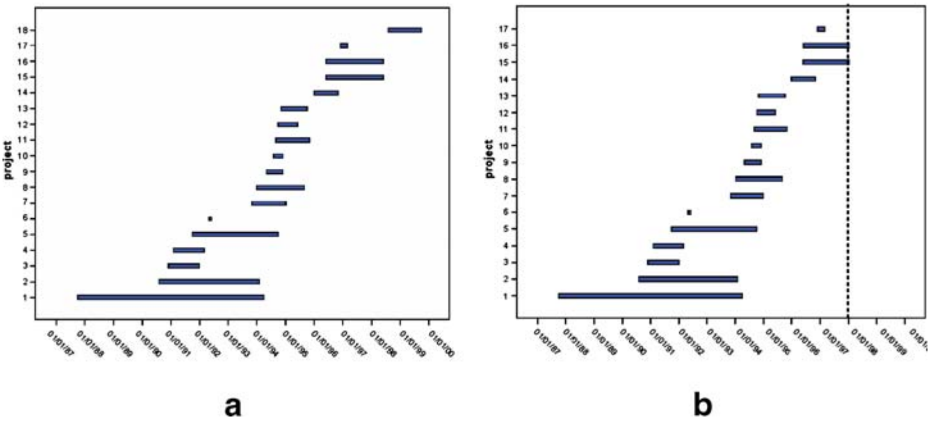


Fig. 1 Durations of (**a**) complete and (**b**) incomplete projects of the re-development type

the implementation. Note that the earliest starting time of a project is 1/10/87 while the latest implementation time of a project is 1/10/99. So, if we arrange the time of study between 01/01/1987 and 01/01/2000 we will have all of our projects completed. However, suppose that the cut-off date of our study is 01/01/1998. Then, in our data set we have only 17 projects (since the 18th started after the cut-off date) two of which are incomplete (right-censored). The duration of the two incomplete projects will be considered in the analysis as the time from their starting date to the cut-off date.

In Fig. 1a we can see the durations of the 18 completed projects in the time interval from 1/1/87 to 1/1/2000. In Fig. 1b we can see the 17 projects (15 complete and 2 incomplete) in the time interval from 1/8/1987 to 1/1/1998.

Under the assumption that there are no changes over time in the conditions of the study or in the types of projects enrolled in the study, the projects can be set to a common start and be arranged in rank order of their duration ignoring their starting and implementation date. Such an arrangement of the projects of Fig. 1 can be seen in Fig. 2 where in a we have the complete durations of all the 18 projects while in b the durations of the two incomplete projects are followed by the "+" sign.

3.2 Functions of Duration Times

The duration times of software projects are subject to random variations, and therefore can be considered as values of a random variable forming a distribution. The distribution of duration times can be described by three functions: (1) the *probability density function*, (2) the *duration function* and (3) the *conditional completion rate*. These three functions are mathematically equivalent in the sense that if one of them is known, the other two can be derived by a mathematical formula. However, their interpretation is different and can be used in different ways to describe the data. Also, these functions are estimated from the data by approximation methods. In what follows, we denote by T the positive continuous random variable representing the duration time of software projects while by t we denote its values.

The *Probability Density Function* (*PDF*) is a nonnegative function denoted by $f(t)$ and is the most known probability function for any random variable. It is defined as the limit of the probability that a project is completed in the time interval $(t, t+\Delta t)$ per unit width Δt.

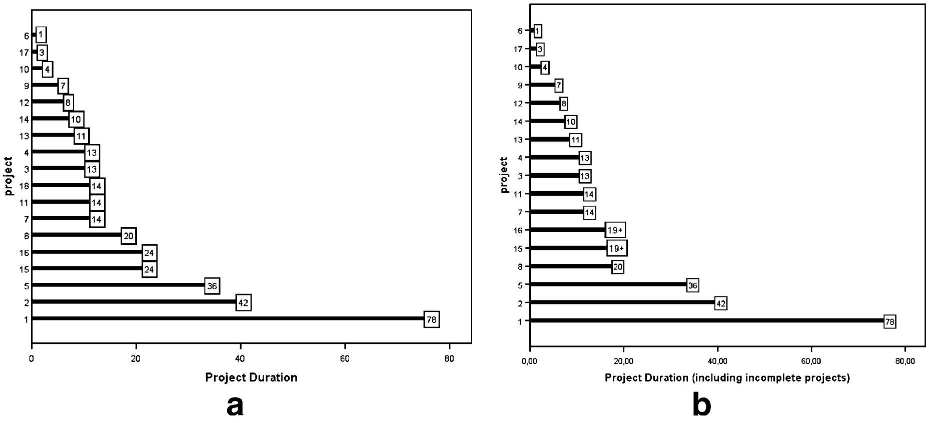


Fig. 2 Duration of (**a**) complete and (**b**) incomplete projects rearranged in rank order (the *plus sign* denotes duration of incomplete project)

The mathematical formula defining PDF is:

$$f(t) = \lim_{\Delta t \to 0} \frac{P(t < T < t + \Delta t)}{\Delta t} \tag{1}$$

The graph of $f(t)$ is called the *density curve* and the area between the curve and the *t*-axis is equal to 1. In practice, if there are no ongoing projects (censored cases), $f(t)$ is estimated as the proportion of projects completed in an interval per unit width:

$$\hat{f}(t) = \frac{\text{number of projects completed in the interval beginning at time t}}{(\text{total number of projects}) \times (\text{interval width})} \tag{2}$$

The density curve can be used to compute the proportion of projects that are completed in any time interval and also to locate peaks of high frequency of completed projects. Another function defining the distribution of the duration times is the *Cumulative Distribution Function* (*CDF*) denoted by *F*(*t*):

$$F(t) = P(T \leq t) = \int_0^t f(u)\mathrm{d}u \tag{3}$$

The *Duration Function* (DF—the well-known analogue in the literature is *Survival Function*) is denoted by D(*t*) and is defined as the probability that the duration of a project is longer than *t*:

$$\mathrm{D}(t) = P(T > t) = 1 - F(t) \tag{4}$$

D(*t*) is a non-increasing function of time *t* with the properties:

$$D(t) = \begin{cases} 1 & \textit{for} \quad t = 0 \\ 0 & \textit{for} \quad t = \infty \end{cases} \tag{5}$$

The graph of D(*t*) is called the *duration curve* and its shape can be interpreted in terms of short or long duration. Specifically, a steep curve shows short duration while a gradual or flat curve is an indication of longer duration. When there are no ongoing projects, the duration function is estimated as the proportion of projects with duration longer than *t*:

$$\hat{D}(t) = \frac{\text{number of projects with duration} > t}{\text{total number of projects}} \tag{6}$$

The *Conditional Completion Rate* (CCR—known in the literature as *Hazard Function*) is denoted by c(*t*) and is defined as the probability of completion during a very small time interval, assuming that the project is still active by the beginning of the interval:

$$\mathrm{c}(t) = \lim_{\Delta t \to 0} \frac{P(t < T < t + \Delta t | T \geq t)}{\Delta t} \tag{7}$$

An alternative definition is given by:

$$c(t) = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{\mathrm{D}(t)} \tag{8}$$

The conditional completion rate is a measure of the tendency for project completion as a function of the duration of the project. When there are no ongoing projects, it can be estimated as the proportion of projects completed in an interval per unit time, given that they are still active at the beginning of the interval. However, in practice an "actuarial" estimation is used:

$$\hat{c}(t)=\frac{(\text{number of projects completed in the interval beginning at time t})/(\text{interval width})}{\text{number of projects active at } t-\text{number of completed in the interval}/2} \tag{9}$$

The curve of c(*t*) can have any shape. It may be increasing, decreasing, constant, or indicate a more complicated process. It should be noted that c(*t*) has inverse meaning of D(*t*) in the sense that D(*t*) is focused on the extension of the project duration while c(*t*) is focused on the termination of the project. That is why large values of D(*t*) correspond to small values of c(*t*) and inversely. The importance of CCR in duration analysis will be apparent later when we will present the Cox regression models.

The *cumulative conditional completion rate* (CCCR) is defined as

$$\mathrm{C}(t)=\int_0^t c(u)\mathrm{d}u \tag{10}$$

In order to summarize the relations between the aforementioned functions, we give the following formulas:

$$f(t)=F'(t)=-\mathrm{D}'(t)=\mathrm{c}(t)\exp\left(-\int_0^t \mathrm{c}(u)\mathrm{d}u\right)=\mathrm{C}'(t)\exp(-\mathrm{C}(t)) \tag{11}$$

$$F(t)=\int_0^t f(u)\mathrm{d}u=1-\mathrm{D}(t)=1-\exp\left(-\int_0^t \mathrm{c}(u)\mathrm{d}u\right)=1-\exp(-\mathrm{C}(t)) \tag{12}$$

$$\mathrm{D}(t)=\int_t^\infty f(u)\mathrm{d}u=1-F(t)=\exp\left(-\int_0^t \mathrm{c}(u)\mathrm{d}u\right)=\exp(-\mathrm{C}(t)) \tag{13}$$

$$\mathrm{c}(t)=\frac{f(t)}{\int_t^\infty f(u)\mathrm{d}u}=\frac{F'(t)}{1-F(t)}=-(\log \mathrm{D}(t))'=\mathrm{C}'(t) \tag{14}$$

$$\mathrm{C}(t)=-\log\left(\int_t^\infty f(u)\mathrm{d}u\right)=-\log(1-F(t))=-\log \mathrm{D}(t)=\int_0^t \mathrm{c}(u)\mathrm{d}u \tag{15}$$

Suppose for example that based on the 18 complete projects of the re-development type discussed earlier we wish to describe theoretically the distribution of the random variable *T* which produced their durations. By the Kolmogorov-Smirnov test (p=0.880) we can assess that the distribution of their logarithms is not significantly different from the Normal distribution with mean μ=2.516 and standard deviation σ=1.0108 (these values are estimated from the sample). Thus we can assume that the random variable *T* has the log-normal distribution Λ(2.516, 1.0108) with PDF given by the equation (see Lee and Wang 2003, pp.143–148):

$$f(t) = \frac{1}{t\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(\log t - \mu)^2\right) \tag{16}$$

The CDF of the log-normal distribution fitted to our data is computed by the following relations:

$$\begin{aligned} F(t) &= \tfrac{1}{\sigma\sqrt{2\pi}} \int_0^t \tfrac{1}{u} \exp\left(-\tfrac{1}{2\sigma^2}(\log u - \mu)^2\right) \mathrm{d}u \\ &= \tfrac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\log t} \exp\left(-\tfrac{1}{2\sigma^2}(y - \mu)^2\right) \mathrm{d}y = \\ &= \Phi\left(\tfrac{\log t - \mu}{\sigma}\right) \end{aligned} \tag{17}$$

where by $\Phi(x)$ we denote the CDF of the standard normal distribution, the values of which are computed numerically and are given in tables. The graphs of PDF and CDF of the lognormal distribution are given in Fig. 3a and b, respectively.

Having computed the values of the CDF, we can easily compute the values of DF by Eq. (4). The graph of the duration function of the log-normal distribution is given in Fig. 4.

The CCR and the CCCR are also computed easily by Eqs. 14 and 15 and their graphs are given in Fig. 5a and b, respectively.

The approach we just described is the parametric approach, in the sense that we assumed that the distribution of the duration is the lognormal. If we are unable to make any assumption on the distribution, we can estimate these curves by the observed data using Eqs. 2, 6 and 9. First of all, we have to define time intervals. Since the number of our complete projects is quite small, there is no meaning in defining a large number of small intervals. Suppose that the durations are grouped into intervals of 10 months. Then, we form Table 1 the columns of which contain the necessary information for the estimation of the functions. The calculations are described below:

$$\hat{f}(0) = \tfrac{5}{18\times 10} = 0.028,\ \hat{f}(10) = \tfrac{7}{18\times 10} = 0.039,\ \hat{f}(20) = \tfrac{3}{18\times 10} = 0.017, \ldots$$

$$\hat{D}(0) = \tfrac{18}{18} = 1,\ \hat{D}(10) = \tfrac{13}{18} = 0.72,\ \hat{D}(20) = \tfrac{6}{18} = 0.33, \ldots$$

$$\hat{c}(0) = \tfrac{5/10}{18-5/2} = 0.032,\ \hat{c}(10) = \tfrac{7/10}{13-7/2} = 0.074,\ \hat{c}(20) = \tfrac{3/10}{6-3/2} = 0.067, \ldots$$

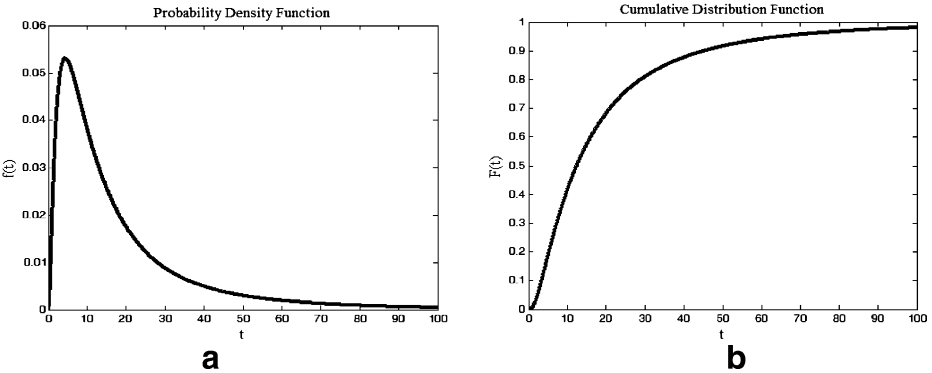
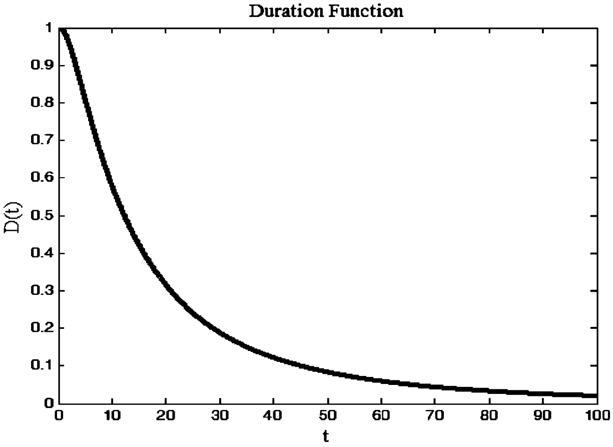


Fig. 3 The (**a**) PDF and the (**b**) CDF of the log-normal distribution $\Lambda(2.516, 1.0108)$

Fig. 4 The DF of the log-normal distribution $\Lambda(2.516, 1.10108)$



Based on the estimations of Table 1 we can now plot the three curves in Figs. 6, 7 and 8. In general, nonparametric methods are considered less efficient than parametric methods when duration times follow a theoretical distribution. However, when no suitable theoretical distribution is known, are more efficient. Usually, nonparametric methods are applied for a preliminary analysis of duration data since the graphs we obtain can provide significant information for choosing a theoretical distribution.

3.3 The Kaplan-Meier Method

The parametric and the nonparametric approaches for estimating the duration functions we already described can be applied when all the projects are completed. In the case where the data contain ongoing or incomplete projects we need to use other nonparametric estimation techniques. The most widely used method for estimating the DF in the presence of censored values is known as *product-limit* (P-L) or *Kaplan-Meier* (K-M) method (Kaplan and Meier 1958) and is the one that we will describe next.

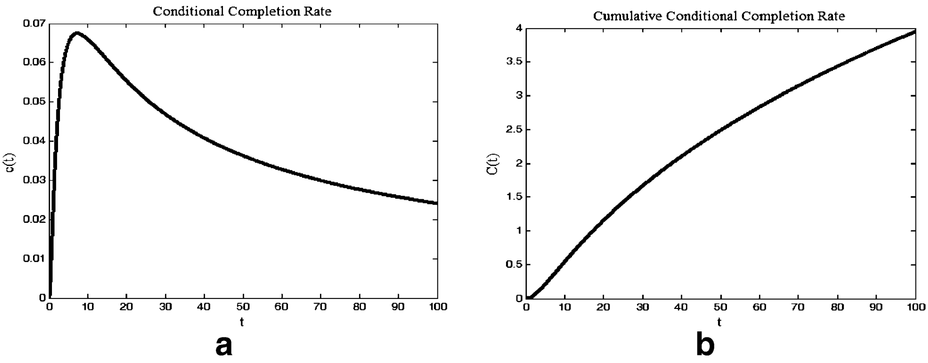


Fig. 5 The (**a**) CCR and the (**b**) CCCR of the log-normal distribution $\Lambda(2.516, 1.0108)$

Table 1 Duration data and estimated functions of 18 complete projects of re-development type

Interval start time	Number of active projects at the start of the interval	Number of projects completed during the interval	$\hat{f}(t)$	$\hat{D}(t)$	$\hat{c}(t)$
0	18	5	0.028	1.00	0.032
10	13	7	0.039	0.72	0.074
20	6	3	0.017	0.33	0.067
30	3	1	0.006	0.17	0.040
40	2	1	0.006	0.11	0.067
50	1	0	0.000	0.06	0.000
60	1	0	0.000	0.06	0.000
70	1	1	0.006	0.06	0.200

First, it is necessary to arrange in ascending order the duration times of all the projects participating in the study. Let us denote these successive times by $t_1 < t_2 < t_3 ...,$. Then, at each time point t_i we calculate

(a) the number of terminations, denoted by d_i and
(b) the number of censored durations, denoted by c_i.

The number of projects *waiting for termination* is denoted by n_i and is computed by:

$$n_i = n_{i-1} - d_{i-1} - c_{i-1} \tag{18}$$

Based on the above numbers, the *conditional probability of duration* is estimated by:

$$P(T > t_i | T > t_{i-1}) = \frac{n_i - d_i}{n_i} = 1 - \frac{d_i}{n_i} \tag{19}$$

Fig. 6 Estimation of the PDF curve from the 18 complete projects

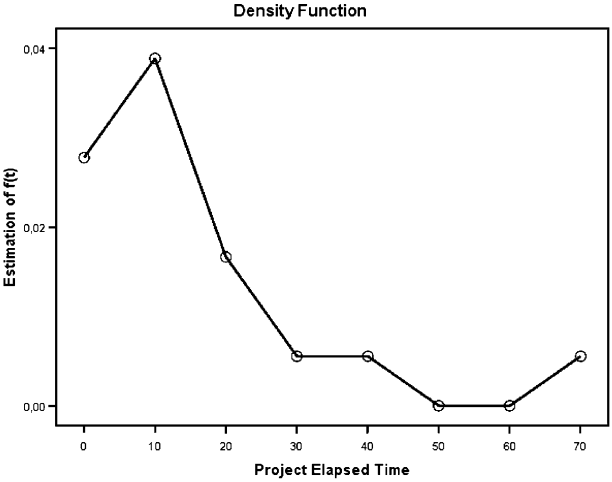
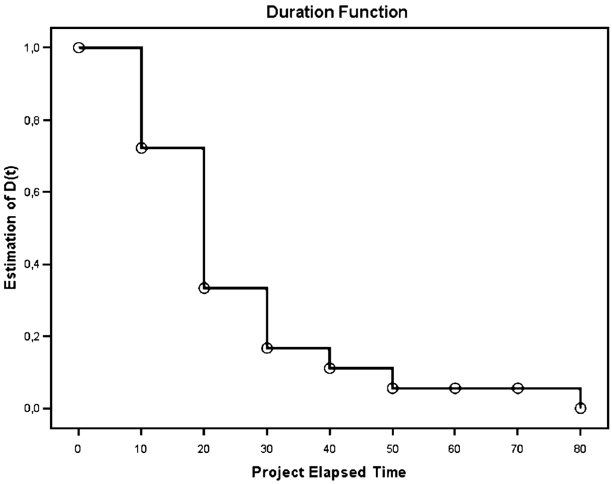


Fig. 7 Estimation of the DF curve from the 18 complete projects



The corresponding *unconditional probability* coincides with the notion of DF and is estimated by the recurrent relation:

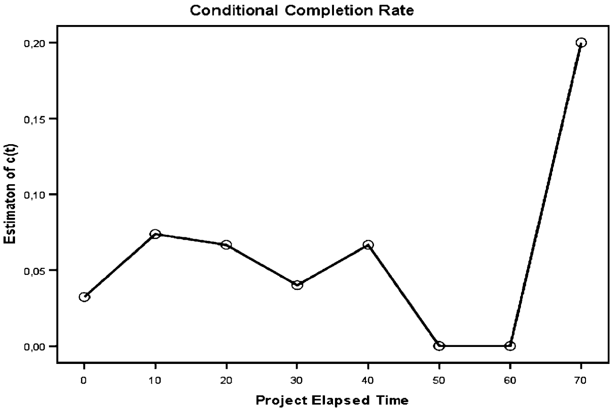
$$\hat{D}(t_i) = P(T > t_i) = P(T > t_i | T > t_{i-1}) \cdot P(T > t_{i-1}) = \left(1 - \frac{d_i}{n_i}\right)\hat{D}(t_{i-1}) \tag{20}$$

In order to derive an explicit expression of DF we denote by t_0 the starting point of the study where we can safely assume that $P(T > t_0) = 1$. Under this assumption, the recurrent relation 20 results in the explicit relation of the K-M estimate of the duration curve:

$$\hat{D}(t_i) = \prod_{j=1}^{i}\left(1 - \frac{d_j}{n_j}\right) \tag{21}$$

The corresponding values of the CCCR, $\hat{C}(t_i)$, are estimated using the third relation in Eq. 15 from the K-M estimations $\hat{D}(t_i)$. The estimations $\hat{D}(t)$ and $\hat{C}(t)$ can then be graphically represented by curves useful for inferences and comparisons. For more details see (Parmar and Machin 1995) and (Lee and Wang 2003). In order to illustrate the K-M method we apply it to the set of the 17 re-development projects we discussed earlier which

Fig. 8 Estimation of the CCR curve from the 18 complete projects



contains two incomplete projects. In the columns of Table 2 we give all the necessary statistics for the estimation of $\hat{D}(t_i)$ and $\hat{C}(t_i)$. The corresponding curves are given in panels (a) and (b) of Fig. 9.

Note that the values of $C(t)$ are not probabilities and the interpretation of the function is not easy. However, it is a function that plays a very important role in the duration analysis theory since it is used for estimation of distribution parameters and comparisons between non-parametric and parametric distributions through models connecting the duration time t and the values of C(t). For example, if the duration follows a log-normal distribution $\Lambda(\mu, \sigma)$, it can be proved (see Lee and Wang 2003) that

$$\log t = \mu + \sigma \Phi^{-1}\left(1 - e^{-C(t)}\right) \tag{22}$$

where by $\Phi^{-1}(x)$ we denote the inverse of the standard normal CDF. Now, even if we have censored values as in the previous example, we can test whether the duration follows the log-normal distribution and furthermore we can estimate its parameters μ and σ. This is done by transforming the second and the last column of Table 2 and next by simply fitting the least squares regression line between log$t$ and $\Phi^{-1}\left(1 - e^{-C(t)}\right)$. The scatter-plot of the transformed values is given in Fig. 10 where we can see that the fitting of the linear model is very good (r^2=0.924) thus the log-normal distribution is fitted well and furthermore, the equation of the straight line is

$$\log t = 2.421 + 1.131 \times \Phi^{-1}\left(1 - e^{-C(t)}\right) \tag{23}$$

from which we can estimate the parameters of the log-normal distribution μ=2.421 and σ=1.131. Note that these values are slightly different from the ones we found when we used the 18 completed projects.

It is therefore obvious that the K-M method through the non-parametric estimation of the duration curve can also provide evidence regarding the nature of the underlying theoretical distribution and also to estimate its parameters. So, the previous example shows that the K-M curve enables us to describe graphically all the durations in the available dataset, even the censored ones. Furthermore, whenever a known theoretical distribution (like the log-normal) can be fitted to the data, the K-M non-parametric method is an effective way to

Table 2 Calculations for the K-M estimation of duration curve

i	t_i	d_I	c_i	n_i	$\hat{D}(t_i)$	$\hat{C}(t)$
1	1	1		17	$1 - \frac{1}{17} = 0.941$	−log(0.941)=0.061
2	3	1		16	$\left(1 - \frac{1}{16}\right) \times 0.941 = 0.882$	−log(0.882)=0.125
3	4	1		15	$\left(1 - \frac{1}{15}\right) \times 0.882 = 0.824$	−log(0.824)=0.194
4	7	1		14	0.765	0.268
5	8	1		13	0.706	0.348
6	10	1		12	0.647	0.435
7	11	1		11	0.588	0.531
8	13	2		10	0.471	0.754
9	14	2		8	0.353	1.041
10	19		2	6	0.353	1.041
11	20	1		4	0.265	1.329
12	36	1		3	0.176	1.735
13	42	1		2	0.088	2.428
14	78	1		1	0.000	

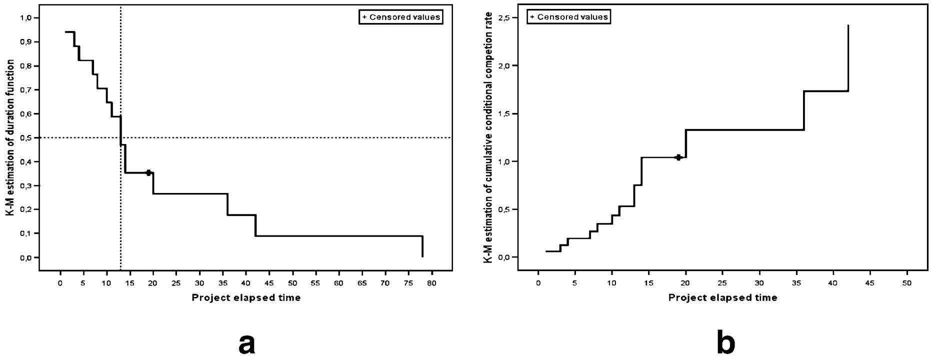


Fig. 9 K-M estimation of (**a**) the duration function and (**b**) the cumulative conditional completion rate

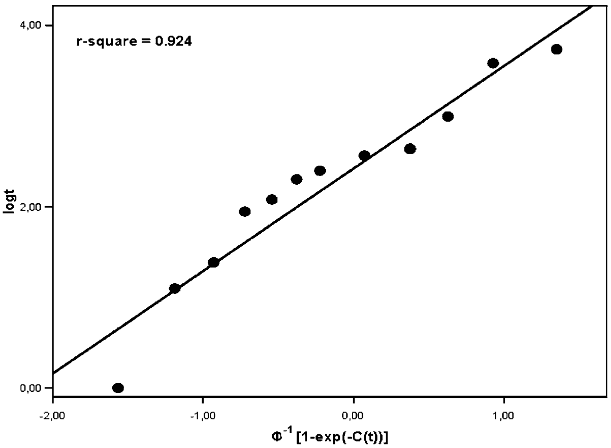
assess the goodness of fit. On the other hand, when the duration function cannot be expressed by a known theoretical distribution, the nonparametric K-M approach is the only means for studying duration.

There are certain limitations in order to apply the K-M method in real data. First of all, it is assumed that the projects participating in the study are independent, in the sense that the duration of any project does not depend on any other project. Second, the censored times should be independent of the duration times. This assumption holds when a project is still active at the end of the study period and its development is continued normally. However, the assumption is violated if the project is forced to terminate unfinished, due to serious problems in the development process. In cases where there is such inappropriate censoring, the method is not applicable. A third important limitation is that there are no changes over time in the conditions of the study or in the types of projects enrolled in the study. From all the above we can conclude that we should be very careful in choosing the projects for such a study in order to be more or less of the same type.

3.4 Useful Statistics for Duration Analysis

The estimated duration function or the duration curve is used to find the *median duration* denoted by *M* which essentially is the 50th percentile of the duration times. Other

Fig. 10 Assessing the linear relationship between transformations of time and CCCR



percentiles (e.g. 25th and 75th) of duration times are also useful especially for comparing duration distributions of two or more groups. Although the mean is the most known measure of central tendency of a distribution, in duration distributions which are generally skew, the median is preferable since a small number of projects with extremely long or short durations can affect the mean duration time to be unreasonably large or small.

If all the projects in a dataset are completed, then the median duration is just the median of all the duration times. However, in the case where censored times are present, the K-M curve is first estimated and the median is found as the value of M satisfying the equation $\hat{D}(M) = 0.5$. The interpretation is quite simple: it is the time in which 50% of the projects will be terminated while the other 50% will be active beyond this time. Note that since the K-M method estimates a step type (and not a smooth) curve, some problems in the calculation of M may occur. These are solved by interpolation techniques (see Lee and Wang 2003). The statistical programs provide M automatically along with confidence intervals.

As for example we can see from Fig. 9 that the median duration of the 17 re-development projects, with two censored times, is 13 months. The horizontal dashed reference line from 0.5 intersects the duration curve in a point which corresponds to M=13 (vertical dashed line). This means that based on our sample, 50% of re-development projects are expected to be completed in 13 months. A 95% confidence interval for the median duration computed by SPSS is from 10 to 16 months. In a similar way we can find the 25% percentile to be 36 months (i.e. 25% of the projects are active beyond 36 months) and the 75% percentile 8 months (i.e. 75% of the projects are active beyond 8 months). Note that the estimation of the mean is 21.3 months, a value which alone is quite misleading for the whole distribution.

3.5 Confidence Intervals for the Duration Curve

As the values $\hat{D}(t)$ obtained by the K-M method are estimations from a sample, they are subject to error. It is therefore useful to accompany the duration curve by confidence intervals (CI) which are also plotted as curves below and above the duration curve. The CI are calculated from the standard error of the estimation, denoted by $SE\left\{\hat{D}(t)\right\}$. There are several ways in the literature to estimate the standard error, but the most usual method is the Greenwood's formula (Parmar and Machin 1995; Collet 1994; Venables and Ripley 2002):

$$\mathrm{SE}\left\{\hat{D}(t)\right\} = \hat{D}(t)\left(\sum_{j=0}^{t-1}\frac{d_j}{n_j\left(n_j - d_j\right)}\right)^{1/2} \tag{24}$$

where the statistics d_j and n_j were defined earlier in the description of K-M method. The CI can then be calculated by the following ways:

If we assume that $\hat{D}(t)$ has a normal distribution, the bounds of a 100(1−a)% CI "*on the plain scale*" are given by the following relations:

$$\left[\hat{D}(t) - z_{a/2} \times \mathrm{SE}\left\{\hat{D}(t)\right\},\quad \hat{D}(t) + z_{a/2} \times \mathrm{SE}\left\{\hat{D}(t)\right\}\right] \tag{25}$$

where by $z_{a/2}$ we denote the point where for a random variable Z having the standard normal distribution it holds that $P\left(Z > z_{a/2}\right) = a/2$.

Sometimes it is more realistic to assume that $\log \widehat{D}(t)$ follows a normal distribution so in this case a $100(1-a)\%$ CI *"on the log scale"* is given by

$$\left[\hat{D}(t) \times \exp\left(-z_{a/2} \times \frac{\mathrm{SE}\left\{\hat{D}(t)\right\}}{\hat{D}(t)}\right),\quad \hat{D}(t) \times \exp\left(+z_{a/2} \times \frac{\mathrm{SE}\left\{\hat{D}(t)\right\}}{\hat{D}(t)}\right)\right] \tag{26}$$

An important disadvantage of the previous CI is that the computed bounds are not always in the interval [0,1]. Since the values of D(t) represent probabilities, this may lead to irrational results. This problem is solved if we assume that $\log\left(-\log\hat{D}(t)\right)$ follows a normal distribution and in this case a 95% CI *"on the log-log scale"* is given by

$$\left[\hat{D}(t)^{\exp\left(+z_{a/2} \times \frac{\mathrm{SE}\left\{\hat{D}(t)\right\}}{\hat{D}(t)\log\hat{D}(t)}\right)},\quad \hat{D}(t)^{\exp\left(-z_{a/2} \times \frac{\mathrm{SE}\left\{\hat{D}(t)\right\}}{\hat{D}(t)\log\hat{D}(t)}\right)}\right] \tag{27}$$

Note that for a 95% CI which is the most commonly used, we have $z_{a/2} = 1.96$. In the following Figs. 11 and 12, the corresponding 95% CI of the duration curve obtained earlier for the 17 projects are given on the plain and the log–log scale. These plots are produced automatically by the S statistical language where there are many other types of CI available for the duration curve.

The standard error and the confidence intervals indicate the precision of the estimated curve by the K-M method. In general, wide intervals show low precision. This precision is affected by the proportion of completed projects available in the dataset. It is clear that the K-M curve can be drawn for any proportion of complete and incomplete (censored) projects, but the confidence intervals are generally wider when the proportion of complete projects is small.

In order to get an idea of how the confidence intervals are affected by the different proportions of complete and incomplete data, let us consider the longitudinal evolution of the small dataset we use for our illustrations. The 18 re-development projects are considered as successive entries in the dataset assuming that the duration analysis study is repeated every year with cutoff dates starting from 1/1/1993 to 1/1/2000.

The evolution of the dataset is described in detailed in Table 3. In the four panels of Fig. 13 we can see the K-M curves and the corresponding log–log confidence intervals of

Fig. 11 95% CI for the duration curve on the plain scale

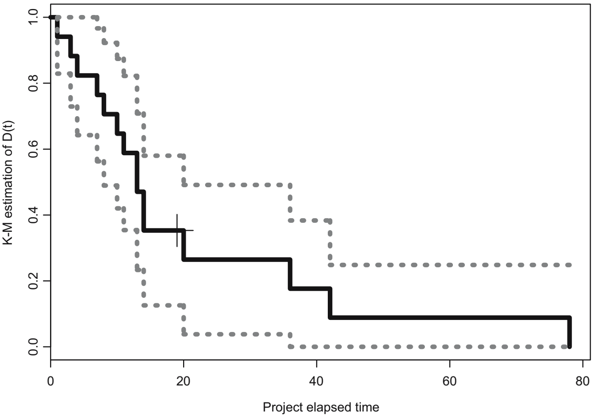
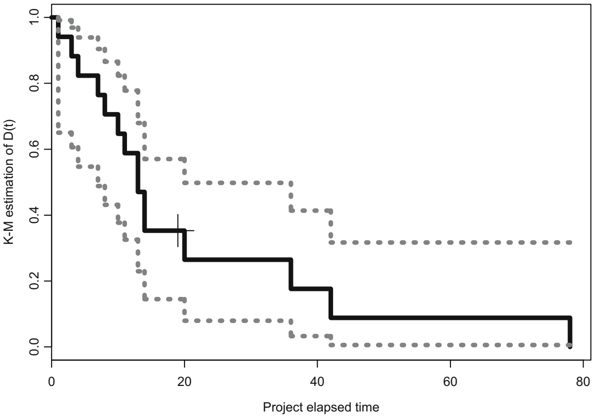


Fig. 12 95% CI for the duration curve on the log–log scale



the data we have in four different cutoff dates. The specific dates were selected because they present quite different curves. What we can clearly see is that although the K-M curve can be plotted for only three completed and three incomplete projects, the confidence intervals are very wide. We can also see that as new projects enter in the dataset and the proportion of completed is growing, the K-M is becoming more and more smooth and the confidence intervals more and more narrower. It is also worth noting (Parmar and Machin 1995, p. 35) that spurious (large) jumps or (long) flat sections may sometimes appear in the K-M curve due to large proportions of censored observations.

In general, we can say that the inclusion of the incomplete projects in the analysis adds valuable information. For example, consider the dataset we suppose to have for the 1/1/93 cutoff date of Table 3. The three completed projects have durations up to 13 months. However, the inclusion of the three active (censored) projects clearly shows the possibility for much longer durations, i.e. more than 60 months. Of course, the confidence intervals are quite wide, but the curve is much more informative than it would be with only the three completed projects.

3.6 Comparison of Duration Curves

One of the most important features of duration analysis is the identification of factors affecting the distribution of the duration times. In most of the available datasets there are various categorical variables (called *factors*) that characterize each project. Examples are the development type, the organization type, business area, language, etc. The values of each factor are called *levels* and they divide the whole dataset into two or more subsets. It is therefore very useful to investigate whether the duration curves of these subsets differ significantly. This is performed either informally, by visual inspection of the duration curves or formally by performing a statistical test. There are two types of statistical tests: The *overall tests* which assess the significance of a factor effect and the *pairwise tests* which compare each pair of levels of a factor to find out where there is a significant difference.

For an example of the graphical methods let us consider the projects of the ISBSG7 data with data quality rating A and B and for which we have non-missing values for the duration and the implementation date. The factor “Development type” has three levels: “Enhanced,” “New Development” and “Re-development” and defines three subsets. By setting a hypothetical cutoff date for our study, the 01/01/98, we result in 538 projects, 47 of which

Table 3 Longitudinal recording of projects' duration with successive cutoff dates

	Cut-off: 1/1/93		Cut-off: 1/1/94		Cut-off: 1/1/95		Cut-off: 1/1/96		Cut-off: 1/1/97		Cut-off: 1/1/98		Cut-off: 1/1/99		Cut-off: 1/1/00	
Project	Duration (months)	Status	Duration (months)	Status	Duration (months)	Status	Duration (months)	Status	Duration (months)	Status	Duration (months)	Status	Duration (months)	Status	Duration (months)	Status
1	63	Active	75	Active	78	Completed	78	Completed	78	Completed	78	Completed	78	Completed	78	Completed
2	29	Active	41	Active	42	Completed	42	Completed	42	Completed	42	Completed	42	Completed	42	Completed
3	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed
4	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed	13	Completed
5	15	Active	27	Active	36	Completed	36	Completed	36	Completed	36	Completed	36	Completed	36	Completed
6	1	Completed	1	Completed	1	Completed	1	Completed	1	Completed	1	Completed	1	Completed	1	Completed
7	–	–	2	Active	14	Completed	14	Completed	14	Completed	14	Completed	14	Completed	14	Completed
8	–	–	–	–	12	Active	20	Completed	20	Completed	20	Completed	20	Completed	20	Completed
9	–	–	–	–	7	Completed	7	Completed	7	Completed	7	Completed	7	Completed	7	Completed
10	–	–	–	–	4	Completed	4	Completed	4	Completed	4	Completed	4	Completed	4	Completed
11	–	–	–	–	4	Active	14	Completed	14	Completed	14	Completed	14	Completed	14	Completed
12	–	–	–	–	3	Active	8	Completed	8	Completed	8	Completed	8	Completed	8	Completed
13	–	–	–	–	2	Active	11	Completed	11	Completed	11	Completed	11	Completed	11	Completed
14	–	–	–	–	–	–	–	–	10	Completed	10	Completed	10	Completed	10	Completed
15	–	–	–	–	–	–	–	–	7	Active	19	Active	24	Completed	24	Completed
16	–	–	–	–	–	–	–	–	7	Active	19	Active	24	Completed	24	Completed
17	–	–	–	–	–	–	–	–	1	Active	3	Completed	3	Completed	3	Completed
18	–	–	–	–	–	–	–	–	–	–	–	–	5	Active	14	Completed

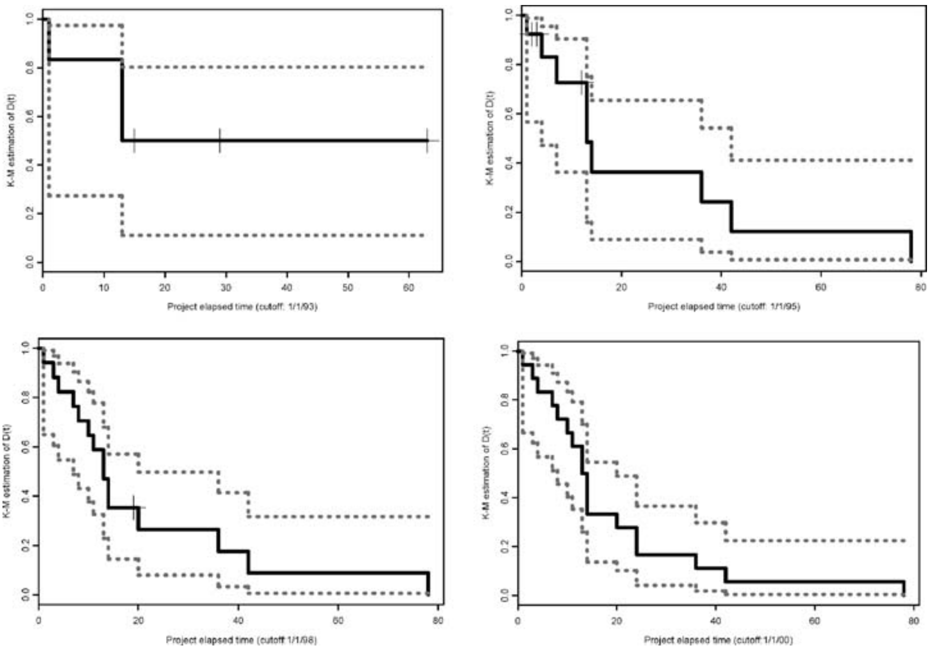


Fig. 13 95% CI for the duration curve on the log–log scale

are incomplete. More specifically, there are 234 enhanced projects with 32 incomplete, 287 new development projects with 13 incomplete and 17 re-development projects with two incomplete projects. The K-M estimations of the corresponding DF and CCCR are plotted in Fig. 14a and b.

In addition to the visual inspection it is also useful to examine the percentiles of the three distributions. These are given in Table 4.

The graphs and the percentiles show that there are differences between the curves. In particular, it seems that the enhancement projects have generally smaller duration while the re-development projects last longer. This is probably due to the fact that enhancement projects are modifications or add-ons that concern a part of a software system, while re-

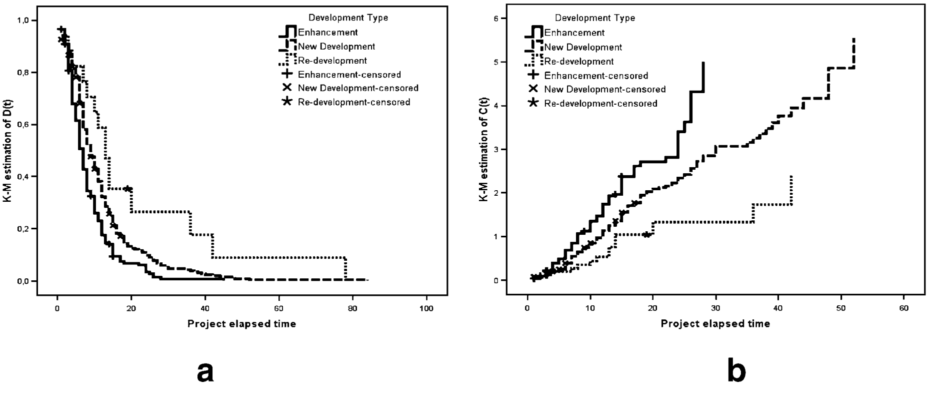


Fig. 14 (**a**) DF and (**b**) CCCR curves of the three development types

Table 4 Estimations of the percentiles of the three distributions

Development Type	25%	Median (50%)	75%
Enhancement	11	7	4
New Development	15	9	6
Re-development	36	13	8
Overall	13	8	5

development projects normally rebuild entire systems. However, there is need for further statistical testing in order to assess if these differences are in fact significant.

There are various statistical tests in the literature. The statistical packages produce automatically certain statistics from the sample and then test their significance based on a theoretical distribution. SPSS provides three different statistics: the *log-rank* (or *Mantel–Cox statistic*), the *modified Wilcoxon* test statistic (or *Breslow statistic*), and the *Tarone and Ware* test statistic. The values of these statistics are compared with values from a chi-square theoretical distribution in order to assess their significance. These statistics can be used either for overall tests or even for pairwise tests. The results of these tests for comparing the three development types of the 538 projects are given in Tables 4 and 5.

From Table 5 we can see that all of the overall tests showed that there is a significant difference between the three levels of the factor ($p<0.0005$) and therefore that the factor development type has a significant effect on the duration. However, in order to locate which of the levels differ significantly we have to perform the pairwise tests of Table 6. The results show that there is clearly a statistically significant difference between the enhancement projects and the other two types ($p<0.0005$ for all tests). Regarding the comparison between new development and re-development projects, two of the tests show significant difference since $p<0.05$, while the Breslow test has $p<0.1$. We can therefore assess that all pairs of the levels have different distributions for their durations.

3.7 Cox Regression Models for Identification of Prognostic Factors

The prediction of the future of a project, which can be called *prognosis*, is important for the project management. Before a manager can make a prognosis and take some critical decisions regarding the resources and the scheduling of a project, historical datasets are often needed. These contain a large number of project attributes, and it is often difficult to identify which ones are most closely related to prognosis. Obviously, an expert can usually decide which of these attributes are irrelevant, but a statistical analysis is very useful in order to prepare a comprehensive report of the relationships that govern the data.

In the previous subsection we discussed techniques for identifying the effects of a single factor on the project duration. However, in the project datasets there are many categorical or continuous variables characterizing the projects. These variables usually interact in complicated ways and their individual effects are confounded in unpredictable manners. For this reason it is preferable to study the combined effect of all the variables on the duration by modeling this relation.

Table 5 Significance of the overall comparisons for the three levels of development type

Statistical test	Significance (p)
Log rank (Mantel–Cox)	0.000
Breslow (Generalized Wilcoxon)	0.000
Tarone–Ware	0.000

Table 6 Significance of all the pairwise comparisons for the three levels of development type

Statistical test	Development type	Enhancement	Development type	
			New development	Re-development
Log rank (Mantel–Cox)	Enhancement	–	0.000	0.000
	New development	0.000	–	0.027
	Re-development	0.000	0.027	–
Breslow (Generalized Wilcoxon)	Enhancement	–	0.000	0.007
	New development	0.000	–	0.082
	Re-development	0.007	0.082	–
Tarone–Ware	Enhancement	–	0.000	0.002
	New development	0.000	–	0.046
	Re-development	0.002	0.046	–

There are various parametric models for modeling the relationship of duration with other variables but they require the knowledge of an underlying theoretical distribution for the error components of the model. This knowledge is often not available, so the common practice is to use a class of comparative, semi-parametric models for duration data. The most known methodology is the *Cox regression analysis* which will be outlined here.

First of all we have to point out the importance of the function c(t). The CCR by definition is an expression of the accumulated knowledge over time, generally known as *aging*. Since CCR is applied only to projects that have stayed active up to a particular time, it accounts for the aging that has taken place in the dataset (see Hosmer and Lemeshow 1999). The point here is that when projects are observed over time, we essentially witness an aging process. It is therefore reasonable to use the CCR for modeling the aging process in a dataset.

Suppose that the CCR is a function depending on a set of predictors $\mathbf{x}=(x_1,\ldots,x_p)$, denoted by $c(t;\mathbf{x})$. The basic assumption for the formation of the model is that any pair of different projects with corresponding values of the predictors (say) $\mathbf{x}_1=(x_{11},\ldots,x_{1p})$ and $\mathbf{x}_2=(x_{21},\ldots,x_{2p})$ has proportional CCRs, i.e. the *completion ratio (CR)*

$$CR(t;\mathbf{x}_1,\mathbf{x}_2)=\frac{c(t;\mathbf{x_1})}{c(t;\mathbf{x_2})} \tag{28}$$

is a constant in the sense that it does not vary with time. The meaning of such an assumption is that the ratio of the conditional completion rate of two projects is the same no matter how long their duration is.

The constant CR property leads to a model where the CCR given the set of predictors for a project is expressed as a product of two functions:

$$c(t;\mathbf{x})=c_0(t)g(\mathbf{x}) \tag{29}$$

The function $c_0(t)$ characterizes how the CCR changes as function of only the duration. The other function, $g(\mathbf{x})$ characterizes how the CCR changes as the function of project predictors and represents their effect. Note that $c_0(t)$ is called *baseline CCR* and represents the CCR when $g(\mathbf{x})=1$. In order to interpret the $c_0(t)$ as the CCR when we remove the effect of all the predictors, a usual choice of $g(\mathbf{x})$ requires that $g(\mathbf{0})=1$.

The *Cox model* assumes that $g(\mathbf{x})$ is an exponential function of the predictors $\mathbf{x}=(x_1,...x_p)$, i.e. an expression of the form:

$$c(t;\mathbf{x}) = c_0(t)\exp\left(\sum_{i=1}^{p} b_i x_i\right) \tag{30}$$

where by x_i, $i=1,...,p$ we denote the *i*th predictor (continuous or categorical) and by b_i the corresponding unknown regression coefficient. Here, the baseline CCR corresponds to a situation where all predictor variables are equal to zero.

An interesting aspect of the Cox regression model is that it can be represented in terms of either the CCR or the DF. It can be proved that the equivalent duration model can be written as:

$$D(t;\mathbf{x}) = [D_0(t)]^{g(\mathbf{x})} \tag{31}$$

where $D_0(t)$ is the baseline duration function and $g(\mathbf{x}) = \exp\left(\sum_{i=1}^{p} b_i x_i\right)$. The regression coefficients and the baseline function are estimated from the data by a version of the maximum likelihood method using the partial likelihood function (Lee and Wang 2003; Hosmer and Lemeshow 1999).

Special attention should be paid on the basic assumption of the Cox model, that the ratio $c(t;\mathbf{x})/c_0(t)$ is not a function of time. The validity of such a hypothesis is tested by statistical tests. However, if the assumption is violated, an extended version of the Cox regression model can be used so as to include time-dependent predictors (Lee and Wang 2003). Available statistical packages (like SPSS and S-plus) provide procedures with a wide variety of options in order to build and validate an efficient model. One of the most important features is the execution of algorithms (e.g. stepwise regression) which are able to produce models containing the most important variables. This is very useful for the cases where the predictors are correlated.

As an illustrative example, we consider from the ISBSG dataset 121 A and B quality projects which all started before the cutoff date 01/01/98 and for which there is no missing data in the predictor variables we used. There are 106 complete and 15 incomplete projects. The predictors and their values we used are given in Table 6. Specifically, there is one continuous variable, the logarithm of the function points and four categorical variables or factors with two or three levels each. For the factor "business area type" we considered only two levels, the insurance and the banking projects. Since the values of the function points can take very large values, the Cox models having this variable as an exponent, estimate its coefficient with values very close to zero. For this reason we use the logarithm of the function points which gives generally better models.

In order to include in the Eq. (29) the categorical variables, there is a need to use binary or "dummy" variables taking only values 0 and 1. When the levels of a factor are only two, these can be represented by 0 and 1. In general, when we have *p* levels in a factor, we need $p-1$ dummy variables to represent them in the form (1,0,...,0), (0,1,...,0), ...(0,0,...,0). The values of dummy variables for our example are given also in Table 7. These are useful for interpreting the regression coefficients. In particular, the level represented by zeros will be used as the *reference level* of this factor in the sense that the regression coefficients will represent the changes of the duration with respect to that level. Note that incomplete projects are used to compute the baseline CCR but not for the computation of the regression coefficients.

Table 7 Predictors and their codings for cox regression

Covariate ln(function points) Factors	Continuous Levels	Frequency	Dummy (1)	Dummy (2)
Business area type	Insurance	21	1	
	Banking	100	0	
Development type	Enhancement	44	1	
	New Development	77	0	
Development platform	PC	10	1	0
	MR	21	0	1
	MF	90	0	0
Language type	ApG	8	1	0
	4GL	43	0	1
	3GL	70	0	0

For the building of the model we used a stepwise Cox regression, as implemented in the statistical program SPSS, which is capable to identify only the most significant predictors for the Cox model. The algorithm resulted in a model with two significant variables: the logarithm of function points and the business area type. The other variables were not considered significant for the model. Here we must emphasize that this does not necessarily mean that the variables which are not included in the model are uncorrelated with duration. The meaning of their exclusion is that their contribution for the model containing the two included variables is insignificant. This may be a result of the correlation among the predictors.

The final model is:

$$c(t;\mathbf{x}) = c_0(t)\exp\left(-0.838 \times \log\left(\text{function\_points}\right) - 0.897 \times \text{Business\_area\_type}\right) \quad (32)$$

The value $e^{-0.838}$=0.432 means that the CCR is reduced by 100% – (100% × 0.432) = 56.8% each time a unit is added to the logarithm of the function points (or equivalently when the function points are multiplied by a factor equal to $e\approx 2.72$). The value $e^{-0.897}$= 0.408 means that the CCR for an insurance project (represented by 1) is reduced 0.408 times that of a banking project (represented by 0). When a factor has more than two levels, the interpretation of the coefficients is similar, in the sense that the coefficient measures the change from the reference level.

It should be emphasized that the Cox regression does not produce point estimations (single values) of the duration of a project. Its prognosis is a whole distribution of duration values with assigned probabilities. So for any value of the predictors we can get interesting graphs of the DF and the CCCR. For example in Fig. 15 (a) and (b) we can see the DF and the CCCR curves corresponding to the *average project* in the sense that in Eq. (32) we substitute the variables by their mean values. In this way we have the curves for a hypothetical project having log(function points)=5.624, the mean value of the variable, business area type=0.174, which is actually the proportion of the ones (21/121) representing the "insurance" projects.

In Fig. 16a and b we see the curves of the average project, only for the mean of log (function points), separately for each business area type. If we want to see how these curves are affected by the change of the variables, in Fig. 17a and b we have the corresponding graphs at log(function points)=7.624. Note the change in the scale on the vertical axis of the CCCR graph in Fig. 17.

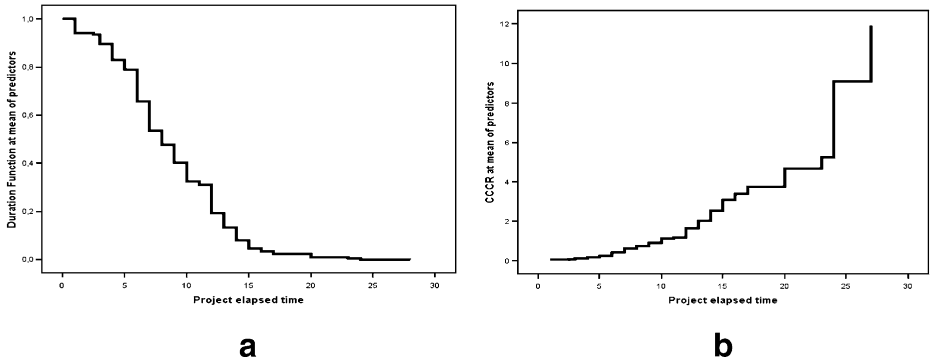


Fig. 15 (**a**) DF and (**b**) CCCR curves at mean of the predictors

A final important issue that has to be examined is the validity of the proportionality assumption, i.e. that that the ratio $c(t;\mathbf{x})/c_0(t)$ is not a function of time. Although there are statistical tests for this, it is much more easy to use graphical methods. Two of these are the graph of the function $\log(-\log(D(t)))$ (*LML curve*) and the plotting of the so-called *partial residuals* against time.

In Fig. 18a we can see the LML curves estimated for the mean of log(function points) for the two levels of the factor. The fact that these curves are parallel indicates that the assumption is valid.

In Fig. 18b there is a scatter plot of the partial residuals of the variable log(function points), i.e. the differences between the observed and expected (given the model is correct) values of the predictor for each case. The partial residuals, and thus the points on the plot, are only produced for uncensored cases. If the proportionality assumption is correct, then there should be no patterns or correlation with time in this plot. In our data we can see that there is no significant correlation (r^2=0.03) and therefore we can accept the assumption.

4 Application

The examples of the previous section were based on projects from a multi-organizational database. In order to investigate the application of the described methods to data from a

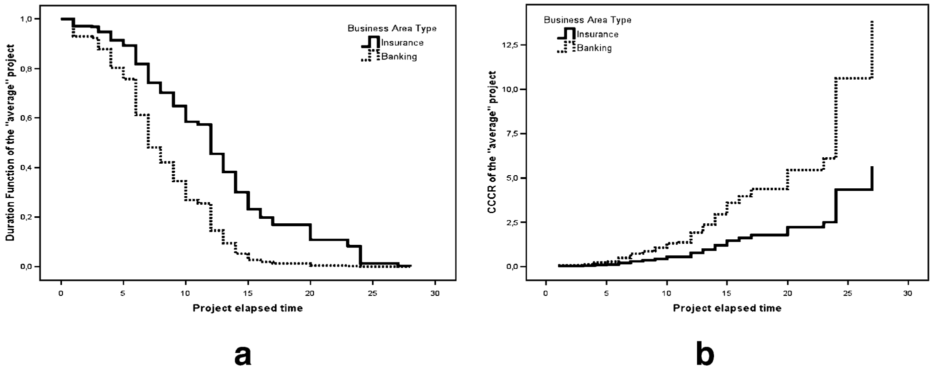


Fig. 16 (**a**) DF and (**b**) CCCR curves at log(function points)=5.624 for both business area types

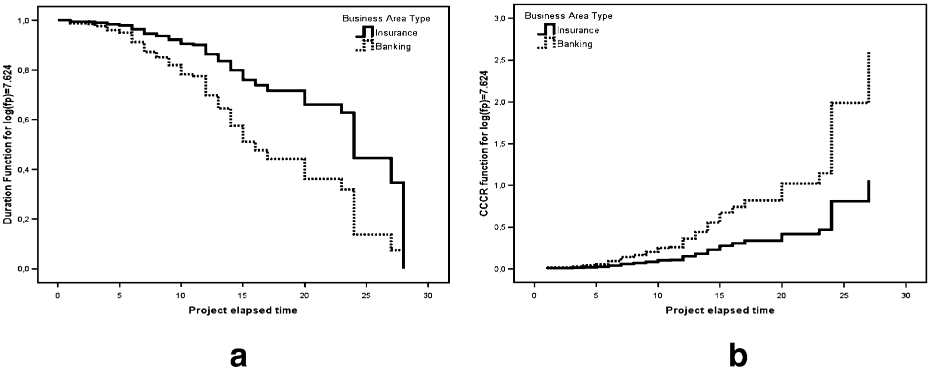


Fig. 17 (a) DF and (b) CCCR curves of the average project where log(function points)=7.624

single organization, we used the dataset published in (Maxwell 2002). The dataset contains completed software projects, so in order to have in our analysis censored observations we defined as final date of the study, a date prior to the actual completion date of several projects. We report the results presenting the K-M estimation of the duration curves, and we construct a Cox regression model describing the relation between the duration and the other predictors in our data set using a forward stepwise procedure. All the results were obtained using the SPSS statistical program.

The dataset contains 62 completed software projects from one of the biggest commercial banks in Finland with starting dates expanding from 10/01/1985 up to 11/01/1993. The duration of each project from specification until delivery is recorded in months (ranging between 6 and 54 months) and so it is easy to compute the termination dates in a range from 10/01/1987 up to 12/01/1992.

In order to simulate the existence of incomplete projects, we pretended that the study stopped at 05/01/1992 (cut-off date). Under this arrangement, six projects were excluded from the analysis as their starting date was after the cutoff date. As a result, the final dataset contained 43 completed and 13 ongoing (censored) projects.

Apart from the duration, there are also 15 categorical variables (factors) characterizing the projects (t01–t15), each with five levels (very low, low, nominal, high, very high). The only numerical variable (covariate) in our analysis is the size. The variables of the data set are described in Table 8.

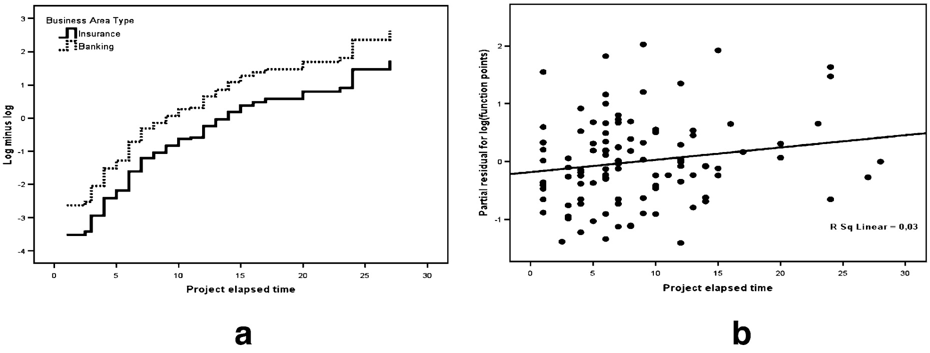


Fig. 18 Graphical tests for the proportionality assumption: (a) LML curves and (b) scatter plots of the partial residuals

Table 8 Variable definition

Variable	Full name	Levels–definition
duration	Duration	Duration of projects from specification until delivery, measured in months.
start	Exact start date	Month/day/year application specification started
size	Application size	Function points measured using the Experience method
t01	Customer participation	1=Very low
t02	Development environment adequacy	2=Low
t03	Staff availability	3=Nominal
t04	Standards use	4=High
t05	Methods use	5=Very high
t06	Tools use	
t07	Software's logical complexity	
t08	Requirements volatility	
t09	Quality requirements	
t10	Efficiency requirements	
t11	Installation requirements	
t12	Staff analysis skills	
t13	Staff application knowledge	
t14	Staff tool skills	
t15	Staff team skills	

Since the number of factors and the number of their levels is very large in relation to the number of projects, we conducted a preliminary analysis aiming at reducing the levels of each factor to two or three categories in order to simplify the data. Specifically, for each factor, one-way ANOVA with post hoc tests (e.g. Tukey's, Bonferoni, etc.) were used with dependent variable the duration in order to identify homogeneous categories and to concatenate them. The new variables resulted from this procedure can be seen in Table 9.

Table 9 The new factors after the concatenation of levels

Original variable	New variable	Levels–values
t01	*t01_3*	1={Very Low, Low}, 2={Nominal}, 3={High, Very High}
t02	*t02_3*	1={Very Low, Low}, 2={Nominal}, 3={High, Very High}
t03	*t03_3*	1={Low}, 2={Nominal}, 3={High, Very High}
t04	*t04_3*	1={Low}, 2={Nominal}, 3={High, Very High}
t05	*t05_3*	1={Very Low, Low}, 2={Nominal}, 3={High, Very High}
t06	*t06_2*	1={Very Low, Low}, 2={Nominal, High}
t07	*t07_3*	1={Very Low, Low}, 2={Nominal}, 3={High, Very High}
t08	*t08_3*	1={Low, Nominal}, 2={High}, 3={Very High}
t09	*t09_3*	1={Low, Nominal}, 2={High}, 3={Very High}
t10	*t10_4*	1={Low}, 2={Nominal}, 3={High}, 4={Very High}
t11	*t11_4*	1={Low}, 2={Nominal}, 3={High}, 4={Very High}
t12	*t12_3*	1={Low, Nominal}, 2={High}, 3={Very High}
t13	*t13_3*	1={Very Low, Low}, 2={Nominal}, 3={High, Very High}
t14	*t14_3*	1={Very Low, Low}, 2={Nominal}, 3={High, Very High}
t15	*t15_2*	1={Very Low, Low, Nominal}, 2={ High, Very High}

In Fig. 19a and b, we can see the corresponding K-M estimations of D(*t*) and *C*(*t*) for this dataset. The median duration is 14 months, which means that half of the projects are expected to finish up to the 14th month. By observing the shape of the curve, we can see that the slope of the curve changes first around the 20th month and then around the 30th month. This may be an indication of the rapid development rate of small projects in contrast to the slower development rate of medium or large projects. These changes in the distribution of duration can be also detected from the CCCR.

As mentioned previously, the curves are used for detecting the effect of various factors on the duration variable. This is a typical preliminary exploration prior to the modeling of such effects with regression models. In our example, from the 15 productivity factors in the dataset, the graphical representation showed that only five of them have different duration distributions between their levels. This was confirmed by the overall statistical tests. The factors are: “customer participation,” “tools use,” “software’s logical complexity,” “requirements volatility” and “staff tool skills.” In Figs. 20, 21, 22, 23, and 24, respectively, we can see the DF curves for the levels of these factors. Table 10 contains the median durations of the levels for each factor that has effect on the duration and the pairwise comparisons which show which of these levels have significant difference in their duration distributions.

The graphical and the statistical results lead us to the following conclusions:

(a) An average (nominal) customer participation in the development work of a project is better than the absence and much better than high participation. This conclusion comes from Fig. 20 where the curve corresponding to level 2 (nominal customer participation) shows smaller duration than the two other curves. This is also noticeable by the median durations in Table 10 (first row) where the median duration of level 2 is 9 months while the other levels 1 and 3 have median durations 13 and 21 correspondingly. The pairwise comparisons (last column) confirm the significant differences between levels 1 and 2 and between 1 and 3.

(b) Using tools has a significant effect on the duration curves, not only in reducing the duration (median duration 13 months instead of 22) but also on the decreasing rate of the curve (Fig. 21, Table 10—second row).

(c) As long as the software’s logical complexity of a project increases, the development duration is extended since the median duration of level 1 is 10 months and significantly different from levels 2 and 3 (Fig. 22, Table 10—third row).

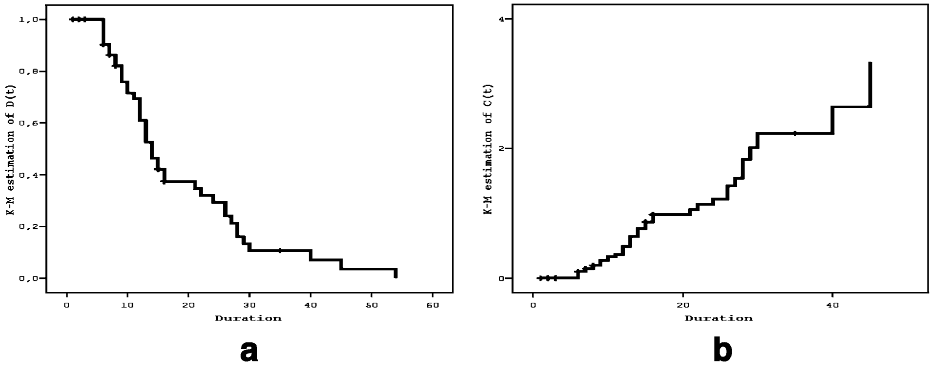


Fig. 19 K-M estimation of the (**a**) DF and (**b**) CCCR

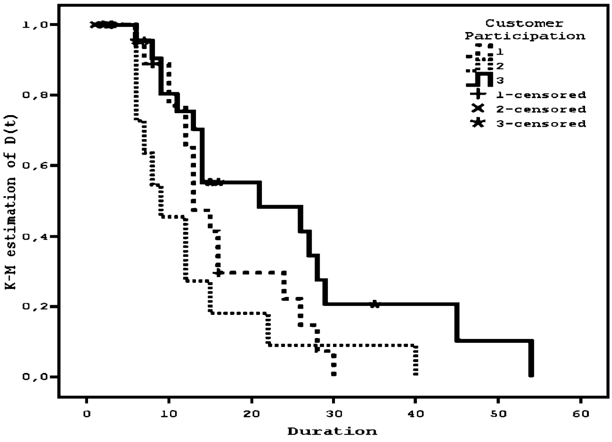


Fig. 20 K-M DF for the three levels of factor "customer participation"

(d) There seems to be an increasing trend of the duration of software projects with the increase of the user requirements volatility as the median duration of level 3 is 26 months and significantly different from levels 1 and 2 (Fig. 23, Table 10—fourth row).

(e) Projects with lower staff tool skills have in general longer duration (median 24 months) than projects with higher staff tool skills (Fig. 24, Table 10—fifth row). Also, another inference comes from the observation that while the staff tool skills are increasing, the differences between the levels are smoothing (the distribution of level 2 has no significant difference to that of level 3 as we can see from Table 10).

Apart from (a) all other inferences seem intuitively correct. The observation that high customer involvement is related to extended duration may be due to customers being heavily involved in large/critical projects or, possibly such involvement leads to requests for continuous changes. Anyway, it is generally considered that customer involvement is beneficial for software projects (consider for example the customer-on-site practice in agile methods) and such a finding deserves further analysis.

In order to construct a valid model describing the relation between the duration (as dependent variable) and the other predictors in our dataset, we used forward stepwise Cox

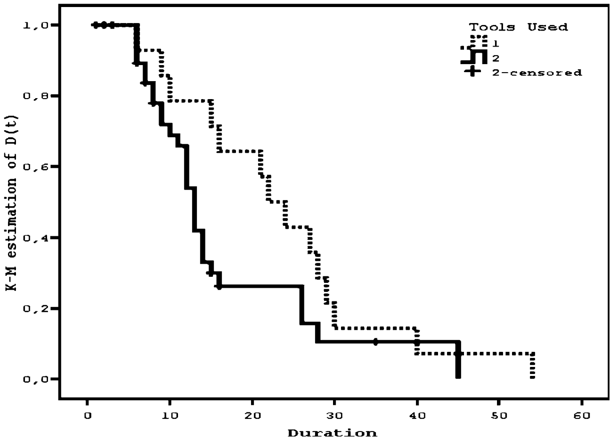
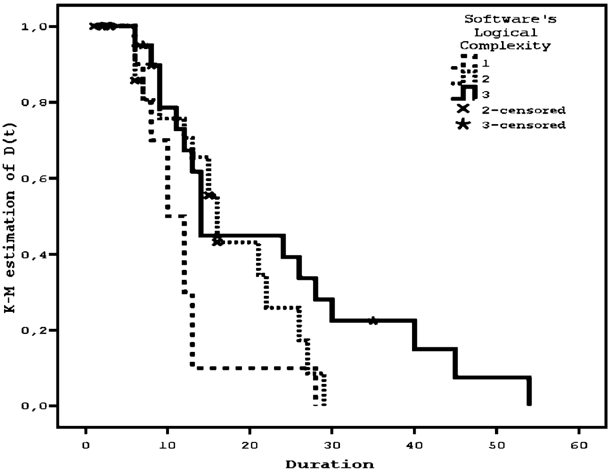


Fig. 21 K-M DF for the two levels of factor "tool used"

Fig. 22 K-M DF for the three levels of factor "software's logical complexity"

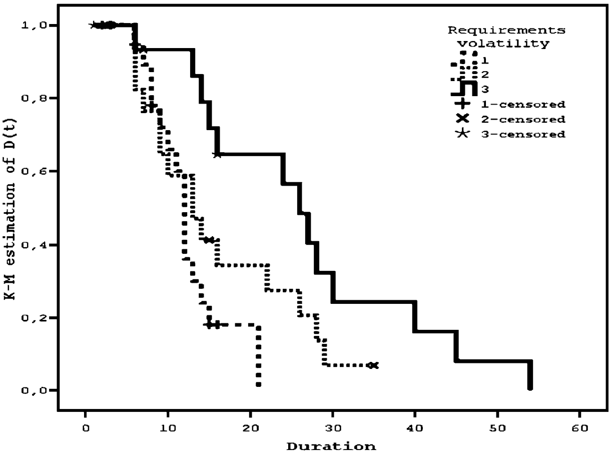


regression, in order to identify the most significant variables for the model and estimate their regression coefficients. As we already mentioned, the procedure may exclude some of the variables that seem to have effect on the duration but they are highly correlated with other predictors so as to result in a compact and flexible model. For the specific dataset, the procedure resulted in a model with predictor variables "requirements volatility" (t08_3) and the logarithm of "application size," log(size). The model is:

$$c(t;\mathbf{x}) = c_0(t)\exp\left(-0.653\times\log(\text{size}) + 1.656\times t08\_3(1) + 0.885\times t08\_3(2)\right) \quad (33)$$

Note that the three levels of factor t08_3, are represented by two binary variables denoted t08_3(1) and t08_3(2) which take the values (1, 0) for level 1, (0, 1) for level 2 and (0, 0) for the reference level 3. Therefore, the coefficients of the binary variables represent the amount of change in CCR with respect to the reference level 3. The DF curve at the mean values of the predictors are displayed in Fig. 25a. In Fig. 25b we see the DF curves for all the levels of the factor "requirements volatility" as produced for the average value of log(size) which is equal to 6.083. The results indicate that for an average value of application size, we have an increasing trend of the duration as requirements volatility increases.

Fig. 23 K-M DF for the three levels of factor "requirements volatility"



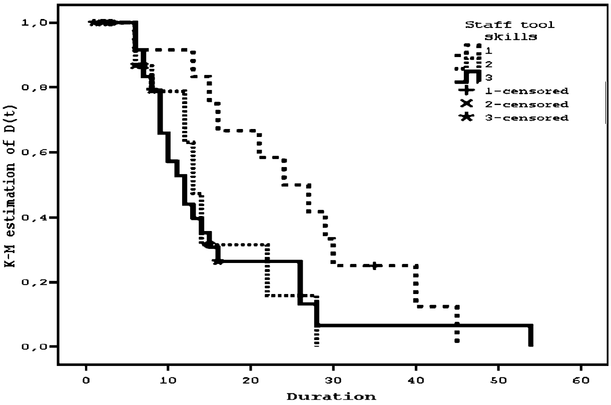


Fig. 24 K-M DF for the three levels of factor “staff tool skills”

As we mentioned in the description of the Cox model, it is important to assess the proportionality assumption graphically. In Fig. 26a we can see the LML curves for the three levels of “requirements volatility” which are parallel while in Fig. 26b we can see that the partial residuals are uncorrelated with the duration ($r^2 = 9.487 \times 10^{-5}$). These graphs show that the proportionality assumption is not violated.

5 Discussion

5.1 Situations Where PDA can be Useful

In this subsection we briefly review estimation situations in which duration analysis can be particularly useful. In all cases it is presumed that a set of software related activities (entire

Table 10 Median durations and pairwise comparisons for the levels of each factor

Factor	Description	Levels (after concatenation)	Median duration (in months)	Significant differences from pairwise comparisons
Customer participation	The customer participation in the development work	1	13	1–2
		2	9	2–3
		3	21	
Tools used	The use of tools during the development face of a software project	1	22	1–2
		2	13	
Software’s logical complexity	The logical complexity of software (computing, I/O needs and user interface requirements)	1	10	1–2
		2	16	1–3
		3	14	
Requirements volatility	The volatility of customer–user requirements during the development face of project	1	12	1–3
		2	13	2–3
		3	26	
Staff tool skills	The experience level of project team (supplier and customer) with development and documentation tools at project kick-off	1	24	1–2
		2	13	1–3
		3	12	

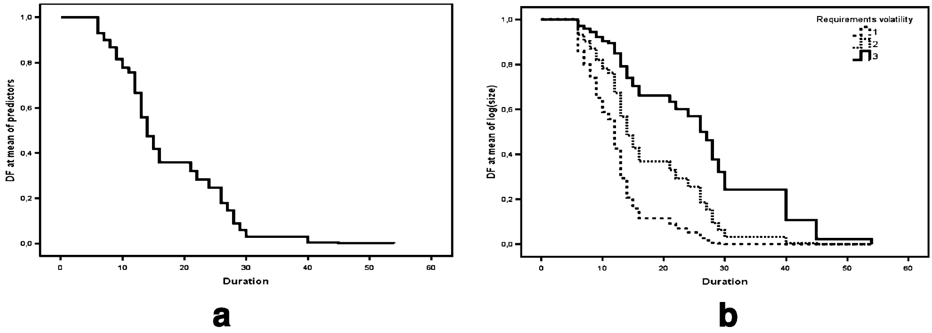


Fig. 25 DF curves (**a**) at mean of the predictors and (**b**) for the levels of "requirements volatility"

projects, programming or testing sessions, etc.) is available, characterized by a number of attributes/factors. For some of them duration is known, while for others duration is wanted to be estimated. Such data sets differ from each other in the type of activities they contain, the number and type of characterizing factors, the number of observations etc.

1. Duration estimation. In this typical case, the analyst wants to estimate the duration of a forthcoming project, based on historical data. PDA, in particular the probability density function, can prove useful when there are too few completed projects in the available data to support a statistically valid analysis. This may be a typical situation in a start-up company or a company that has just initiated a measurement program, with few measured completed projects but with a number of interesting on-going projects. A similar estimation situation appears when the analyst possesses a large number of completed projects but for some reason only few of them are really interesting. Such a case may be for example when many completed projects are too old to be trusted. Or they are too different from the project in hand, because they differ in critical factors, such as control / real-time systems versus data intensive systems. Multi-organizational data sets tend to produce such problems (see for example studies on ISBSG (ISBSG 2005) where data quality level is used to filter data points for subsequent analysis (Angelis et al. 2001). PDA helps in considering on-going projects as well and may lead to valid results.

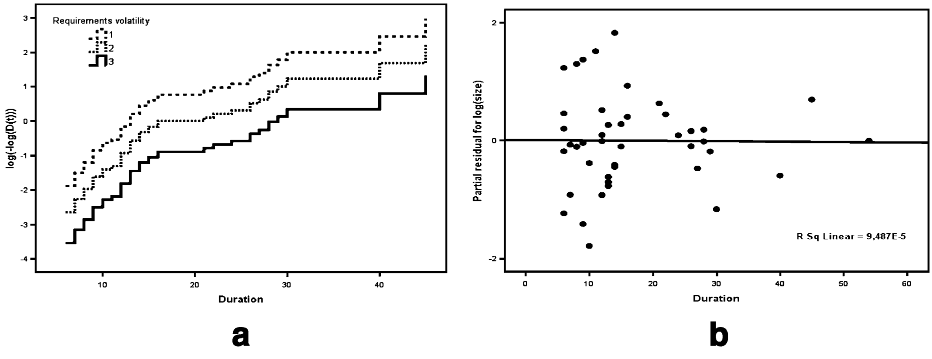


Fig. 26 Graphical tests for the proportionality assumption: (**a**) LML curves and (**b**) scatter plots of the partial residuals

2. Project monitoring and assessment. PDA tools can be helpful during the project execution. E.g. using the duration function, the manager / analyst may estimate the probability that the duration of the *project* or a critical activity, such as testing, is longer than the time remaining for the scheduled finish date. In this way, PDA produces probability figures which are essential for quantitative risk analysis.
3. Project cancellation. It is reasonable to consider cancellation a negative outcome for a software project. However, Boehm (Boehm 2000) has stated that project cancellation is not always a bad thing, software managers are not to be blamed in any case, and may be seen as a natural project development in certain cases. The project manager / analyst through the use of the conditional completion rate may estimate the probability that the project terminates in a very short time interval, understanding in this way whether his / her project is similar to other projects that may have been abruptly terminated in the past. Typically, such information should be expected to be found in a single organization's project data base. Overall, the project manager/*analyst* will be assisted in determining the probability that a delayed project has to (a) catch up with its baseline, (b) suffer additional delay but come to a successful end, (c) suffer additional delay and eventually be cancelled, and (d) by cancelled in a short time period. Such information is particularly critical for project portfolios that include high-risk projects.
4. Software Industry Trend Analysis. The availability of multi-organizational project data provides the opportunity for trend analyses that may reveal such tendencies as the productivity, duration, quality of software projects in specific business areas (e.g. banking, public sector, utilities) during specific time periods. Analyses of this kind may provide useful insight concerning the impact of a new technology (e.g. web information systems in e-banking) or help understand the reaction of software organizations in the presence of a special event (e.g. Year 2K). PDA tools will provide the means for introducing cut-off days to set the start and finish dates of the trend analysis
5. Open Source Software Project Analysis. Apart from analyzing and monitoring conventional, 'closed source' projects, there is a growing interest in observing open source system (OSS) development. Researchers are continuously studying the evolution of known open software systems and analyze demographic data from internet sites that massively support OSS projects. A large private company or a public sector organization, slow in migrating to new software products, may want to know the status and the perspectives of an OSS application before adopting it. A software organization would like to know the chances that one of their closed source project would have if it were turned to an OSS project. In all of the above cases, decisions may be made based on the wealth of the available information that is available for OSS projects. However, while it is possible to define a start date for such projects, it is normally hard to determine a termination date, because OSS projects are voluntarily, ever-going projects, not subject to contract agreements. In this sense most of the interesting OSS projects have to be considered on-going projects and this is where PDA comes into play. Recently (Koru et al. 2007) survival analysis was applied to OSS debugging related data, providing clear relationship between design coupling and number of bugs. We expect that in the following years more similar works will appear in the literature.

We wish to emphasize the fact that in all of the above situations, PDA provides the software manager with quantitative information that must be integrated with his / her personal intuition and experience to make the final decision.

5.2 Validity of Project Duration Analysis as a Statistical Analysis Tool

In this subsection we consider some reservations and issues that may be raised regarding the underlying assumptions and the applicability of the method on software management data.

1. Ambiguity of the notion of duration.

It may be argued that the notion of project duration is ambiguous in software project development due to the fact that the starting and terminating events for a project are not well-defined. Under such an argument, the validity of PDA is questionable.

In this regard, we have to emphasize that survival analysis is a statistical method applied to time variables concerning duration of any process. The application of the method requires a dataset with individual cases and a variable representing duration. This variable (elapsed time or duration) is essentially a measure recorded in all of the known software project datasets (COCOMO, ISBSG, etc) along with the effort variable. The definition given for duration by ISBSG is: “elapsed time indicating the total duration in months required to carry out and complete the software project.” Similar definitions of duration are provided in other datasets as well.

So the real argue does not concern the validity of the PDA methodology itself but rather the ambiguity of the measurements recorded as duration or elapsed time. However, PDA methodology, as any other statistical methodology (e.g. regression or ANOVA) assumes that the recorded data are precise and valid. Of course the same argue on the ambiguity of the duration measurements can be stated for the effort and or even for the size of a project. It is clear that no statistical method can be applied to erroneous measurements and as far as we know there is no methodology or model in the related literature taking into account the possibility of flawed data. Finally, it is worth mentioning that the general survival analysis has been applied in much more ambiguous events, such as the treatment of patients (“cure” is not always easily defined).

2. Overlooking of delays due to human and other factors

Another point that may be raised against the PDA methodology is that it fails to take into account delays due to project abandonment by members of the development or the design team, complexity, change in requirements and other unpredictable factors.

However, PDA does not ignore the existence of all these factors. What PDA (as any other statistical method) assumes is that the duration is a random variable and that its randomness and its variability depend on various factors (such as the leave of a chief programmer). It is clear that PDA operates on available datasets. What PDA does is that it takes the duration variable recorded and also the other factors recorded (project characteristics) and tries to explain a high percentage of the variability by building models on the available factors. Yet, it is impossible to explain 100% of the variability. So the variability that is not explained by the recorded factors can be due to unpredictable and unrecorded situations, such as the ones mentioned above. The models built by PDA or any other method are in a great degree explorative in the sense that they try to explain the effect of various factors on duration but also predictive. It is important therefore to realize that the statistical methods and models are tools for understanding variability and helping managers’ decisions.

Note also that in situations where we have some information that an unpredictable unmeasured factor (usually human) causes a large heterogeneity in the distribution of duration, there are special models, called *frailty models* for handling this (see Hosmer and Lemeshow 1999, p.317–326).

3. Dependency of the projects

As we mentioned in the description of the Kaplan-Meier method, a basic assumption for its application (and the application of other PDA techniques) is that the durations of the projects are independent in the sense that the duration of any project cannot affect the duration of any other project. It may be argued that this assumption is very often violated in datasets of the same organization which share common resources. In large, cross-organizational datasets, such as the ISBSG dataset, this phenomenon is diluted, since projects come from different companies around the world. Of course, the same can be stated for the effort or generally the cost of the projects. However, the fact is that the bibliography on cost estimation includes a plethora of statistical methods and models (regression, ANOVA, etc) whose basic assumption is the independence of projects. In general, we need to choose carefully the data for our models and remove as much as possible the effect of any dependence. In case we know the nature of dependence (although this is generally very difficult), it is possible to include special variables in our models for taking into account the dependence. An example is the time-dependent covariates in Cox regression models.

4. The constrained nature of duration

Another point of concern is the fact that duration is usually constrained by customer requirements, varying team size and common resources shared by many projects of an organization. Moreover, projects from different organizations are not homogeneous regarding their distribution.

Indeed, it is almost impossible to find a scientific area where the sense of duration can be considered as infinite in practice. One way or another, all notions of life are constrained. However, this fact does not affect the general approach. Of course, some of the parametric distributions we use for modeling duration (like the lognormal) have their right tail tending to infinity, but this is just an approximation which assigns practically zero probabilities to large unrealistic values. In this regard, the nonparametric methods have an interpretative advantage. The constraints caused by the interaction of projects are essentially related to the dependency of the projects and this problem was discussed in the previous point.

It is also a fact that the duration, as a variable in a dataset, can have a large heterogeneity due to the projects that come from various sources. This of course can be a problem for the parametric methods which assume a certain density function for all the projects, but the nonparametric ones are not constrained by such an assumption. However, if we wish to find some projects which are outliers, i.e. they differ significantly from the others in a database and cause problems in the modeling of duration, we can detect them through the appropriate statistical analysis. It is also important to note that all these possible constraints may appear in any other dependent variable, such as the effort or the productivity.

5. Removing of censored observations and application of traditional methods

Since the notion of censored observation is essential for the duration analysis, a reasonable question is whether it is meaningful to remove them from the analysis and moreover, to apply an ordinary model, such as the least squares regression.

First of all, we must emphasize that the occurrence of censored values is a unique characteristic of variables related to time and specifically duration. By ignoring those values from our analysis we lose valuable information and we cause bias in our estimation. Indeed, the projects with long duration are most likely to be censored, so by removing them, all the information regarding large projects may be lost leading to reduced representation of an important part of the population. It is hence essential to take the censored observations into account and to use them as well as the uncensored observations.

Regarding the traditional methods such as regression analysis and ANOVA, these can model the values of duration (point estimations) by ignoring the incomplete projects. On the contrary, PDA models the probability of the duration to belong in an open interval and for this it can use the incomplete durations as well. Due to the completely different approach taken, the methods are not directly comparable. Of course in statistics there is no "best for all data" model. So, nothing can guarantee that for a specific dataset, PDA is better than regression or vise versa, since the notion of "better" is meaningless. However, since time variables are very special cases of random variables and since there are so many techniques for analyzing them under the general framework of survival analysis, we believe that it is a good idea to take into account their nature and exploit all the available information.

6 Conclusions

In this paper we have proposed a statistical framework for studying the distribution of the duration of software projects and the factors that affect it. We suggested the use of a generic class of statistical methods known as survival analysis. The main feature of the method is the construction of probabilistic models based on data of complete and ongoing projects. Under the general framework, we can model the dependence of duration on various independent variables. We described the basic principles and notions of the method, we applied it on real data and we provided a discussion regarding several potential applications of the method.

Future research includes the use of PDA to establish which projects should be considered 'recent' or 'modern' ones, i.e. how can a software organization separate projects to 'old' projects (with potentially misleading productivity or duration values) and 'recent' ones, to be used in estimations of future projects. Such analysis may be implemented by applying repeatedly PDA on progressively smaller historical data subsets, excluding each time the older projects from the examined data set.

Acknowledgement The authors wish to thank the editor and the anonymous reviewers for their comments which helped in improving the paper.

References

Angelis L, Sentas P (2005) Duration analysis of software projects. Proceedings of the 10th Panhellenic Conference on Informatics, pp 258–269

Angelis L, Stamelos I, Morisio M (2001) Building a software cost estimation model based on categorical data. Proceedings of the 7th IEEE International Software Metrics Symposium, pp 4–15

Ayal M (2004) The effect of scope changes on project duration extensions. PhD Dissertation, Faculty of Management, Tel Aviv University

Barry EJ, Mukhopadhyay T, Slaughter SA (2002) Software project duration and effort: an empirical study. Information Technology and Management 3(1–2):113–136

Boehm BW (1981) Software engineering economics. Prentice-Hall, Upper Saddle River, NJ

Boehm B (2000) Project termination doesn't equal project failure. Computer 33(9):94–96

Boehm BW, Horowitz E, Madachy R, Reifer D (2000) Software cost estimation with COCOMO II. Prentice-Hall, Englewood Cliffs, NJ

Clark B, Chulani S, Boehm B (1998) Calibrating the COCOMO II post-architecture model. Proceedings of the 20th International Conference on Software Engineering, pp 477–480

Collet D (1994) Modelling survival data in medical research. Chapman & Hall, London

Hosmer DW, Lemeshow S (1999) Regression modelling of time to event data. Wiley, New York

ISBSG Data Disk (2005) Release 7 (http://www.isbsg.org.au)

Jain G (2002) Reducing the software project duration using global software development. Master thesis. Dept of Computer Science and Engineering, Indian Institute of Technology, Kanpur

Kaplan EL, Meier P (1958) Nonparametric estimation for incomplete observations. J Am Stat Assoc 53:457–481

Kitchenham B (1998) A procedure for analysing unbalanced datasets. IEEE Trans Softw Eng 24:278–301

Lee ET, Wang JW (2003) Statistical methods for survival data analysis, 3rd edn. Wiley, New York

Lind MR, Sulek JM (2000) A methodology for forecasting knowledge work projects. Comput Oper Res 27:1153–1169

Koru AG, Zhang D, Liu H (2007) Effect of coupling on defect proneness in evolutionary open-source software development. Proceedings of the IFIP 3rd International Conference on Open Source Systems (OSS) 2007, Springer, 271–276

Maxwell K (2002) Applied statistics for software managers. Prentice-Hall, Upper Saddle River, NJ

Maxwell K, Briand L, Emam K, Surmann D, Wieczorek I (2000) An assessment and comparison of common software cost estimation modelling techniques. Proceedings of the 22nd International Conference on Software Engineering, ICSE, 377–386

Mizuno O, Adachi T, Kikuno T, Takagi Y (2001) On prediction of cost and duration for risky software projects based on risk questionnaire. Proceedings of the Second Asia-Pacific Conference on Quality Software, pp 120–128

Oligny S, Bourque P, Abran A, Fournier B (2000) Exploring the relation between effort and duration in software engineering projects. Proceedings of the World Computer Congress, pp 175–178

Parmar MKB, Machin D (1995) Survival analysis. A practical approach. Wiley, Chichester

Putnam LH, Myers W (2003) Five core metrics: the intelligence behind successful software management. Dorset House, New York

Rainer A, Hall T (2003) A quantitative and qualitative analysis of factors affecting software processes. J Syst Softw 66(1):7–21

Rainer A, Hall T (2004) Identifying the causes of poor progress in software projects. Proceedings of the 10th International Symposium on Software Metrics, pp 184–195

Rainer A, Shepperd MJ (1999) Re-planning for a successful project schedule. Proceedings of the 5th IEEE International Software Metrics Symposium, pp 72–81

Sentas P, Angelis L (2005) Survival analysis for the duration of software projects. Proceedings of the 11th IEEE International Software Metrics Symposium

Shepperd MJ, Schofield C (1997) Estimating software project effort using analogies. IEEE Trans Softw Eng 23:736–743

Venables WN, Ripley BD (2002) Modern applied statistics with S. Springer, New York

Panagiotis Sentas received his BSc in Computer Science from Aristotle University of Thessaloniki (A.U.Th.). He is currently a Ph.D. student at the Department of Informatics of A.U.Th.. His research is focused on software cost estimation with statistical methods.

Lefteris Angelis received his BSc and Ph.D. degree in Mathematics from Aristotle University of Thessaloniki (A.U.Th.). He is currently an Assistant Professor at the Department of Informatics of A.U.Th.. His research interests involve statistical methods with applications in information systems and software engineering, computational methods in mathematics and statistics, planning of experiments and simulation techniques.

Ioannis G. Stamelos received a degree in Electrical Engineering from the Polytechnic School of Thessaloniki (1983) and a Ph.D. degree in Computer Science from the Aristotle University of Thessaloniki (1988). He is an Assistant Professor at the Dept. of Informatics, Aristotle University of Thessaloniki, and a Teaching Consultant at the Hellenic Open University. His research interests include empirical software evaluation and management, software education, agile methods and open source software engineering. He is author of approx. 80 scientific papers and has coedited three books.