Automatic Identification of Change Points for the System Testing Process

João W. Cangussu
Department of Computer Science
University of Texas at Dallas
Richardson-TX 75083-0688, USA
cangussu@utdallas.edu

Michael Baron
Department of Mathematical Sciences
University of Texas at Dallas
Richardson-TX 75083-0688, USA
mbaron@utdallas.edu

Abstract

The application of the `CDM Model` *has proven to be effective and accurate. The piecewise approximation used relies on the proper identification of re-calibration points. These points are identified based on the observation of the testing process. However, not all change points are easily identifiable. In this paper we present a new technique for the automatic identification of change points which increases the applicability and accuracy of the* `CDM Model`*. In addition, identification of change points can also be used to increase the understanding of the general behavior of a testing process by providing an expected signature for future projects.*

1 Introduction

The state variable model of the software testing process, hereafter referred to as the `CDM Model`[1], has been successfully applied to a series of industrial projects. The application of the `CDM Model` is done, in most of the cases, using a piecewise approach. That is, the model is {re}calibrated when a "visible" change in the testing process is observed. For example, if the number of testers changes, the model is adjusted according to the changes; also, if a new testing tool is applied, a recalibration is done to account for the advantages of using the tool. However, not all changes in the testing process are easily identified as the ones just mentioned. When a new part of the system that has a higher defect density, starts to be tested the behavior of the decay of defects will suffer some changes that cannot be immediately identified by just "looking" at the data. These changes in the behavior can be statistically identified and are known as change points [5, 11].

[1] CDM stands for *C*angussu, *D*eCarlo, and *M*athur, the authors of the model.

In this paper we describe a technique to identify change points for an ongoing testing process. This allows for the automatic identification of re-calibration points (previously done manually) for the `CDM Model` which simplifies the use of the model as well as increases its accuracy. The proper identification of change points can also be used to determine a signature for the testing process of future projects allowing the estimation of the time of occurrence of specific change points.

In the remainder of this paper we first present a brief description of the `CDM Model` in Section 2. Section 3 presents a discussion about the need for an automatic change point detection mechanism. The nature of changes points is discussed in Section 4 followed by the description of the technique used to identify them in Section 5. The application of the approach proposed here to a testing process is the subject of Section 6. Concluding remarks are presented in Section 8 along with a brief description of the potential use of change points to capture a signature for a testing process.

2 The `CDM Model`

A linear deterministic model of the system test phase of the Software Testing Process (STP) is based upon three assumptions. The assumptions are based on an analogy of the STP with the physical process of a spring-mass-dashpot system and also in Volterra's predator-prey model [15]. A description and justification of this analogy and the choice of a linear model are given elsewhere [6]. The model has been validated using sets of data from testing projects and also by means of an extremal case and a sensitivity analysis [7]. The assumptions and the corresponding equations follow.

Assumption 1: *The rate at which the velocity of the remaining errors changes is directly proportional to the net applied effort (e_n) and inversely proportional to the complexity (s_c)*

of the program under test, i.e.,

$$\ddot{r}(t) = \frac{e_n(t)}{s_c} \quad \Rightarrow \quad e_n(t) = \ddot{r}(t)\ s_c \tag{1}$$

Assumption 2: *The effective test effort (e_f) is proportional to the product of the applied work force (w_f) and the number of remaining errors (r), i.e., for an appropriate $\zeta(s_c)$,*

$$e_f(t) = \zeta(s_c)\ w_f\ r(t) \tag{2}$$

where $\zeta(s_c) = \dfrac{\zeta}{{s_c}^b}$ *is a function of software complexity. Parameter b depends on the characteristics of the product under test.*

Assumption 3: *The error reduction resistance (e_r) opposes and is proportional to the error reduction velocity ($\dot{r}$), and is inversely proportional to the overall quality (γ) of the test phase, i.e., for an appropriate constant ξ,*

$$e_r(t) = -\xi\ \frac{1}{\gamma}\ \dot{r} \tag{3}$$

Combining Eqs. 1, 2, and 3 in a force balance equation and organizing it in a State Variable format ($\dot{x} = Ax + Bu$) [15] produces the following system of equations.

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-\zeta\ w_f}{s_c^{(1+b)}} & \frac{-\xi}{\gamma\ s_c} \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{s_c} \end{bmatrix} F_d \tag{4}$$

F_d above is included in the model to account for unforeseen disturbances such as hardware failures, personnel illness, team dynamics, economic or any event that slows down or even interrupts the continuation of the test process.

Along with the model in Eq. 4, an algorithm, based on system identification techniques [13], has been developed to calibrate the parameters of the model [6]. The fast convergence presented by the algorithm increases the model applicability and accuracy. Finally, a parametric control procedure is used to compute required changes in the model in order to converge to desired results according to time constraints [6]. The `CDM Model` has been applied to data from actual large industrial projects with encouraging results [6].

3 Need for the Automatic Identification of Change Points

One of the major goals of the feedback control approach based on the `CDM Model` is the development of a tool to help managers predict and control their testing process. The tool must be user friendly and the process must be as automated as possible. As stated earlier, the {re}calibration of the model is done when certain conditions that are either known (personnel re-allocation) or easily observable occur. However, even the easily observable conditions are, in general, noticed only many steps after the fact. Even in this case, users are not interested in finding out when to re-calibrate the model, they just want to up-load data and collect the estimations. The simpler the tool the better. In this case an automatic identification of change points can have a high impact on the applicability of the approach. In addition, the accuracy of the approach can be further improved with an automatic identification of change points that are not necessarily visible by naked eyes. The nature of some change points of interest is discussed next in Section 4.

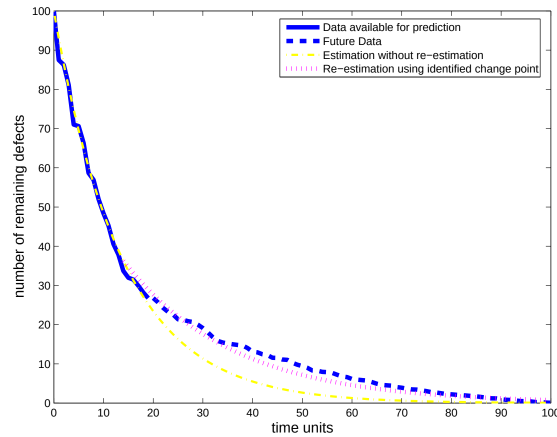

Figure 1. Prediction using the `CDM Model` with and without the use of the identification of a change point.

Assume a scenario where 20 data points are available as represented by the solid line in Figure 1. Also assume that no change has been made in the testing process. The dash-dotted line in Figure 1 represents prediction when the 20 data points are used to calibrate the model. As it can be seen, the prediction differs from what would be the rest of the data. A change in the rate of the decay of defects has been introduced at time t=14. If the model is re-calibrated at that change point, the accuracy of the prediction significantly increases as seen in Figure 1. If we assume the goal is to achieve 90% defect reduction, the estimation without the identification of the change point results in an expected completion time of 33 time units. When the recalibration is done, the completion time is estimated to be 45 time units. The completion time for the actual data, based on the established goal, is 49 time units. Therefore, in this case, the proper identification of the change leads to a 25% increase of accuracy of estimation which clearly justifies its use.

4 Nature of Change Points

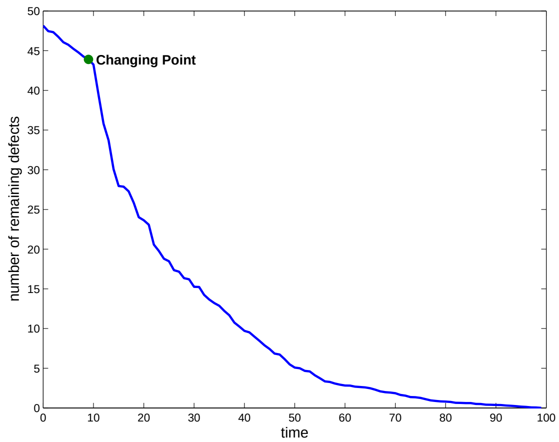

Figure 2. Change point associated with an initial linear decay during system testing.

Many different types of events can affect the behavior of the decay of defects during the testing process. For example, a three hours' meeting with all the testers would reduce their productivity for that day and impact the behavior of the process. Another example could be that a major defect has been found and testing has to temporarily stop while waiting for the defect to be fixed. Also, testers may be more productive on days preceding an evaluation. In summary, these events can be considered as noise in the testing process and the consideration of all possible events would lead to the identification of change points in almost every day of the process. This is clearly not a desired solution since they do not change the overall behavior of the process. Our interest is on two specific events that, though have a major impact on the behavior of the process, are not easily/immediately identifiable.

The beginning of a system test phase is sometimes disturbed by a sequence of defects that prevent the progress of testing. Many of them are associated with the installation of the product on a different testing environment. Also, when testing a new product, the team may need additional time to learn the specifics about the software. The combination of these two events leads to an initial almost linear small decrease of the number of defects followed by an exponential decay. This change in behavior, as seen in Figure 2, is one of the change points of concern here.

One could argue that the change point described above can be easily identified by just looking at the data. Figure 2 shows the number of defects per time unit for the data from Figure 3. A naive approach would be to consider the first large difference between two consecutive points as a change point. However, this would lead to many false positive change points in Figure 2. Points such as 15 and 20 would be identified as change points while they are due to noise and not to changes in the behavior of the process. Statistical analysis needs to be conducted to minimize the number of false alarms and accurately identify actual change points.

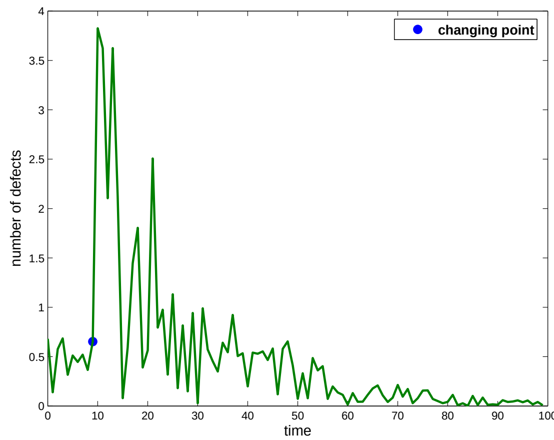

Figure 3. Change point associated with an initial linear decay during system testing.

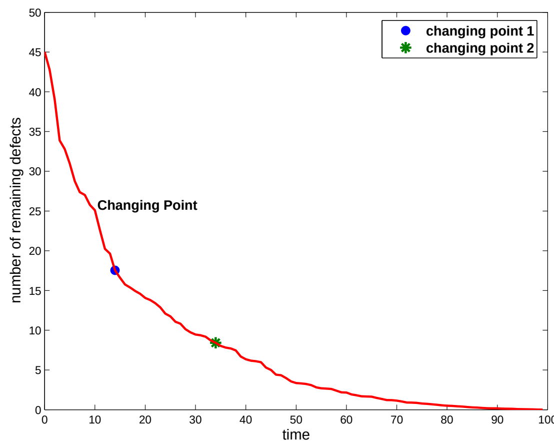

Figure 4. Change points associated with changes in the rate of decay of defects.

Another scenario that frequently occurs during the testing process is the change in the rate of decay of defects. One reason for this change may be the testing of a more defective part of the system. Also, even after system testing has started, sometimes new functionality is integrated to the system. This would also change the rate of decay of defects as the new code may present a higher defect density than the already partially tested code.

Figure 4 shows two change points associated with the

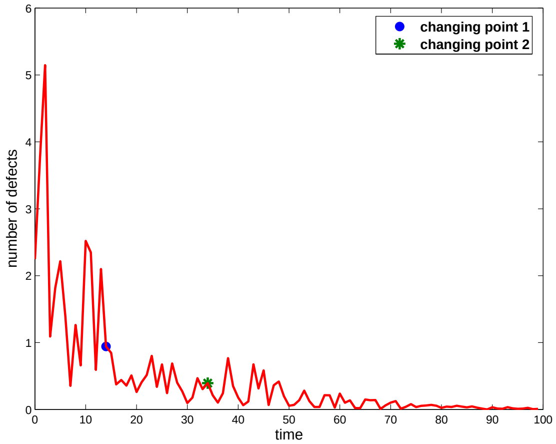

Figure 5. Change point associated with changes in the rate of decay of defects.

events described above. The change points were randomly generated by changing the rate of decay of defects. As it can be seen from Figures 4 and 5, the identification of the change points in this case is more subtle than the previous case and any simple approach to identify them would lead to series of missing or false positive identifications.

5 Detection of multiple changes

In classical change point problems, the distribution of observed data changes at an unknown moment, which is the parameter of interest. Sample $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2)$ consists of two subsamples from distributions f and g, respectively, separated by an unknown change point ν.

A process with *multiple change points* is described as $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \ldots, \boldsymbol{X}_\lambda)$,

$$\left\{ \begin{array}{lcl} \boldsymbol{X}_1 = (X_{\nu_0}, \ldots, X_{\nu_1}) & \sim & f_1 \\ \boldsymbol{X}_2 = (X_{\nu_1+1}, \ldots, X_{\nu_2}) & \sim & f_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots & & \\ \boldsymbol{X}_\lambda = (X_{\nu_{\lambda-1}+1}, \ldots, X_{\nu_\lambda}) & \sim & f_\lambda \end{array} \right. \tag{5}$$

where $f_1, \ldots, f_\lambda$ are either known or unknown densities or trends, ν_i, $i = 1, \ldots, \lambda - 1$, are change points (with a convention that $\nu_\lambda = N$ is the total sample size and $\nu_0 = 1$), and λ is the unknown number of homogeneous subsamples.

The objective here is to estimate $(\lambda; \nu_1, \ldots, \nu_{\lambda-1})$, a parameter of an unknown dimension! The possibility of a homogeneous sample with no change points ($\lambda = 1$) is not excluded.

Several multiple change point estimation schemes have been described in the literature. A maximum likelihood based procedure is proposed in [8]. However, especially in the case of unknown distributions, the naive maximum likelihood scheme is likely to return a change at every point, unless restrictions are enforced on the number of change points λ or the minimum distance Δ between them [12]. Still, this *restricted* maximum likelihood scheme is often inefficient. For instance, it can be shown to detect change points in *any* sample of $N > 2\Delta$ Bernoulli(p) variables if and only if at least two observations are different. It follows that the probability of a false alarm is as high as $1 - p^N - (1-p)^N$ in this case.

A conceptually different *binary segmentation* scheme [19] is an iterative procedure that divides the observed sample into two most distant subsamples, then divides each subsample, etc., until all the obtained subsamples are homogeneous. The disadvantage of this scheme is that no more than one change point is assumed at each step. For example, in the case of two intermittent distributions, it is unlikely to find a point separating two significantly different subsamples.

These problems can be resolved by a *sequential estimation scheme* that (1) considers increasing subsamples instead of the entire sample that may contain complicated patterns; (2) detects one change point at a time and does not assume its uniqueness in the observed data; (3) has an option of detecting 0 change points if the entire sample has the same density or follows the same general trend.

This sequential algorithm, formally described in [2], has found direct applications in epidemiology and energy finance [4].

Step 1: sequential detection

Observations are sampled sequentially until a stopping rule detects a change point. A popular well-studied change point detection scheme is based on the *cusum rule*

$$T(h) = \inf\{n : W_n \geq h\},$$

where

$$W_n = \max_k \log \frac{g(X_{k+1}, \ldots, X_n)}{f(X_{k+1}, \ldots, X_n)}$$

is the *cusum process*, f and g are the pre- and post-change densities or probability mass functions. This stopping rule is known to possess a number of optimal properties, including minimization of the mean delay under a constraint on the rate of false alarms [14, 17]. In the case of unknown or partially known densities, one uses their best estimates for each "potential" value k of a change point, computes the *generalized likelihood ratio* based cumulative sums

$$\tilde{W}_n = \max_k \log \frac{\hat{g}(X_{k+1}, \ldots, X_n)}{\hat{f}(X_{k+1}, \ldots, X_n)} \tag{6}$$

and defines the stopping rule $\tilde{T}(h)$ similarly to $T(h)$. This stopping rule achieves asymptotically equivalent mean delay and mean time between false alarms [1].

Facing a possibility of early or frequent change points, one should increase sensitivity of the algorithm by choosing a low threshold h or a high probability of type I error α. The price to be paid is the increasing rate of false alarms, however, false change points will (hopefully) get filtered at Step 3.

If only a sample of size N is available, all abovementioned stopping rules are curtailed so that $P\{T \leq N\} = 1$. In the case when $T(h) = N$ and $\tilde{W}_n < h$, the scheme results in zero detected change points. In all the other cases, a change point is detected and its location needs to be estimated.

Step 2: post-estimation

Notice that the stopping rule T itself is a poor estimator of the change point ν. Indeed, if $T \leq \nu$, it is a false alarm. If $T > \nu$, it is a biased estimator that always overestimates the parameter. Therefore, the detected change point has to be *post-estimated*, i.e., estimated after its occurrence is detected by a stopping rule.

One way of obtaining an approximately unbiased estimator of ν is to estimate the bias of $T(h)$ and subtract it from $T(h)$. According to [1], this bias, also known as *mean delay*, is asymptotically $(h + C)/K(f, g)$, as $h \to \infty$, where $K(f, g)$ is the Kullback information number, and C is a constant. In the case of sufficiently long phases before and after the change point, subtracting the estimated bias from $T(h)$ yields an approximately unbiased estimator of ν. However, in the case of frequent change points and unknown densities, no reliable estimators of C and K are available.

A *last-zero* estimator

$$\hat{\nu}_{LZ} = \sup \{k < T(h),\ W_k = 0\},$$

proposed in [16] and [18], is essentially the maximum likelihood estimator of ν, assuming a fixed-size sample rather than a sample of a random size T, which is the stopping rule. The corresponding estimator in the case of unknown densities is

$$\tilde{\nu} = \sup \left\{k < \tilde{T}(h),\ \tilde{W}_k = 0\right\}.$$

It can be shown that this estimator fails to satisfy an important property of *distribution consistency* [3].

Notice that for integer-valued λ and $\{\nu_j\}$, distribution consistency is equivalent to convergence in probability. It also implies that a sample with no change points ($\lambda = 0$) provides no false alarms with the probability converging to 1, and the probability of not detecting a change point in a sample with change points converges to 0.

Distribution-consistent schemes exist. One of them is based on the cusum stopping rule $\tilde{T}$ and the *minimum p-value* estimator

$$\hat{\nu}_{MP} = \arg \min_{1 \leq k < \tilde{T}} p(k, \tilde{T}, \boldsymbol{X}),$$

where $p(k, \tilde{T}, \boldsymbol{X})$ is the p-value of the likelihood ratio test comparing subsamples $\mathbf{X_1} = (X_1, \ldots, X_k)$ and $\mathbf{X}_2 = (X_{k+1}, \ldots, X_{\tilde{T}})$.

Step 3: tests of significance

To eliminate false alarms, significance of each detected change point has to be tested. Likelihood ratio tests are easy to implement here, and significance of the detected change point is measured by the minimum p-value $p(\hat{\nu}_{MP}, \tilde{T}, \boldsymbol{X})$.

If the test is significant, one applies steps 1–3 to the post-change subsample $\{X_k, k > \hat{\nu}_{MP}\}$, searching for the next change point. Otherwise, we have a false alarm, and the search continues based on the initial sample, or a part of it starting after the last change point that was found significant.

5.1 Adaptation of the sequential detection scheme for the CDM Model

The described CDM Model does not belong to the type of classical change point models considered by [11] and others. Essentially, involvement of new testing tools followed by recalibration forces abrupt *trend changes* in the parameter of Poisson distribution of the number of defects. The trend, in each segment between successive change points, is linear or exponential, as described above, but its characteristics such as the slope and the intercept change at each change point.

Change point detection algorithms for the intrinsic problem where changes occur in the trend of unobserved parameters were considered in [9] and [10]. The abovementioned cusum change detection algorithm, proposed there, happens to involve computational complications when segments contain unknown parameters.

Alternatively, one can use least-squares method for our types of nonlinear trends because the parameter of Poisson distribution equals its expectation. For each segment, the choice between a linear and an exponential trend model is also decided by the lower sum of squares. Following this simple scheme, one still follows Steps 1-3, with the following modifications:

– maximization of the generalized likelihood ratio in (6) is replaced by the minimization of the weighted sum of squared residuals

$$\tilde{W}_n = \min_k \left\{ \sum_{i=1}^{k} \sqrt{\hat{f}_i}(X_i - \hat{f}_i)^2 + \sum_{i=k+1}^{n} \sqrt{\hat{g}_i}(X_i - \hat{g}_i)^2 \right\};$$

– the stopping rule is defined as

$$T(h) = \inf \{n : p_n \leq \alpha\},$$

where p_n is a $\tilde{W}_n$-based p-value testing significance of a change point at k;

– likelihood ratio tests are replaced by ANOVA F-type tests.

Fitting the linear trend model $\boldsymbol{E}(X_i) = a + bt_i$ and the exponential trend model $\boldsymbol{E}(X_i) = \exp(a + bt_i) - 1$ to each segment of data leads to nonlinear minimization of the sum of squares, however, standard least squares estimates

$$\hat{b} = \frac{\sum(X_i - \bar{X})(t_i - \bar{t})}{\sum(t_i - \bar{t})^2} \text{ and } \hat{a} = \bar{X} - \hat{b}\bar{t}$$

for the linear model, and

$$\hat{b} = \frac{\sum \log(1 + X_i)(t_i - \bar{t})}{\sum(t_i - \bar{t})^2} \text{ and } \hat{a} = \overline{\log(1 + X)} - \hat{b}\bar{t}$$

for the exponential model can serve as the initial approximation.

Once the preliminary set of change point estimates is obtained by the sequential scheme, it can be noticeably refined. Indeed, given the set of estimated change points, each of them can be re-estimated from the segment exactly between the preceding and the succeeding change points. At the same time, change points that are not significant relative to the chosen level α are removed by merging the corresponding segments of data. Each refinement of the estimated set of change points allows to better estimate the remaining change points.

After a cycle of iterations, the procedure converges. As a result, all the remaining change points are significant at the level α, and each of them is estimated from the segment exactly between the neighboring two change points.

Distribution consistency of the proposed scheme follows similarly to [3].

6 Case Studies

The proposed multiple change point detection algorithm described in Section 5 has been applied to a series of 14 data sets from large industrial projects. The described sequential algorithm was coded in MatLab. The duration of these projects range from 50 to more than 400 days of system testing [2]. Due to proprietary reasons, the plots presented here have been normalized to 100% for the decay of defects and to a 100 for the number of defects detected per time unit (testing days in all the case studies). Due to space constraints only one of the 14 case studies is presented next and is referred hereafter as Project α. The results from the other 13 data sets are similar varying mostly with respect to the number of identified change points. The initial scheme detected up to 11 change points although some of them were eliminated during the subsequent refinement and re-testing. The final set of change point estimates contained between 0 and 7 change points significant at the 1% level. Six datasets did not provide a single change point, showing that a model with steady trend is more suitable for them.

Project α has around 100 days of collected data for system testing. The size of the project can be considered as medium to large and the testing team was relatively large with good experience in the testing of similar products. Specific details cannot be given here due to proprietary reasons. However, this does not interfere with the results presented here. Also, Project α is referent to the first release of the product.

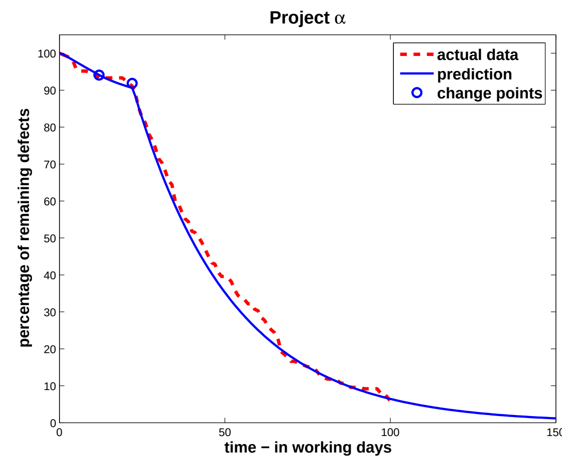

Figure 6. Change point associated with changes in the rate of decay of defects.

As can be seen from Figure 6, Project α has a clear linear decay in the first twenty days of testing. This is due to installation problems in a different environment but is mainly due to severity one defects [3] and also to a learning period for the testers. These issues are common in the first release of products as in the case of Project α. After this initial linear decay, as it can be seen in Figure 6, an exponential decay in the number of remaining defects is observed.

Only small changes in the testing process were made during the time period specified in Figure 6 and therefore recalibration would not be needed for the process. However, a clear change in the behavior of the process occurs around day 20 of Project α. As stated before, in order to increase accuracy and also to account for un-modeled characteristics of the testing process, the application of the `CDM Model` is done using a piecewise approximation. In the case of Project α, to achieve this goal the manager would have to first plot the data and select points of recalibration incurring an extra step in the utilization of the `CDM Model`.

[2] Though system testing was the predominant phase, the integration of new functionality was not uncommon during the phase

[3] Defects classified as severity one in the specific company refers to stop testing defects, that is, testing cannot proceed before the defect can be fixed.

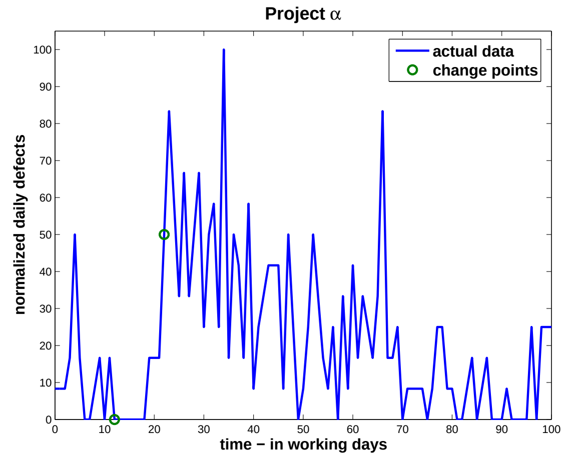

Figure 7. Change point associated with changes in the rate of decay of defects.

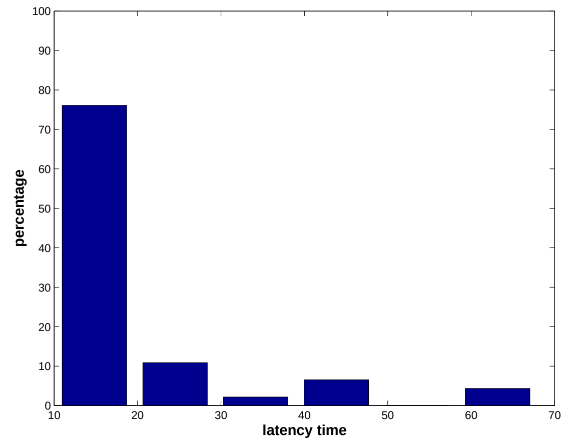

Figure 8. latency in the identification of change points for 14 data sets.

The availability of technique described in Section 5 allows for the automatic identification of changes in the behavior of the process releasing the manager from the task. Two change points are identified for Project α, both being associated with an initial linear decay. The first change point occurs at day 13 and the second at day 23. As can be seen in Figure 6 the use of the identified change points to recalibrate the model, results in a better approximation of the actual behavior of the process. Figure 7 shows the normalized number of defects per time unit and the associated change points. Changes are clear at the initial part of the data and though variation can be observed after that, the overall behavior of an exponential decay is preserved as seen in Figure 6.

7 Performance Analysis

Two features must be taken into consideration when considering the accuracy in the identification of change points. The first one regards the correct identification of the points. A careful analysis of the identified change points for the case study presented in Section 6 and thirteen other projects had revealed the proper identification of the points. Clearly, the significance level plays a major role when identifying the points. The lower the significance level, the more sensitive is the approach. This results in the identification of a smaller number of change points but also avoids false positives. A very small significance level can be used when change points are associated with critical applications. In all the data sets analyzed, a significance level of 0.01 has been used.

The latency in the identification of the change points is the second measurement that provides a good indication of the accuracy of the approach. A change point that is identified too late does not allow time so that the appropriated actions can be taken. For example, assume a change point at time $t = 25$ is identified only at time $t = 150$; by the time of the identification the results of any possible corrective actions are, most likely, negligible. In general, the longer it takes to identify the change points, the smaller the effects of the corrective actions. Therefore, a minimization of the latency time is desired. A minimum number of data points is required to collect enough information to identify the change points. In the case studies conducted, a minimum of 10 data points were chosen to start the identification of a new change point. A total of 46 change points were identified for all 14 data sets analyzed. As can be seen in Figure 8, 76% of the change points were identified using arround 14 data points and 11% using around 24 data points. This is a very good indication of the accuracy of the approach. In only a few cases, the change points were identified very late. However, in all these cases, the changes were so subtle that the late identification would have minimal impact on the re-calibration of the model.

8 Concluding Remarks and Future Work

One of the major factors associated with the wide use of a technique is its usability. Test managers are not interested in learning the intrinsic mathematical details of a technique even though the benefits may justify such effort. A considerably less improvement with low investment is, in general, preferred in detriment of a larger improvement associated with high costs. Furthermore, managers may not have the mathematical skill to apply the methods. The easier a technique is to use, the more it will be used by test managers. The `CDM Model` is not different. Until a friendly interface is available, its use will be limited by studies conducted in conjunction with researchers.

The technique described here is one step closer to allow the implementation of a friendly interface for the `CDM Model`. The technique allows the automatic identification of recalibration points, increasing the accuracy of the predictions and releasing the manager from the tedious work of the manual identification of the points. The results presented here are a clear indication of the applicability of the change point technique to the testing process. The points were successfully identified for the case studies conducted within a reasonable latency.

Some change points may result in just a small increase in the accuracy of the model. However, they can be used to acquire a better understanding of the testing process. If we could trace back all the facts at the days of the change points, we could be able to identify the causes of the changes and use them to improve not only the prediction of the model but mainly the overall behavior of the process. For example, if we knew that the integration of a new functionality was done at certain day and this caused a change in the decay of defects, we could apply the same (or average) decay every time a new functionality is added during system testing. As stated before, though these additions should not happen during system testing, it is not uncommon to observe them. Inferences from the change in behavior could be derived and the expected behavior predicted for future projects. Many other sources of change points can be identified if a detailed study relating their occurrence and cause is conducted. However, such study is outside the scope of this paper and deferred to future work.

References

[1] M. Baron. Nonparametric adaptive change-point estimation and on-line detection. *Sequential Analysis*, 19:1–23, 2000.

[2] M. Baron. Sequential methods for multistate processes. In *N. Mukhopadhyay, S. Datta and S. Chattopadhyay, Eds., Applications of Sequential Methodologies*, New York, 2004. Marcel Dekker, Inc.

[3] M. Baron and N. Granott. Consistent estimation of early and frequent change points. In *Y. Haitovsky, H. R. Lerche, Y. Ritov, eds., Foundations of Statistical Inference, Proceedings of the Shoresh Conference 2000*, pages 181–192, Springer, 2003.

[4] M. Baron, M. Rosenberg, and N. Sidorenko. Automatic spike detection for the modeling and prediction of electricity pricing. *Energy & Power Risk Management*, 2001.

[5] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. PTR Prentice-Hall, Inc., 1993.

[6] João W. Cangussu, Raymond A. DeCarlo, and Aditya P. Mathur. A formal model for the software test process. *IEEE Transactions on Software Engineering*, 28(8):782–796, August 2002.

[7] João W. Cangussu, Raymond A. DeCarlo, and Aditya P. Mathur. Using sensitivity analysis to validate a state variable model of the software test process. *IEEE Transactions on Software Engineering*, 29(5):430–443, May 2003.

[8] Y.-X Fu, R. N. Curnow. Maximum likelihood estimation of multiple change points. *Biometrika*, 77:563–573, 1990.

[9] R. Gill. *Change-Point Analysis of Gradual Change Models in Exponential Families*. Ph. D. Dissertation, The University of Texas at Dallas, 2002.

[10] R. Gill and M. Baron. Consistent estimation in generalized broken-line regression. *Under review*, 2002.

[11] D. V. Hinkley. Inference about the change-point in a sequence of random variables. *Biometrika*, 57:1–17, 1970.

[12] C.-B. Lee. Nonparametric multiple change-point estimators. *Statist. Probab. Letters*, 27:295–304, 1996.

[13] Lennart Ljung. *System Identification: Theory for the user*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[14] G. Lorden. Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42:1897–1908, 1971.

[15] David G. Luenberger. *Introduction to Dynamic Systems: Theory, models and applications*. John Wiley & Sons, New York, 1979.

[16] D. C. Montgomery. *Introduction to Statistical Quality Control, Third Ed.* Wiley, New York, 1997.

[17] Y. Ritov. Decision theoretic optimality of CUSUM procedure. *Ann. Statist.*, 18:1464–1469, 1990.

[18] M. S. Srivastava, Y. Wu. Quazi-stationary biases of change point and change magnitude estimation after sequential cusum test. *Sequential Analysis*, 18:203–216, 1999.

[19] L. Ju. Vostrikova. Detecting “disorder” in multidimensional random processes. *Sov. Math. Dokl.*, 24:55–59, 1981.