Statistical Analysis of Field Data for Repairable Systems

David Trindade and Swami Nathan
4140 Network Circle
Mailstop : USCA14-204
Santa Clara, CA 95054
Email : david.trindade @sun.com, swami.nathan@sun.com

SUMMARY and PURPOSE

The purpose of the tutorial is to present simple graphical methods for analysing the reliability of repairable systems. Many talks and papers on repairable systems analysis deal primarily with complex parametric modeling methods. Because of their highly esoteric nature, such approaches rarely gain wide acceptance into the reliability monitoring practices of a company. This tutorial will present techniques based on non-parametric methods which have been successfully used within Sun Microsystems to transform the way reliability of repairable systems is analysed and communicated to management and customers.

Upon completion of this tutorial, attendees should be able to analyse a large dataset of repairable systems, identify trends in the rates of failures, identify outliers, causes of failures and present this information using a series of simple plots that can be understood by management, customers and field support engineers alike.

David Trindade, Ph.D.

Dr. David Trindade is a Distinguished Engineer at Sun Microsystems. Formerly he was a Senior Fellow at AMD. His fields of expertise include reliability, statistical analysis, and modeling of components, systems, and software; and applied statistics, especially design of experiments (DOE) and statistical process control (SPC). He is co-author (with Dr. Paul Tobias) of the book *Applied Reliability*, 2nd ed., published in 1995. He has a BS in Physics, an MS in Statistics, an MS in Material Sciences and Semiconductor Physics, and a Ph.D. in Mechanical Engineering and Statistics. He has been an adjunct lecturer at the University of Vermont and Santa Clara University.

Swami Nathan, Ph.D.

Dr. Swami Nathan is a senior staff engineer at Sun Microsystems. His field of interest is on field data analysis, statistical analysis and reliability/availability modeling of complex systems. He received his B.Tech from Indian Institute of Technology, and M.S. and Ph.D in reliability engineering from the University of Maryland, College Park. He has authored over twenty papers in peer reviewed journals and international conferences and holds 2 patents.

Table of Contents

1. INTRODUCTION

A repairable system, as the name implies, is a system which can be restored to operating condition in the event of a failure. The restoration involves any manual or automated action that falls short of replacing the entire system. Common examples of repairable systems include computer servers, network routers, printers, automobiles, locomotives, etc. Although repairable systems exist in all walks of life, the techniques for analysing repairable systems are not as prevalent as those for non-repairable systems. This situation often leads to incorrect analysis techniques due to confusion between the hazard rate and rate of occurrence of failures [1,2].

The techniques for repairable systems found in the literature are primarily parametric methods, requiring a certain degree of statistical knowledge on the part of the practitioner. The difficulty of communicating techniques, such as testing distributional assumptions, to management renders them impractical for widespread usage within an organization.

Recently analysis of repairable systems based on non-parametric methods are becoming increasingly popular due to their simplicity as well as ability to handle more than just *counts of recurrent events* [3,4,5,6,7]. This tutorial provides a simple yet powerful approach for performing reliability analysis of repairable systems using non-parametric methods. Innovative reliability plotting methods are explored for the identification of trends, discerning deeper issues relating to failure modes, assessing effects of changes and comparing across platforms, vintages, environments etc. These approaches have been applied with great success to datacenter systems (both hardware and software), and the tutorial is based on courses and training sessions given to sales, support services, management, and engineering personnel within Sun Microsystems™. These techniques can be easily applied within a spreadsheet environment such as StarOffice™ or Excel™ by anybody and demands only a very rudimentary knowledge of statistics. Interesting examples and case studies from actual analysis of computer servers at customer datacenters will be provided for all concepts.

1.1　Notation and Acronyms

MTBF　　　mean time between failure
MCF　　　mean cumulative function
CTF　　　calendar time function
RR　　　recurrence rate
ROCOF　　rate of occurrence of failures
HPP　　　homogeneous poisson process
NHPP　　non-homogeneous poisson process

2. DANGERS of MTBF

The most common metric used to represent the reliability of repairable systems is an MTBF, which is calculated by adding all the operating hours of all the systems and dividing by the number of failures. The popularity of the MTBF metric is due to its simplicity and its ability to cater to the *one number syndrome*. MTBFs are often stated by equipment manufacturers with imprecise definitions of a failure most often in *fine*

print. MTBF hides information by not accounting for any trends in the arrival of failures and treating machines of all ages as coming from the same population.

There are several assumptions involved in stating an MTBF. Firstly, it is assumed that the failures of a repairable system follow a renewal process, i.e., all failure times come from a single population distribution. A further assumption is that the times between events are independent and exponentially distributed with a constant rate of occurrence of events, and consequently, we have a homogeneous Poisson process (HPP). The validity of a HPP is rarely checked in reality. As a result, strict reliance on the MTBF without full understanding of the consequences can result in missing developing trends and drawing erroneous conclusions.
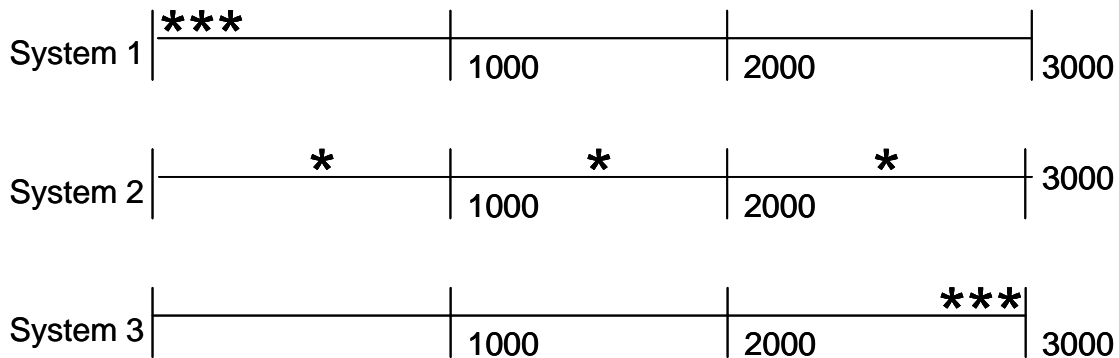
Figure 1   MTBF hides information

In Figure 1, we have three systems that have operated for 3000 hours, with each experiencing three failures. Thus, all three systems have the same MTBF of 1000 hours. However, System 1 had three early failures and none thereafter. System 2 had a failure in each 1000 hour interval while System 3 had three late failures. The behaviour of the three systems are dramatically different and ***yet they have the same MTBF!***

Clearly there is a need for better reliability metrics that account for trends in the failure data.

2.1   The "Failure Rate" Confusion

Well intentioned practitioners often invert the MTBF and quote a failure rate. However the term failure rate has become a confusing term in the literature [1,2]. Often engineers analyse data from repairable systems using methods for the analysis of data from non-repairable systems. Let us consider the failure of a computer due to a central processing unit or CPU. The computer is a repairable system while the CPU is a non-repairable component. When we have a dataset of times of failures of the computer due to the CPU, analysts often take the times between failures and treat them as times to failures of CPUs. What this implies is that the times to failures of individual CPUs arise from the same distribution, i.e., they are independent and identically distributed. Consequently, the

4

sequence of times to failures are neglected. This assumption is valid only if the distribution of the times to first failure is identical to the distribution of the times to second failure, and so on *ad infinitum*. If for example, the cooling fan inside the computer is degrading, then the times to successive CPU failures will start getting shorter, violating the *iid* assumption. Usher[2] shows an interesting case study where the times between failures are treated as lifetimes from the same distribution to fit a Weibull distribution with a decreasing hazard rate, while a simple cumulative plot shows that the rate of occurrence of failures is actually increasing! The hazard rate is a property of a time to failure while ROCOF is a property of a sequence of times to failures i.e., order of occurrence of failures matters.

Despite Ascher's passionate arguments more then twenty years ago [14], the term failure rate continues to be arbitrarily used in the industry and sometimes in academia to describe both a hazard rate of a lifetime distribution of a non-repairable system and a rate of occurrence of failures of a sequences of failure times of a repairable system. This lack of distinction can lead to poor analysis choices even by well intentioned individuals.

3. PARAMETRIC METHODS

One of the common parametric approaches to modeling repairable systems reliability typically assumes that failures occur according to a non-homogeneous Poisson process with an intensity function. One of the popular intensity functions is the power law Poisson process [8,9] or "Weibull Poisson process" which has an intensity function of the form

$$u(t) = \lambda \beta t^{\beta-1} \qquad \lambda, \beta > 0 \qquad (1)$$

The probability that a system experiences n failures in t hours has the following expression

$$P\big(N(t) = n\big) = \frac{\big(\lambda t^{\beta}\big)e^{-\lambda t^{\beta}}}{n!} \qquad (2)$$

To estimate the two parameters in the model one can use maximum likelihood estimation. The equations for the parameter estimates are given in [8,9]

$$\hat{\lambda} = \frac{\displaystyle\sum_{q=1}^{K} N_q}{\displaystyle\sum_{q=1}^{K} T_q^{\hat{\beta}} - S_q^{\hat{\beta}}}, \qquad \hat{\beta} = \frac{\displaystyle\sum_{q=1}^{K} N_q}{\hat{\lambda}\displaystyle\sum_{q=1}^{K}(T_q^{\hat{\beta}} \ln T_q - S_q^{\hat{\beta}} \ln S_q) - \displaystyle\sum_{q=1}^{K}\sum_{i=1}^{N_q} \ln X_{iq}} \qquad (3)$$

where we have K systems, S and T are start and end times of observation accounting for censoring, N_q is the number of failures on the qth system and X_{iq} is the age of the qth

system at the *i*th failure.

These equations cannot be solved analytically and require an iterative procedure or special software. Crow [8] also provides methods for confidence interval estimation and a test statistic for testing the adequacy of the power law assumption.

Further extensions of renewal process techniques known as Generalized Renewal Process were proposed by Kijima[10,11]. Kijima models removed several of the assumptions regarding the state of the machine after repair present in earlier models. However, because of the complexity of the renewal equation closed form solutions are not possible and numerical solutions can be quite tedious. A Monte Carlo simulation based approach for the Kijima formulation was developed in [12]. Mettas and Zhao [13] present a general likelihood function formulation for estimating the parameters of the general renewal process in the case of single and multiple repairable systems. They also provide confidence bounds based on Fisher information matrix.

Despite the abundance of literature on the subject, parametric approaches are computationally intensive and not intuitive to the average person who performs data analysis to support his/her particular customer. Special solution techniques are required along with due diligence in justifying distributional assumptions (rarely done in practice).

Non parametric approaches based on MCFs are far simpler, understandable by lay persons and customers, and are easily implementable in a spreadsheet. The next sections cover the methodology.


4. MEAN CUMULATIVE FUNCTION
4.1   Cumulative Plot

Given a set of failure times for a repairable system, the simplest graph that can be constructed is a cumulative plot. The cumulative plot is a plot of the number of failures versus the age of the system. This plot can be constructed for all failures, outages, system failures due to specific failure modes etc. A cumulative plot can be constructed for just 1 machine or for a group (all) machines in a population. Figure 2 shows an example cumulative plot. There are four machines in the population. We have data on the age of the machine at various failure events. The cumulative plot reveals the evolution of failures with time. For example, machine C had one failure at 50 days and was failure free for over 400 days. After about 450 days machine C had a rash of failures within the next 100 days of operation.
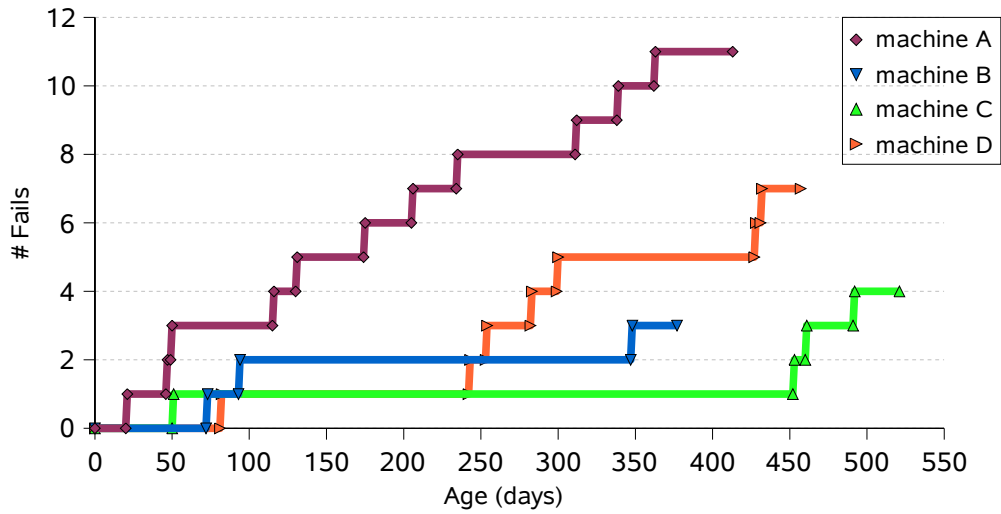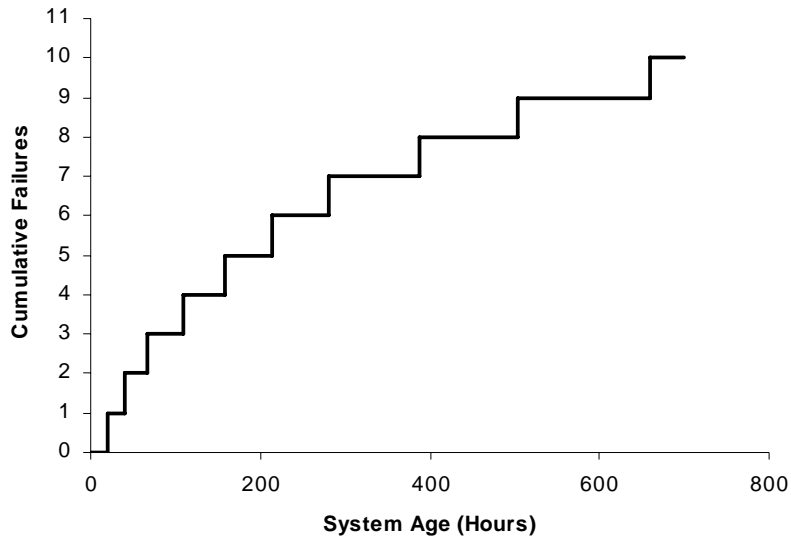
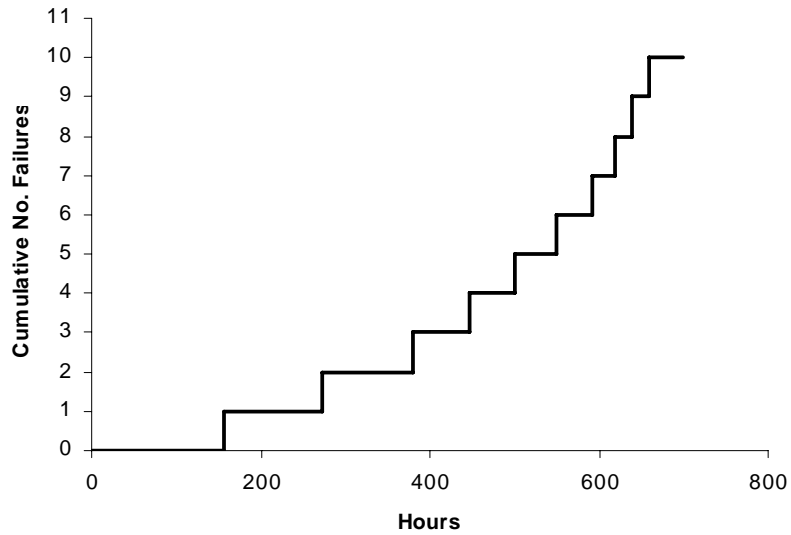**Figure 2 Cumulative plots for a group of four machines.**

Although a cumulative plot looks quite simple it is of great importance because of its ability to reveal trends. Figures 3, 4, and 5 show three different cumulative plots.
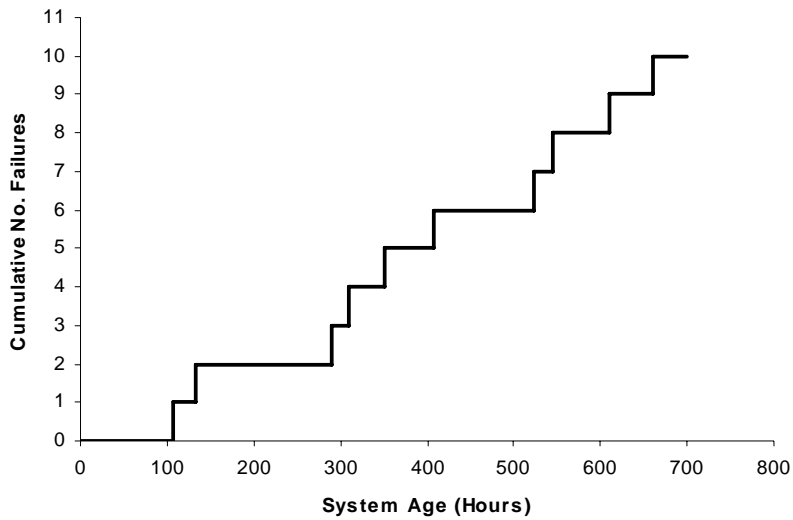
Figure 3   Cumulative plot for a trendless system.

Figure 4   **Cumulative plot for an improving system**

Figure 5   **Cumulative plot for a worsening system.**

The shape of the cumulative plot can provide ready clues as to whether the system is improving, worsening, or trendless. An improving system has the times between failures lengthening with age (takes longer to get to the next failure) while a worsening system has times between failures shortening with age (takes less time to get to the next failure).
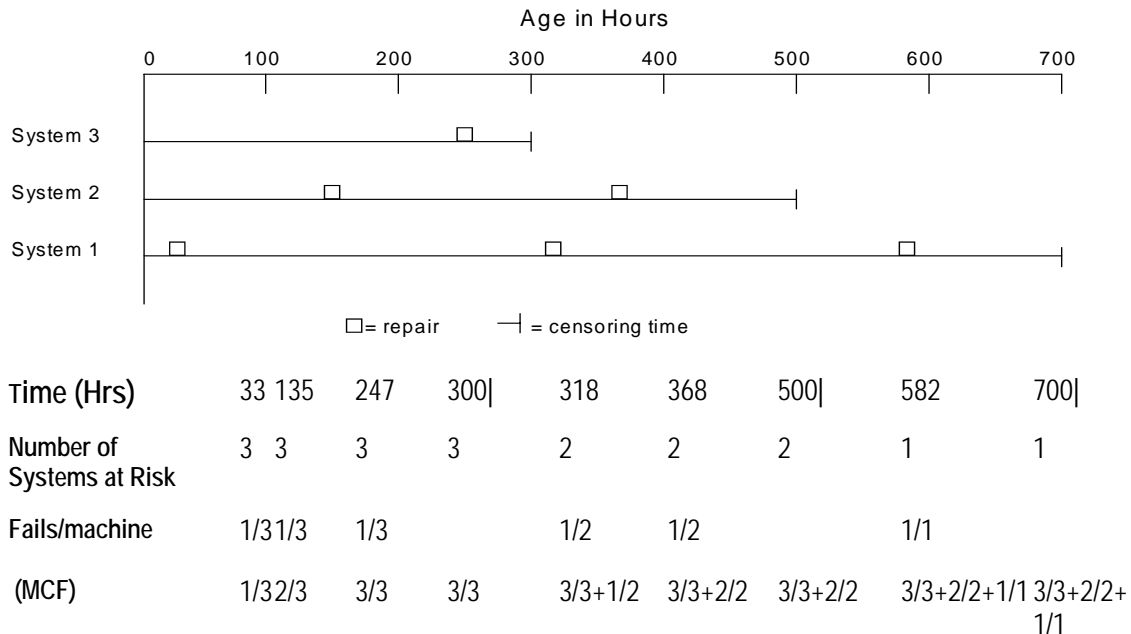
It is to be noted that all three plots show a system with 10 failures in 700 hours, i.e., MTBF of 70 hours. Despite having identical MTBFs, the behaviour of the three systems are dramatically different.

4.2   Mean Cumulative Function versus Age

If there are lots of machines in the population it would be fairly tedious to construct cumulative plots for each individual machine. A useful construct would be to plot the average behaviour of these numerous machines. This is accomplished by calculating the Mean Cumulative Function (MCF).

The MCF is constructed incrementally at each failure event by considering the number of machines at risk at that point in time. The number of machines at risk depends on the how many machines are contributing information. Information can be obscured by the presence of censoring and truncation. Right censoring occurs when information is not available beyond a certain age, e.g., a machine that is 100 days old cannot contribute information to the reliability at 200 days, and hence is not a *machine at risk* when calculating the average at 200 days. Similarly information may be obscured at earlier ages if for example a machine is installed on Jan 1 2004 and service contract was initiated on Jan 1 2005. In this case there is no failure information available during the 1st year of operation. Therefore, this machine cannot contribute any information before 365 days of age but will factor into the calculation only after 365 days. One could also have interval or window censoring that is dealt with extensively in [15]. The MCF accounts for gaps in information by appropriately normalizing by the number of machines at risk.

The example below illustrates a step by step calculation of the MCF for three systems.

Time (Hrs)	33 135	247	300|	318	368	500|	582	700|
Number of Systems at Risk	3 3	3	3	2	2	2	1	1
Fails/machine	1/3 1/3	1/3		1/2	1/2		1/1	
(MCF)	1/3 2/3	3/3	3/3	3/3+1/2	3/3+2/2	3/3+2/2	3/3+2/2+1/1	3/3+2/2+ 1/1

Figure 6   Step by step calculation of the MCF.

9

The ages of the systems at failure and censoring are first sorted by magnitude. The row of times in the table above show the evolution of events by age. At age 33 system 1 had a failure, and since three machines operated beyond 33 hours, the fails/machine is 1/3 and the MCF is 1/3. The MCF aggregates the fails/machine at all points in time where failures happen. At 135 hours, system 2 has a failure and there are still 3 machines at risk in the population. Therefore the fails/machine is 1/3, and the MCF aggregate of the fails/machine at points of failure is now 2/3. Similarly at 247 hours the MCF jumps to 3/3 due to a failure of System 3. At 300 hours, system 3 drops out of the calculation and the number of machines at risk becomes two. System 3 drops out not because it is removed (in this case) but simply because it is not old enough to contribute information beyond its current age. At 318 hours, system 1 has a failure and the fails/machine is now1 /2 since we have only two machines in the population that are contributing information. The MCF now becomes 3/3+1/2 and so on. This fairly straightforward procedure can be easily implemented in a spreadsheet.
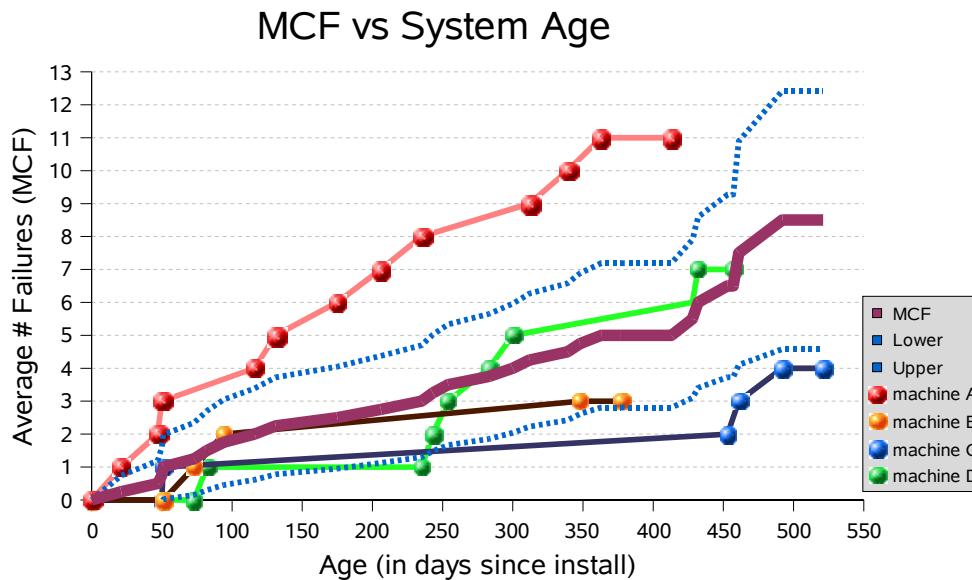
Figure 7 : MCF and confidence intervals for the population in figure 2.

Figure 7 shows the MCF for the population of machines shown in figure 2. The MCF represents the average number of failures experienced by this population as a function of age. If a new machine enters the population, the MCF represents its expected behaviour. Confidence intervals can be provided for the MCF. Nelson[3,7] provides several procedures for pointwise confidence bounds.

### 4.3   Identifying anomalous machines

In computer systems installed in datacenters, often a small number of misbehaving machines tend to obscure the behaviour of the population at large. When the sample sizes

are not too large, the simple confidence bounds can serve to graphically point out machines that have been having an excessively high number of failures compared to the average. Although it is not a statistically correct test of an outlier, overlaying the cumulative plots of individual machines with the MCF and confidence bounds tend to visually point to problem machines. Support engineers can easily identify these problem machines and propose remediation measures to the customer. More rigorous approaches for identifying these anomalous machines has been the subject of recent research. Glosup [16] proposes an approach for comparing the MCF with N machines with the MCF for (N-1) machines and arrive at a test statistic for determining if the omitted machine had a significant influence on the MCF. Heavlin[17] proposed a powerful alternate approach based on 2X2 contingency tables and the application of Cochran Mantel Hanzel statistic to identify anomalous machines.

4.4   Recurrence Rate vs Age

Since the MCF is the cumulative average number of failures versus time one can take the slope of the MCF curve to obtain a rate of occurrence of events as a function of time. This slope is called the recurrence rate to avoid confusion with terms like failure rate [7].

The recurrence rate can be calculated by a simple numerical differentiation procedure i.e., estimate the slope of the curve numerically. This can be easily implemented in a spreadsheet using the Slope(Y1:Yn;X1:Xn) function where MCF is the Y axis and time is the X axis. One can take 5 or 7 adjacent points and calculate the slope of that section of the curve by a simple ruler method and plot the slope value at the midpoint. The degree of smoothing is controlled by the number of points used in the slope calculation [18]. The rate tends to amplify sharp changes in curvature in the MCF. If the MCF rises quickly, it can be seen by a sharp spike in the recurrence rate, and similarly, if the MCF is linear the recurrence rate is a flat line. When the recurrence rate is a constant, it may be a reasonable assumption to conclude that the data follows a HPP, allowing for the use of
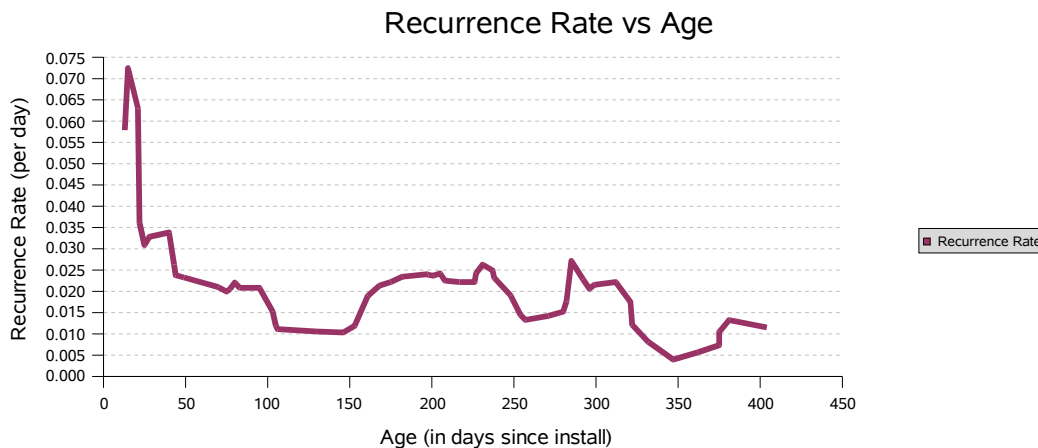
metrics such as MTBF to describe the reliability of the population.

Figure 8   Example of Recurrence Rate vs Age

11

One can see from Figure 8 that the recurrence rate is quite high initially and drops sharply after around 50 days. Beyond 50 days the recurrence rate is fairly trendless and keeps fluctuating around a fairly constant value. If the cause of failures were primarily hardware, then this would indicate potential early life failures, and one would resort to more burn-in or pre release testing. In this example, the cause of the failures were more software and configuration type issues. This problem was identified as *learning curve issues* with systems administrators. When new software products are released, there is always a learning process to figure out the correct configuration procedures, setting up the correct directory paths, network links and so on. These activities are highly prone to human error because of lack of knowledge, improper documentation, and installation procedures. Making the installation procedure simpler and providing better training to the systems administrators resolved this issue in future installs of the product.

5. Calendar Time Analysis

Most reliability literature focuses on analysing reliability as a function of the age of the system. In the case of advanced computing and networking equipment installed in datacenters, the systems undergo changes on a routine basis. There are software patches, upgrades, new applications, hardware upgrades to faster processors, larger memory, physical relocation of systems, etc. This situation can be quite different from other repairable systems like automobiles where the product configuration is fairly stable since production. Cars may undergo changes in the physical operating environment, but rarely do we see upgrades to a bigger transmission.

In datacenter systems many of the effects will not be age dependent but are a result of operating procedures that change the configuration and operational environment. These changes are typically applied to a population of machines in then datacenter and the machines can all be of different ages. It will be difficult to catch changes if the analysis is done as a function of the only of the age of the machine, but some effects will be quite evident when the events are viewed in calendar time [6]. This possibility is illustrated in Figures 9a and 9b.
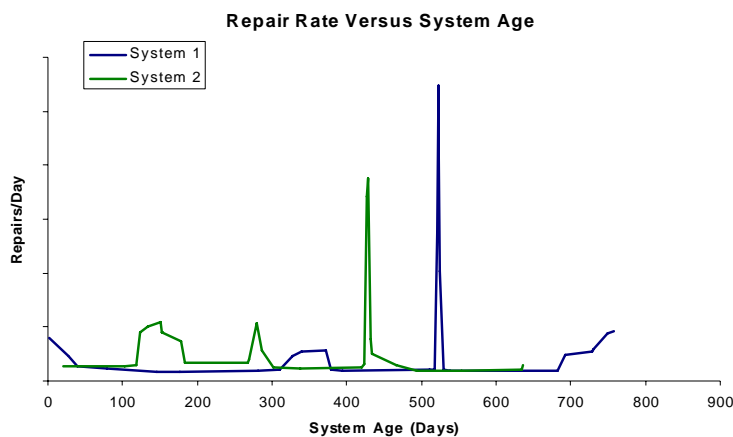
Figure 9 a   Recurrence Rates for two systems versus system age.

Repair Rate Versus Calendar Date
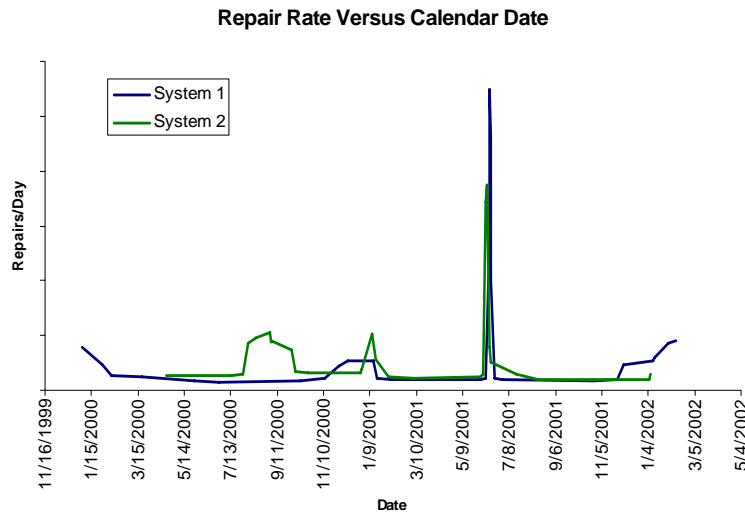
**Figure 9 b   Recurrence Rates for 2 systems versus Date**

Figure 9a shows the recurrence rate vs age for two systems i.e., the slopes of their cumulative plots. One can see that System 1 had a spike in the rate around 450 days while system 2 had a spike in the rate around 550 days. When looked at purely from an age perspective one can easily conclude that they were two independent spikes related only to that particular system. However in figure 9b the recurrence rate vs date shows that the two spikes coincide on the same date. This indicates clearly that we are not dealing with an age related phenomenon but an external event related to calendar time. In this case it was found that a new operating systems patch was installed on both machines at the same time, and shortly thereafter, there was an increase in the rate of failures. By plotting the date as a function of calendar time one can easily separate the age related phenomenon from the date related phenomenon.

In order to analyse the data in calendar time, one can perform an analogous procedure by calculating the cumulative average number of fails per machine at various dates of failure. This result is called the Calendar Time Function (CTF). We begin with the date on which the first machine was installed and calculate the number of machines at risk at various dates on which events occurred. As more machines are installed, the number of machines at risk keeps increasing until the current date. The population will decrease if machines are physically removed from the datacenter at particular dates. This consideration is contrary to the machines at risk as a function of age where the number of machines will be maximum at early ages and will start decreasing as machines are no longer old enough to contribute information. The calculation is identical to the table shown in Figure 5 except that we have calendar dates instead of age. The recurrence rate versus date is extremely important in practical applications because support engineers and customers can more easily correlate spikes or trends with specific events in the datacenter. The calculation of the recurrence rate vs date is identical to the procedure

13

outlined for recurrence rate vs age. The Slope() function in spreadsheets automatically converts dates into days elapsed and can calculate a numerical slope. This routine is an extremely useful and versatile function in spreadsheets.Figure 9b shows an example of a recurrence rate vs date.
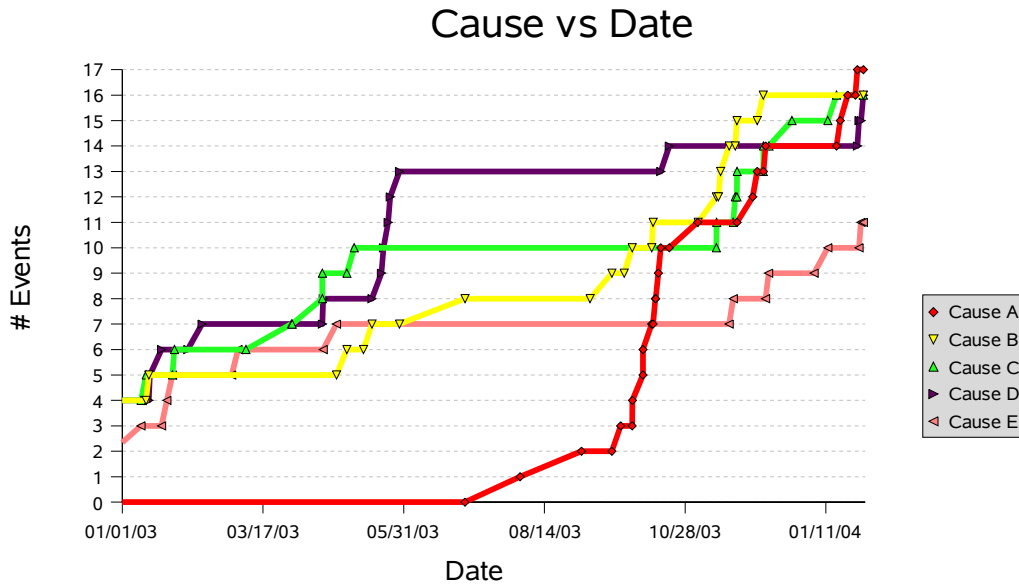
6. Failure Cause Plots

The common approach to representing failure cause information is a Pareto chart or simple bar chart as shown in Figure 10.
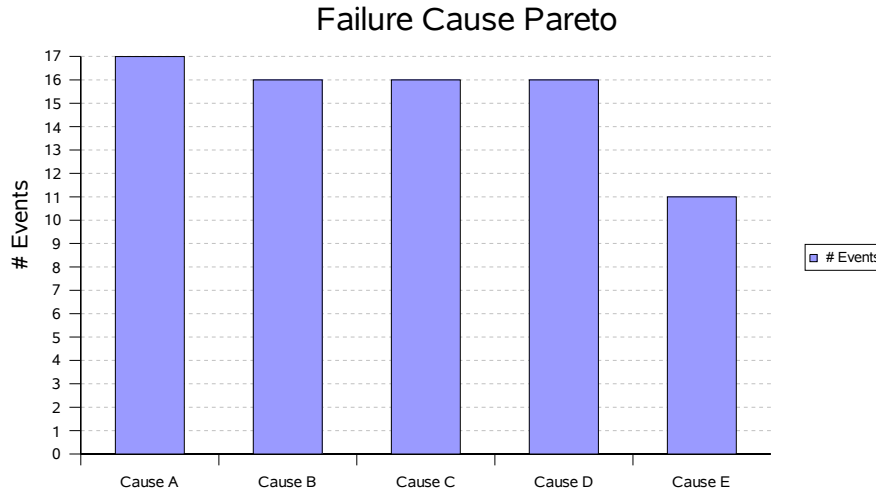
Figure 10   Example Pareto chart showing failure causes.

One can conclude that Cause A is the highest ranking cause while causes B,C and D are all equal contributors, while Cause E is the lowest ranked cause in terms of counts. However, one can see that the above chart has no time element, i.e., one cannot tell which causes are currently a threat and which have been remediated. Yet this chart is one of the most popular representations in the industry. One can plot the failure causes as a function of time (age or calendar) to ascertain various hypotheses. Figure 11 shows the same plot as a function of calendar time, and it is quite revealing.

14

Cause vs Date

Figure 11   Failure causes vs date.

One can see that even though Cause A is only slightly higher than the other causes in Figure 10, its effect is dramatic when viewed in calendar time. It was non-existent for a while but became prevalent around September, with an extremely increasing trend. Even though Figure 10 showed that causes B,C and D were all equal contributors their contributions in time are clearly not equivalent. Cause E was shown as the lowest ranked cause but we can see in Figure 11 that even though it has been dormant for a long time, there have been a rash of cause E events in very recent times, a situation that needs to be addressed immediately.

In Figure 11 the causes are plotted simply as counts. One can definitely plot MCFs for each of the causes and normalize them by the machines at risk. One can easily imagine an MCF of all events with MCFs for individual causes plotted along with it to show the contribution of each cause to the overall MCF at various points in time.

7 MCF comparisons
 7.1   Comparison by Location, Vintage or Application

Often the attention is on comparing populations of machines. Customers are interested in comparing a population of machines in Datacenter X with their machines in Datacenter Y to see if there are differences in operating procedures. Engineers might be interested in comparing machines running high performance technical computing with machines running online transaction processing to see if the effect of applications is something that needs to be considered in designs. Manufacturing might be interested in comparing machines manufactured in a particular year with machines manufactured in the following year to see if there are tangible improvements in reliability. Typically people compare MTBFs of random subgroups and try to conclude if there are differences. This approach is not a correct because of inherent flaws in the MTBF metric. The MCF by virtue of

15

being time dependent and normalization by the number of machines at risk facilitates meaningful comparisons of two or more populations. Figure 12 compares populations of machines belonging to the same customer but located in different datacenters.

MCF by Location
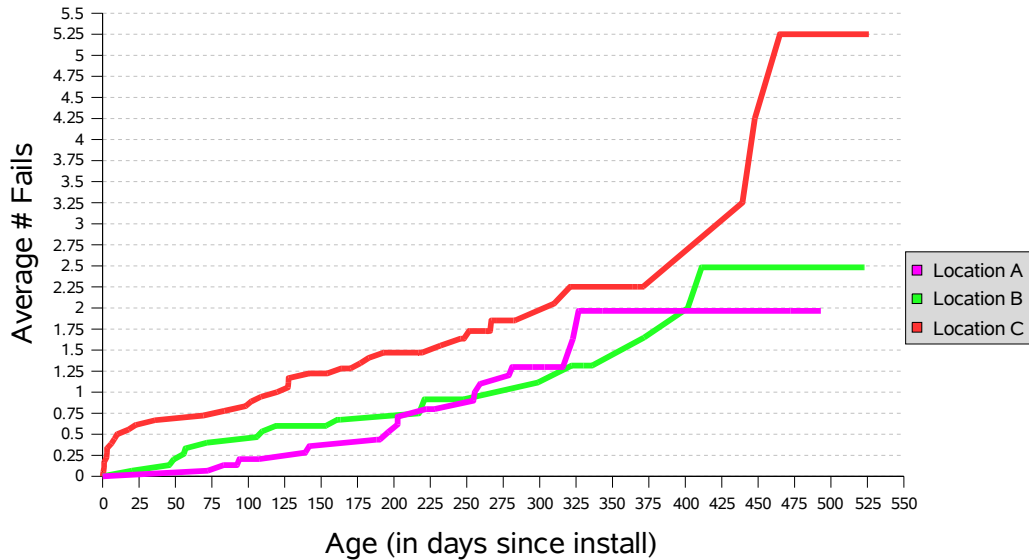
Figure 12   Comparison by Location.

One can see that Location C has an MCF that has been consistently higher than the other locations. The difference between the locations starts to become visually apparent after about 300 days. Investigation into the procedures at Location C revealed that personnel were not following correct procedures for avoiding electrostatic discharge while handling memory modules. This was rectified by policy and procedural changes and the reliability at this location improved. One can see that flattening of the MCF towards the end becoming parallel with the MCFs for the other locations, i.e., same slope or recurrence rate. Nelson provides procedures for assessing statistically significant difference between two MCFs [3].

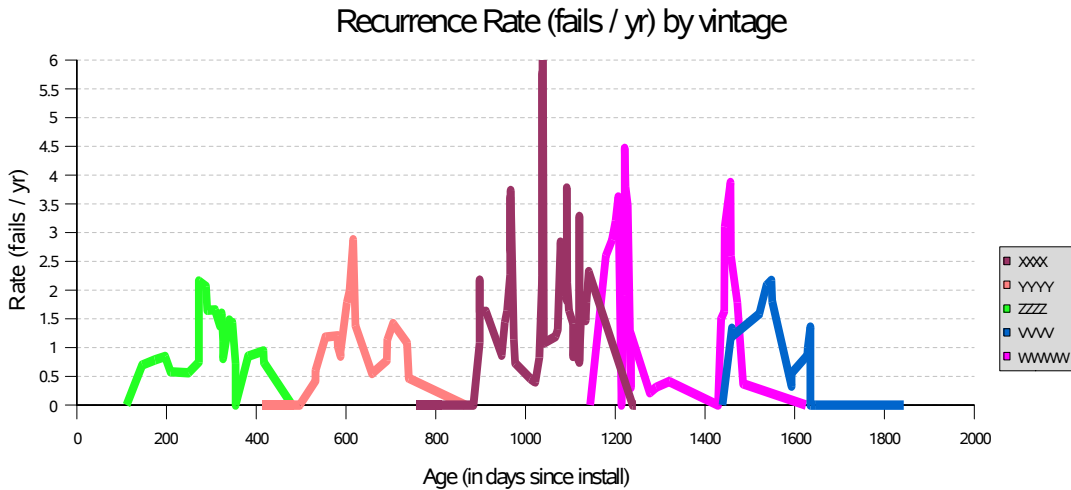### 7.2   *Comparing recurrence rates by vintage to handle left censoring/truncation*

Often times there are gaps in data collection, i.e., machines may have been installed since 1999 and data collection begins in a window starting only after 2003 because that was when service contracts were initiated. Due to the amount of missing information in the earlier ages it would be difficult to compare MCFs because we don't know how many failures have occurred before we started collecting data. One may apply statistical regression models on the MCF in the measurement window to estimate failure counts at the beginning of the window, and thereby create adjusted MCF curves as shown in Figure 13.

16
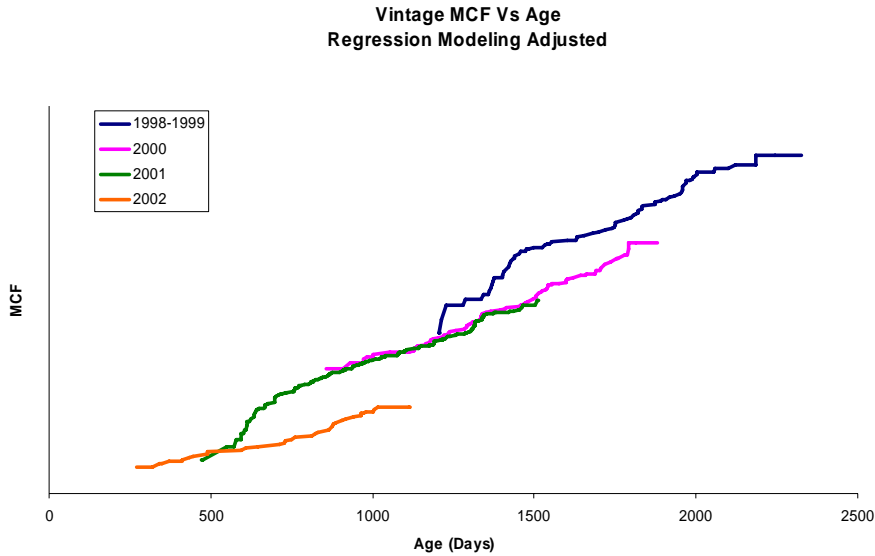
Figure 13   Adjusted MCF curve to Account for Window Truncation

Alternatively, under this situation it would be advantageous to compare recurrence rates as a function of time instead of expected number of failures. This idea is shown in Figure 14.

Figure 14   Comparison of recurrence rates vs age by vintage

Machines manufactured in year XXXX appear to have the highest rate of failures. Year WWWW had small spikes in the rate due to clustering of failures but otherwise has enjoyed long periods of low rates of failure. There appears to be no difference among years VVVV, YYYY and ZZZZ. There does not appear to be a statistically rigorous procedure to assess significant difference between two recurrence rate curves. Visual interpretation has proved to be sufficient in practical experience.

17

8. MCF Extensions

All parametric methods apply primarily to "counts" data i.e., they provide an estimate of the expected number of events as they are generalizations of counting processes. However, the MCF is far more flexible than just counts data. It can be used in availability analysis by accumulating average downtime instead of just average number of outage events. MCFs can be used to track service cost per machine in the form of Mean Cumulative Cost Function. It can be used to track any continuous cumulative history (in addition to counts) such as energy output from plants, amount of radiation dosage in astronauts etc. In this section we show two such applications that are quite useful for computer systems, namely downtime for availability and service cost.

8.1 Mean Cumulative Downtime Function

Availability is of paramount importance to computing and networking organizations because of the enormous costs of downtime to business. Such customers often require service level agreements on the amount of downtime they can expect per year and the vendor has to pay a penalty for exceeding the agreed upon guarantees. For such situations it is useful to plot the cumulative downtime for individual machines and get a cumulative average downtime per machine as a function of time. The calculation would proceed identical to Figure 6 except that the integer counts of failure are replaced by the actual downtime due to the event. Since availability is a function of both the number of outage events and the duration of outage events, one needs to plot the Mean Cumulative Downtime Function as well as the MCF based on just outage events. Sometimes the cumulative downtime may be small but the number of outage events may be excessive because of the amount of failure analysis overhead that goes into understanding the outage. Contracts are often drawn on both the number of outage events as well as the amount of downtime. Figure 15 shows an example mean cumulative downtime function.
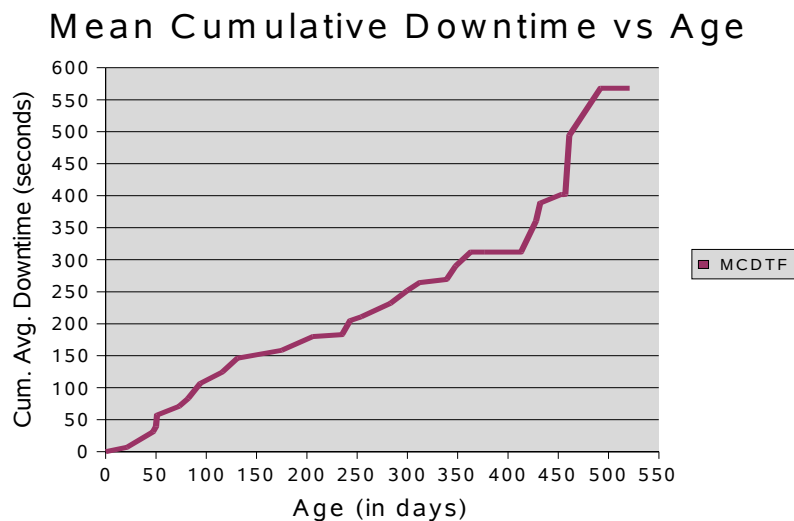
Figure 15   **Mean cumulative downtime function**

18

8.2   Mean Cumulative Cost Function

This application is quite similar to the downtime analysis mentioned in the previous section. This cost analysis could be performed by the vendor on service costs to understand one's cost structure, warranty program, pricing of subscription programs for support services, etc. The cost function could also be created by the customer to track the impact of failures on the business. The notion of downtime and the outage events can be combined to just one plot by looking at cost. The cost would be lost revenue due to loss of functionality plus all administrative costs. So in the situation of lots of outages with small amounts of downtime, the administrative costs will become noticeable. Again the calculation of the mean cumulative cost function would be similar to one of calculating an MCF for failure events except costs are used instead of counts of failures. These mean cumulative cost and downtime functions enjoy all the properties of an MCF in terms of being efficient non parametric estimators and identifying trends in the quantity of interest.

9. CONCLUSIONS

This tutorial addressed the dangers of using summary statistics like MTBF and the important distinction between analysing the data as a non-repairable or repairable system. The analysis of repairable systems does not have to be difficult. Simple graphical techniques can provide excellent estimates of the expected number of failures without resorting to solving complex equations or justifying distributional assumptions. MCFs as a function of calendar time can provide important clues to non age related effects for certain classes of repairable systems. MCFs and recurrence rates are quite versatile because of their extensions to downtime and cost while parametric methods mostly handle counts type data. The approaches outlined in this tutorial have been successfully implemented at Sun Microsystems and also have found ready acceptance among people of varied backgrounds, from support technicians and executive management to statisticians and reliability engineers.

19

REFERENCES

1. H. Ascher, "A set of number is NOT a data-set", *IEEE Transactions on Reliability,* Vol 48 No. 2 pp 135-140, June 1999.
2. J. Usher, " Case Study : Reliability models and misconceptions", *Quality Engineering*, **6**(2), pp 261-271, 1993
3. W. Nelson, *Recurrence Events Data Analysis for Product Repairs, Disease Recurrences and Other Applications*, ASA-SIAM Series in Statistics and Applied Probability, 2003.
4. P.A. Tobias, D. C. Trindade, *Applied Reliability*, 2nd ed., Chapman and Hall/CRC, 1995.
5. W.Q. Meeker, L.A. Escobar, *Statistical Methods for Reliability Data*, Wiley Interscience, 1998.
6. D. C. Trindade, Swami Nathan, Simple Plots for Monitoring the Field Reliability of Repairable Systems, *Proceedings of the Annual Reliability and Maintainability Symposium (RAMS)*, Alexandria, Virginia, 2005.
7. W. Nelson, "Graphical Analysis of Recurrent Events Data", *Joint Statistical Meeting, JSM'05*, Minneapolis, Aug 2005.
8. L. H. Crow, Reliability Analysis of Complex Repairable Systems in *Reliability and Biometry,* ed. By F. Proschan and R.J. Serfling, pp. 379-410, 1974, Philadelphia, SIAM.
9. L.H. Crow, Evaluating the Reliability of Repairable Systems, *Proceedings of the Annual Reliability and Maintainability Symposium (RAMS)*, pp 275-279, 1990.
10. M. Kijima, Some Results for Repairable Systems with General Repair, *Journal of Applied Probability*, Vol. 26, pp 89-102, 1989.
11. Kijima, M. and Sumita, N. "A useful generalization of renewal theory: counting process governed by non-negative Markovian increments." *Journal of Applied Probability,* 23, 71–88, 1986.
12. Kaminskiy, M. and Krivtsov, V. "A Monte Carlo approach to repairable system reliability analysis." *Probabilistic Safety Assessment and Managemen*t, New York: Springer; p. 1063–1068, 1998.
13. A. Mettas, W. Zhao, "Modeling and Analysis of Repairable Systems with General Repair", *Proceedings of the Annual Reliability and Maintainability Symposium(RAMS)*, 2005.
14. H. Ascher, H. Feingold, *Repairable Systems Reliability: Modeling, Inference, Misconceptions and Their Causes,* 1984; Marcel Dekker.
15. J. Zuo, W. Meeker, H. Wu, "Analysis of Window-Observation Recurrence Data", *Joint Statistical Meeting, JSM'05*, Minneapolis, Aug 2005.
16. J. Glosup, "Detecting Multiple Populations within a Collection of Repairable Systems", *Joint Statistical Meeting*, Toronto, 2004.
17. W. Heavlin, Identification of Anomalous machines using CMH statistic, *Sun Microsystems Internal Report*
18. Trindade, D.C., "An APL Program to Numerically Differentiate Data", IBM TR Report 19.0361, January 12, 1975