International Journal of Applied Research 2017; 3(2): 428-435

**ISSN Print:** 2394-7500
**ISSN Online:** 2394-5869
**Impact Factor:** 5.2
IJAR 2017; 3(2): 428-435
www.allresearchjournal.com
Received: 08-11-2016
Accepted: 09-12-2016

# Cost benefit and availability analysis for customized application S/W for insurance sector

**Navneet Malik, Dr. Naresh Kumar and Dr. Pardeep Goel**

**Navneet Malik**
Research scholar, Dept. of CSS, Mewar University Rajasthan, Rajasthan, India

**Dr. Naresh Kumar**
Assistant Professor, Dept. of Mathematics, Mewar University Rajasthan, Rajasthan, India

**Dr. Pardeep Goel**
Visiting Faculty, Mewar University Rajasthan, Rajasthan, India

**Correspondence**
**Navneet Malik**
Research scholar, Dept. of CSS, Mewar University Rajasthan, Rajasthan, India

## Abstract

This paper discusses the Behavioral and Availability Analysis of Customized Application Software against the standard ERP for the insurance sector. This application software contains different modules such as policy management, claim management, customer management and profit management. Customized Application software can work in full working and reduced working capacity. Reduced working capacity means from the given set of modules one or more than one may not be working as expected. This is a single Application Software model which can work in reduced state which means the software has old features and needs adaptive, corrective and preventive maintenance to reach the full capacity working state.So two types of failures state for application software are there; partially failed and completely failed states. This model consists of single Application Software ‘A’. It can work in reduced capacity. The Application Software ‘A’ can fail partially and hence can be in up state. Partially failed state means any module is not working and totally failed state which means that customized application software is not working due to multiple module failures or hardware failures. The module failure can be due to S/W or H/W. The application software can work with reduced capacity in a partially failed state. There is a single maintenance engineer available for both types of failures. This customized application software system is discussed for fixed state environments. The performance of system can be measured by Regenerative Point Graphical Technique (RPGT).

**Keywords:** Reliability, Availability, Base-State, RPGT, MTSF, Software Failure, Hardware Failure, Probability, Module

## 1. Introduction

Availability is the most important factor for all the organizations. With the speedy development of information technology, all industries are using software in one or other way to achieve the reliability. Enterprise Resource Planning (ERP) systems are used by small and medium sized companies. Standard ERP is less used as compared to customized ERP as more benefits are there in the customized ERP [1]. Availability of customized application software can be enhanced by maintenance and inspection. Standard ERP may not be working with full functionality at a particular instant of time then customization can be considered as a form of maintenance [2]. Different triggers for customization are categorized as technology change, organization change and external environment [3].This level of customization is used to decide when the application software is to be declared in the unaccepted state which helps the management to take the decision whether to replace or upgrade the existing application software with customized ERP.

Availability analysis of various process industries and software has been done by a number of researchers Cao, Jinhva and Wu, Yanhong [4] evaluated Reliability Analysis with Repair facility, Chander, S. and Bansal, R. K. [5] discussed Profit analysis with repair at different failure modes, Malik S. C., Chand, P. and Singh, J. [6] presented the availability analysis of dispersed hardware and software systems. Gupta, V. K. Singh J., &Vanita [7] discussed the Reliability Analysis by introducing the base state, Fukuta [8] discussed the reliability, Osaki [9] discussed about redundant system and Chung [10] discussed the repairable and non-repairable system, Gupta V.K [11] and Jindal [12] discussed the RPGT technique for reliability analysis.

Discussion on ERP and customized ERP is done by different researchers such as Colette Rolland & Naveen Prakash [13] discussed the gap between organizational requirement and ERP, Sudhaman Parthasarathy and Maya Daneva [14] discussed the customization by giving the priority to requirement, Robert C. Beatty and Craig DWilliams [15] discussed ERP upgrading, Sudhaman Parthasarathya, Srinarayan Sharma [14] discussed the productivity of ERP package from customization point of view, Helmut Klaus [16] and E. M. Shehab [17] discussed general ERP. Analysis has been carried out by the researchers for single and multi units such as Barlow R. E. Barlow*et al.* [18] discussed different reliability analysis factors for single unit, R. Natarajan [19] discussed the need of repair and spares to achieve the given availability, A Kumar *et al.* [20] discussed the reliability model for single unit using preventive maintenance. A. Kumar *et al.* [21] proposed a model for the reliability of computer system by considering independent software and hardware failure. A. Kanso *et al.* [22] and M. Jain *et al.* [23] discussed a highly available system by using redundancy. In this paper, the cost and availability analysis of single Application software system, which can work in reduced capacity is done. The mean time to system failure, availability and other key parameters of the software are estimated using the *Regenerative Point Graphical Technique (RPGT)*

In view of the above, in this paper, there is single Application Software model for insurance sector containing multiple modules such as agent management, policy management, customer management and claim and profit management. Application software can work in reduced capacity after failure of a module instead of completely failed. Thus there are two types of failures: Partially failed and completely failed. The software system consists of Application software ‘A’ which can work in reduced state after failure. The Application Software ‘A’ can work partially and hence can be in up state, partially failed state (reduced state) or completely failed state. The software system can work with reduced capacity in a partially failed state. There is a single maintenance engineer for both type of failure. Repairs are perfect. The application software is down if all software modules are failed completely. The distributions of the repair times are exponential and are different for both types of failuresi.e. Software and hardware failure. The structure is for steady state environments. By the *Regenerative Point Graphical Technique (RPGT)* the following system characteristics have been evaluated to study the system performance.

i Availability
ii The busy period of the maintenance engineer doing any given job.
iii The number of the maintenance engineer’s visits.

Tables are there to represent the performance of the MTSF and Steady state Availability of the software for a particular case.

**States and Notations:** Following are the states possible at any instant of time:

$S_0$ = A    $S_1$ = A(H)
$S_2$ =A′    $S_3$ = a
$S_4$ = A′(H)

The regenerative states are $S_0$, $S_1$, $S_2$, $S_3$, $S_4$.

The transition diagram for all these states is shown in Fig 1

Where $S_0$ is full capacity working state.

Where $S_1$ is a failure state attained due to hardware failure from full capacity working state.

Where $S_2$ is a failure state attained due to any module failure from full capacity working state.

Where $S_3$ is a failure state attained due to further module failure from partial capacity working state.

Where $S_4$ is a failure state attained due to hardware failure from partial capacity working state.

## 1.2Transition Diagram ofthe System

Using the above assumptions, the transition diagram is shown in Fig 1

**Table 1:** Notations of Transition Diagram

| State | Sign |
|---|---|
| Regenerative state | ● |
| Working-state | ○ |
| Failed state | ▭ |
| Reduced/partial working state | ⬭ |

## Transition Diagram

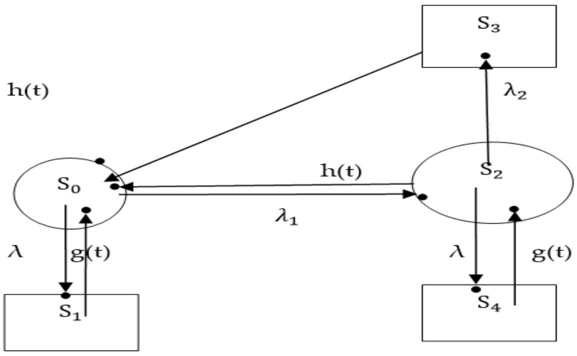

**Fig 1:** Transition Diagram

### 1.3 Evaluation parameters of the system

The significant parameters of the software system are designed by finding the 'base-state' and applying RPGT. The MTSF is determined w.r.t. the initial state '0' and the other parameters are obtained by using base-state.

### Transition Probabilities

**$q_{x,y}(t)$:** It represent the probability density function of a regenerative state x to a regenerative state y or to a failed state y without visiting any other regenerative state in (0,t].

**$p_{x,y}$:** It represent the transition probability(steady state) from a regenerative state x to a regenerative state y without visiting any other regenerative state. $p_{x,y} = q^*_{x,y}(0)$; where $*$ denotes Laplace transformation.

**Table 2:** Transition Probabilities

| $\mathbf{q_{x,y}(t)}$ | $\mathbf{p_{x,y} = q^*_{x,y}(0)}$ |
|---|---|
| $q_{0,1}(t) = \lambda e^{-(\lambda+\lambda_1)t}$<br>$q_{0,2}(t) = \lambda_1 e^{-(\lambda+\lambda_1)t}$ | $p_{0,1} = \frac{\lambda}{\lambda+\lambda_1}$<br>$p_{0,2} = \frac{\lambda_1}{\lambda+\lambda_1}$ |
| $q_{1,0}(t) = g(t)$ | $p_{1,0} = g^*(0)$ |
| $q_{2,0}(t) = h(t)e^{-(\lambda+\lambda_2)t}$<br>$q_{2,3}(t) = \lambda_2 e^{-(\lambda+\lambda_2)t}\bar{H}(t)$<br>$q_{2,4}(t) = \lambda e^{-(\lambda+\lambda_2)t}\bar{H}(t)$ | $p_{2,0} = h^*(\lambda+\lambda_2)$<br>$p_{2,3}(t) = \frac{\lambda_2}{\lambda+\lambda_2}\bar{H}(t)$<br>Where<br>$\bar{H}(t) = \frac{\lambda_2}{\lambda+\lambda_2}\,(1-h^*(\lambda+\lambda_2))$<br>$p_{2,4}(t) = \frac{\lambda}{\lambda+\lambda_2}\bar{H}(t)$ |
| $q_{3,0}(t) = h(t)$ | $p_{3,0}(t) = h^*(0) = 1$ |
| $q_{4,2}(t) = g(t)$ | $p_{4,2}(t) = g^*(0)=1$ |

It can be easily verified that;

$p_{0,1} + p_{0,2}=1; p_{1,0}=1; p_{2,0} + p_{2,3} + p_{2,4}=1; h^*(0)=1;$

$p_{3,0} = h^*(0) = 1;\ p_{4,2}(t) = g^*(0)$

### Mean Sojourn Times

$R_x(t)$ : reliability of an application software at given time when the software is in regenerative state x.

$\mu_x$: mean time spanned in state x

$\mu_x = \int_0^\infty R_x(t)dt = R^*_x(0).$

**Table 3:** Mean Sojourn Times

| $\mathbf{R_x(t)}$ | $\boldsymbol{\mu_x = R^*_x(0).}$ |
|---|---|
| $R_0(t) = e^{-(\lambda+\lambda_1)t}$ | $\mu_0 = \frac{1}{\lambda+\lambda_1}$ |
| $R_1(t) = G(t)$ | $\mu_1 = g^{*\prime}(0)$ |
| $R_2(t) = e^{-(\lambda+\lambda_2)t}H(t)$ | $\mu_2 = \frac{1-h^*(\lambda+\lambda_2)}{\lambda+\lambda_2}$ |
| $R_3(t) = H(t)$ | $\mu_3 = -h^{*\prime}(0)$ |
| $R_4(t) = G(t)$ | $\mu_4 = -g^{*\prime}(0)$ |

### Evaluation of Parameters

Regenerative Point Graphical Technique (RPGT) is used to calculate the different performance parameters. All parameters are evaluated for the base state '0'.

Taking base state '0', following are the transition probabilities factors:

$$V_{0,0} = \left[(0,1,0) + \frac{(0,2,0)}{1-L_1} + \frac{(0,2,3,0)}{1-L_1}\right] = 1$$

$$V_{0,1} = (0,1) = p_{0,1}$$

$$V_{0,2} = \frac{(0,2)}{1\text{-}L_1}$$

$$V_{0,3} = \frac{(0,2,3)}{1-L_1} = \frac{p_{0,2}p_{2,3}}{1-L_1}$$

$$V_{0,4} = \frac{(0,2,4)}{1-L_1} = \frac{p_{0,2}p_{2,4}}{1-L_1}$$

Where

$$1 - L_1 = 1 - \{2,4,2\} = 1 - p_{2,4}p_{4,2}$$

**(a). Availability of the software system:** The states in which the application software is available are represented by y = 0,2. Total regenerative states as per transition diagram are x = 0 to 4. For ‘θ’ = ‘0’, the total span of time for which the system is available is given by:

$$A_0 = \left[\sum_{y,s_r}\left\{\frac{\{pr(\theta \overset{s_r}{\rightarrow} y)\}f_y.\mu_y}{\prod_{m_1 \neq \theta}\{1 - V_{m_1,m_1}\}}\right\}\right] \div \left[\sum_{x,s_r}\left\{\frac{\{pr(\theta \overset{s_r}{\rightarrow} x)\}.\mu_x^1}{\prod_{m_2 \neq \theta}\{1 - V_{m_2,m_2}\}}\right\}\right]$$

$$A_0 = \left[\sum_y V_{\theta,y}.f_y.\mu_y\right] \div \left[\sum_x V_{\theta,x}.\mu_x^1\right]$$

$$= \left[V_{0,0}.f_0.\mu_0 + V_{0,2}.f_2.\mu_2\right] \div \left[V_{0,0}\mu_0^1 + V_{0,1}\mu_1^1 + V_{0,2}\mu_2^1 + V_{0,3}\mu_3^1 + V_{0,4}\mu_4^1\right]$$

$$= [f_0\mu_0 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}} f_2\mu_2] \div \left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$$

$$= N_0 \div D_0$$

Where,

$$N_0 = [f_0\mu_0 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}} f_2\mu_2]$$

$$D_0 = \left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$$

**(b). Busy period of the Maintenance engineer:** Total states whenmaintenance engineer is busy for adaptive, corrective or preventive maintenance are y =1, 2, 3, 4. Totalregenerative states for proposed model are x = 0 to 4.For ‘θ’ = ‘0’, the total duration of time for which the maintenance engineer is busy is given by:

$$B_0 = \left[\sum_{y,s_r}\left\{\frac{\{pr(\theta \overset{s_r}{\rightarrow} y)\}.\eta_y}{\prod_{m_1 \neq \theta}\{1 - V_{m_1,m_1}\}}\right\}\right] \div \left[\sum_{x,s_r}\left\{\frac{\{pr(\theta \overset{s_r}{\rightarrow} x)\}.\mu_x^1}{\prod_{m_2 \neq \theta}\{1 - V_{m_2,m_2}\}}\right\}\right]$$

$$B_0 = \left[\sum_y V_{\theta,y}.\eta_y\right] \div \left[\sum_x V_{\theta,x}.\mu_x^1\right]$$

$$= \left[V_{0,1}.\eta_1 + V_{0,2}.\eta_2 + V_{0,3}.\eta_3 + V_{0,4}.\eta_4\right] \div \left[V_{0,0}\mu_0^1 + V_{0,1}\mu_1^1 + V_{0,2}\mu_2^1 + V_{0,3}\mu_3^1 + V_{0,4}\mu_4^1\right]$$

$$= [p_{0,1}\eta_1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\eta_2 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\eta_3 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\eta_4] \div \left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$$

$$= N_1 \div D_1$$

Where, $N_1 = [p_{0,1}\eta_1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\eta_2 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\eta_3 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\eta_4]$

$$D_1 = \left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$$

**(c). Expected number of Maintenance engineer’s visits:** It represent the regenerative states where the maintenance engineer visits for maintenance of the system. As per given fig 1 these states are y = 1, 2and the regenerative states are: x = 0 to 4. For ‘θ’ = ‘0’, the number of maintenance engineer’s visits is represented by:

$$V_0 = \left[\sum_{y,s_r}\left\{\frac{\{pr(\theta \overset{s_r}{\rightarrow} y)\}}{\prod_{m_1 \neq \theta}\{1 - V_{m_1,m_1}\}}\right\}\right] \div \left[\sum_{x,s_r}\left\{\frac{\{pr(\theta \overset{s_r}{\rightarrow} x)\}.\mu_x^1}{\prod_{m_2 \neq \theta}\{1 - V_{m_2,m_2}\}}\right\}\right]$$

$$V_0 = \left[\sum_y V_{\theta,y}\right] \div \left[\sum_x V_{\theta,x}.\mu_x^1\right]$$

$$= (V_{0,1} + V_{0,2}) \div \left[V_{0,0}\mu_0^1 + V_{0,1}\mu_1^1 + V_{0,2}\mu_2^1 + V_{0,3}\mu_3^1 + V_{0,4}\mu_4^1\right]$$

$$= [p_{0,1} + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}] \div$$

$$\left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$$

$V_0 = N_2 \div D_2$

$N_2 = p_{0,1} + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}$

$$D_2 = \left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$$

### 1.4 Special Case

Considering the values:

$g(t)=\omega_1 e^{-\omega_1 t}, h(t)=\omega_2 e^{-\omega_2 t}$

Assuming $\omega_1=\omega_2=\omega$

$p_{0,1}=\frac{\lambda}{\lambda+\lambda_1}$, $p_{0,2}=\frac{\lambda_1}{\lambda+\lambda_1}$, $p_{1,0}=1$, $p_{2,0}=\frac{\omega}{w+\lambda+\lambda_2}$,

$$p_{2,3}=\frac{\lambda_2}{\lambda+\lambda_2} - \frac{\lambda_2}{\lambda+\lambda_2} * \frac{\omega}{\omega+\lambda+\lambda_2}$$

$$p_{2,4}=\frac{\lambda}{\lambda+\lambda_2} - \frac{\lambda}{\lambda+\lambda_2} * \frac{\omega}{\omega+\lambda+\lambda_2},\ p_{3,0}=1, p_{4,2}=1$$

$$\mu_0=\frac{1}{\lambda+\lambda_1}, \mu_1=\frac{1}{\omega}\quad, \mu_2=\frac{1}{\omega+\lambda+\lambda_2},\ \mu_3=\frac{1}{\omega},\ \mu_4=\frac{1}{\omega}$$

Following factors can be calculated by above values:

Availability $(A_0) = [V_{0,0}.f_0.\mu_0 + V_{0,2}.f_2.\mu_2] \div [V_{0,0}\mu_0^1 + V_{0,1}\mu_1^1 + V_{0,2}\mu_2^1 + V_{0,3}\mu_3^1 + V_{0,4}\mu_4^1]$

Availability $= [f_0\mu_0 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}} f_2\mu_2] \div \left[\mu_0^1 + p_{0,1}\mu_1^1 + \frac{p_{0,2}}{1-p_{2,4}p_{4,2}}\mu_2^1 + \frac{p_{0,2}p_{2,3}}{1-p_{2,4}p_{4,2}}\mu_3^1 + \frac{p_{0,2}p_{2,4}}{1-p_{2,4}p_{4,2}}\mu_4^1\right]$

### 1.5Analytical Discussion for Availability's Repair Rate

Considering different values of the repair rates (ω) = 0.80, 0.85, 0.90, 0.95 and 1.0 the Availability of the software system was calculated. For different Failure Rates ($\lambda_1$) such as$\lambda_1$ = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.01 the availability was calculated and shown in Table 4.

**Table 4:** Availability vs Failure and Repair Rate

| $\lambda_1$ | $A_0$ (ω=0.80) | $A_0$ (ω=0.85) | $A_0$ (ω=0.90) | $A_0$ (ω=0.95) | $A_0$ (ω=1) |
|---|---|---|---|---|---|
| 0.005 | 0.998740 | 0.998815 | 0.998881 | 0.998940 | 0.998994 |
| 0.006 | 0.998738 | 0.998812 | 0.998879 | 0.998939 | 0.998992 |
| 0.007 | 0.998735 | 0.998810 | 0.998877 | 0.998937 | 0.998991 |
| 0.008 | 0.998733 | 0.998808 | 0.998875 | 0.998935 | 0.998989 |
| 0.009 | 0.998731 | 0.998806 | 0.998874 | 0.998934 | 0.998988 |
| 0.01 | 0.998728 | 0.998804 | 0.998872 | 0.998932 | 0.998986 |

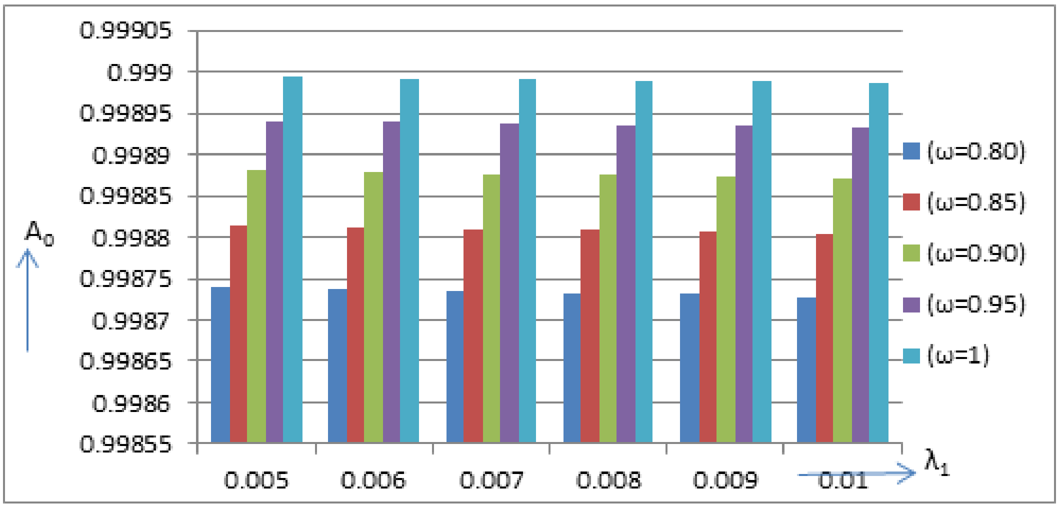

**Fig 2:** Availability vs Failure and Repair Rate

Availability ($A_0$) was calculated for different Repair Rates (ω), Failure Rates ($\lambda_1$) of
the module and given in Table 4 and can be represented with the graph as in Fig. 2.Itcan be concluded that as we increase the values of the Repair Rate (ω) the availability also increases. Further, it can be observed from the table that values of Availability ($A_0$) shows the expected trend for different values of Failure Rates. Availability decreases with the increase in the values of Failure Rates ($\lambda_1$).

### 1.6 Analytical Discussion for $B_0$vs Repair Rate (ω)

The busy period of repairman for the software is calculated for diverse values of the Failure Rates ($\lambda_1$) such as$\lambda_1$ = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.010 and for different repair rates (ω) such asω = 0.80, 0.85, 0.90, 0.95 and 1.0.Table 5 represent the calculated data.

**Table 5:** Busy Period of Maintenance Engineers Failure and Repair Rate

| $\lambda_1$ | $B_0$ (ω=0.80) | $B_0$ (ω=0.85) | $B_0$ (ω=0.90) | $B_0$ (ω=0.95) | $B_0$ (ω=1) |
|---|---|---|---|---|---|
| 0.005 | 0.0101239 | 0.0090675 | 0.0081815 | 0.0074312 | 0.0067901 |
| 0.006 | 0.0116817 | 0.0104515 | 0.0094197 | 0.0085458 | 0.0077992 |
| 0.007 | 0.0132297 | 0.0118265 | 0.0106493 | 0.0096523 | 0.0088003 |
| 0.008 | 0.0147700 | 0.0131943 | 0.0118723 | 0.0107524 | 0.0097955 |
| 0.009 | 0.0163037 | 0.0145561 | 0.0130898 | 0.0118476 | 0.0107859 |
| 0.10 | 0.0178317 | 0.0159128 | 0.0143027 | 0.0129384 | 0.0117724 |

Table 5 and Fig 3 represent the behaviour of busy period of repairman vs failure rates and repair rates. It is observed from the table and graphs that by increasing the failure ratesBusy period of repairman ($B_0$) increases. Taking the failure rate constant, for increased value of repair rate, busy period of repairman decreases.

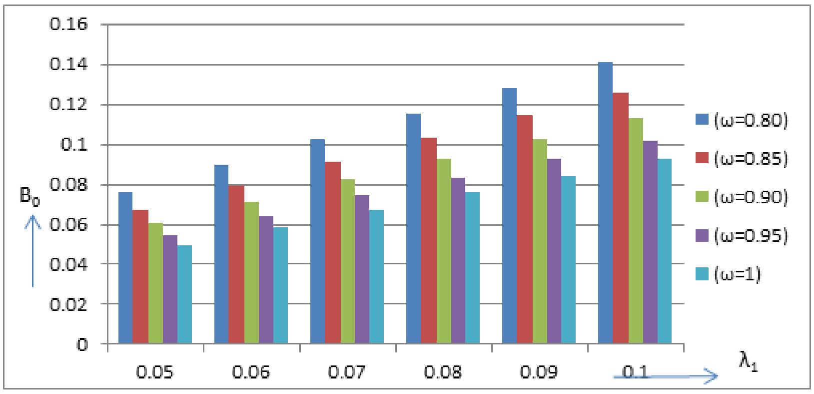

**Fig 3:** Busy Period of Maintenance Engineers Failure and Repair Rate

### 1.7 Analytical Discussion for Maintenance Engineer's Visits Repair Rate (ω)

The expected number of engineer visits can be calculated for different failure rates. Considering λ  = 0.001and $\lambda_2$= 0.002 the expected number of engineer visits are calculated. By taking $\lambda_1$ = 0.005, 0.006, 0.007, 0.008, 0.009 and 0.010 different number of engineer visits are calculated and shown in table 6.

**Table 6:** Maintenance Engineer Visits vs Failure and Repair Rate

| $\lambda_1$ | $V_0$ (ω=0.80) | $V_0$ (ω=0.85) | $V_0$ (ω=0.90) | $V_0$ (ω=0.95) | $V_0$ (ω=1) |
|---|---|---|---|---|---|
| 0.005 | .0084020 | .0084077 | .0084128 | .0084173 | .0084214 |
| 0.006 | .0098644 | .0098723 | .0098793 | .0098856 | .0098912 |
| 0.007 | .0113208 | .0113312 | .0113404 | .0113487 | .0113562 |
| 0.008 | .0127712 | .0127845 | .0127963 | .0128068 | .0128163 |
| 0.009 | .0142158 | .0142322 | .0142468 | .0142599 | .0142717 |
| .010 | .0156544 | .0156743 | .0156920 | .0157079 | .0157223 |

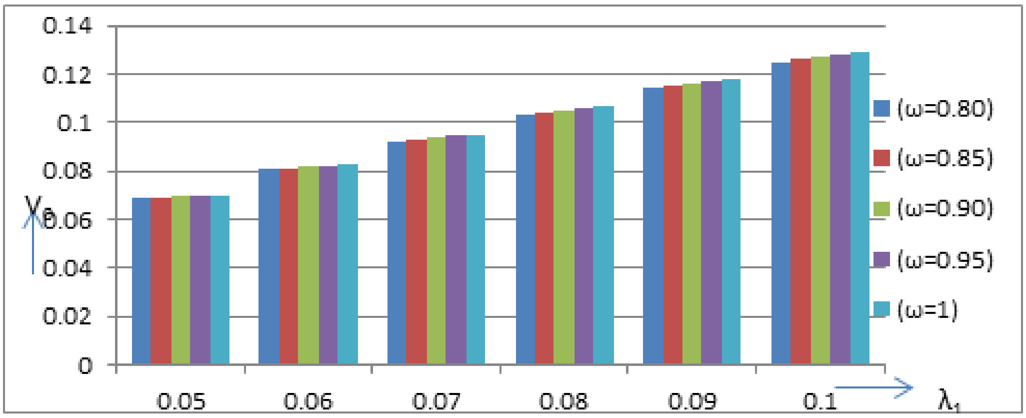

**Fig 4:** Repairman Visits vs Failure and Repair Rate

From the table and graph, it can be observed that with the increase in repair rate the number of visits increases, and if failure rate increases it also leads to increase in the number of repairman visits.

### 1.8 Cost Calculation

Maintenance cost is a major contributor to the software cost.Cost factor associated with downtime is more than 40 per cent of total cost provided to the owner of the software [24]. Many companies depend upon outsourcing for software maintenance [25]. This profit function can be used in software industry in contract maturity. The methods used for cost estimation are categorised into three types as expert judgment, algorithmic estimation and analogy based estimation [26]. The net profit can be calculated by the difference of revenue generated and cost involved in the number of maintenance engineer's visits and busy period of maintenance engineer.

$P(t) = RA_0 - C_1B_0 - C_2V_0$

Where R = revenue per unit when software is available.

$C_2$ = cost per visit of the maintenance engineer.

$C_1$ = cost involved for busy time ofmaintenance engineer.

Considering the fixed values for R, $C_1$and $C_2$ the net profit can be calculated from the software system. $A_0$= Availability of software and $V_0$= Number of maintenance engineervisits $B_0$= busy period of maintenance engineer.

**Special case:** Assuming the values for R = 1000, $C_1$=20000, $C_2$=50000 and$\lambda_2$= 0.002, $\lambda$= 0.001;

**Table 7:** Profit Earned vs Failure and Repair Rate

| $\lambda_1$ | P(t) (ω=0.80) | P(t) (ω=0.85) | P(t) (ω=0.90) | P(t) (ω=0.95) | P(t) (ω=1) |
|---|---|---|---|---|---|
| 0.005 | 376 | 397 | 415 | 429 | 442 |
| 0.006 | 271 | 296 | 317 | 334 | 348 |
| 0.007 | 168 | 196 | 219 | 238 | 255 |
| 0.008 | 65 | 96 | 122 | 144 | 162 |
| 0.009 | -38 | -3 | 25 | 49 | 70 |
| 0.010 | -141 | -103 | -72 | -45 | -25 |

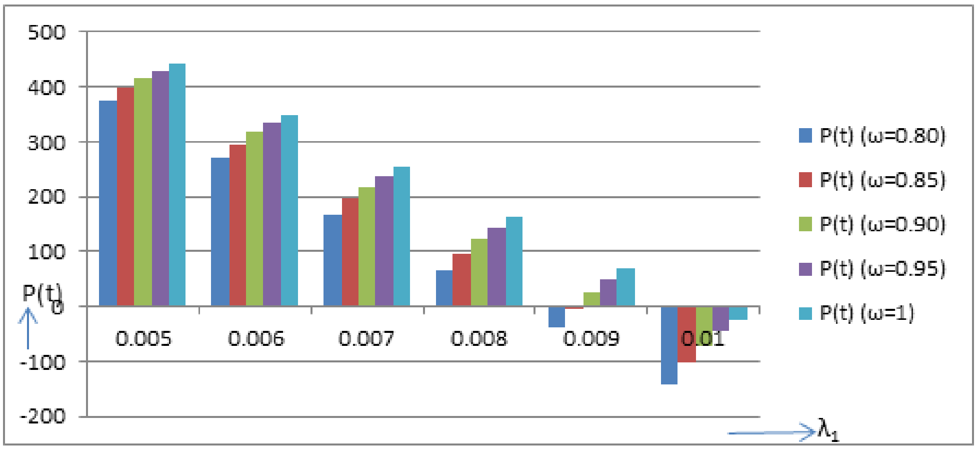

**Fig 5:** Profit Earned vs Failure and Repair Rate

### 1.9 Conclusion

From the Tables and graphs, we see that Availability of the application software for insurance is dependent on maintenance rate. As we increase the maintenance rate (ω) the Availability also increases. After so many modifications when profit is decreasing and is shifting from positive to negative, management can decide to one- time upgrade or shift to customize ERP.The model software system can be extended for two or more application software and with perfective maintenance. Failure rate and Repair rate can be treated as variables. Results attained can be used for cost and benefit analysis. Any state can be taken as the Base-state to evaluate the various parameters. The study can also be extended for time-dependent cases.

## 2. References

1. Khaleel Y, Abuhamdah A, Sara MA, Al-Tamimi B. Components and Analysis Method of Enterprise Resource Planning (ERP) Requirements in Small and Medium Enterprises (SMEs)," International Journal of Electrical and Computer Engineering (IJECE). 2016; 6:682-689.
2. Light B. Going beyond 'misfit'as a reason for ERP package customisation, Computers in Industry. 2005; 56:606-619.
3. Mackay WE. Triggers and barriers to customizing software," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1991, 153-160.
4. Cao J, Wu Y. Reliability Analysis of A Two-unit Cold Standby System With Replaceable Repair facility, Microelectronics, Reliab. 1989; 29(2):145-150.
5. Chander S, Bansal R. Profit Analysis of Single-Unit Reliability Models with Repair at Different Failures Modes, Proc. Increase IIT Kharagpur, India, 2005, 577-587.

6. Malik SC, Chand P, Singh J. Reliability and Profit Evaluation of an Operating System with Different Repair Strategy Subject to Degradation, ed: JMASS, 2008.
7. Gupta VK, Singh J, Vanita. The New Concept of a Base-State in the Reliability Analysis, Proce. Of National Conference; JMASS, 2010, 6.
8. Fukuta J, Kodama M. Mission reliability for a redundant repairable system with two dissimilar units, IEEE Transactions on Reliability. 1974; 4:280-282.
9. Osaki S. On a Two-Unit Standby-Redundant System with Imperfect Switchover, IEEE Transactions on Reliability. 1972; 1:20-24.
10. Chung WK. Reliability Analysis of Repairable and Non-Repairable Systems with Common-Cause Failures, Microelectronics Reliability. 1989; 29:545-547.
11. Gupta VK, Kumar, Kuldeep Singh J, Goel Pardeep. Profit Analysis of A Single Unit Operating System With a Capacity Factor and Undergoing Degradation, Proce. Of National Conference, JMASS, 2009, 5.
12. Jindal G, Goel P, Gupta VK, Singh J. Availability and Behavioral Analysis of a Single Unit Redundant System having Perfect Switch-over Device, Proc. Of International Conference on Emerging Issues and Challenges in Higher Education, 2011.
13. Rolland C, Prakash N. Bridging the gap between organisational needs and ERP functionality, Requirements Engineering. 2000; 5:180-193.
14. Parthasarathy S, Daneva M. An approach to estimation of degree of customization for ERP projects using prioritized requirements, Journal of Systems and Software. 2016; 117:471-487.
15. Beatty RC, Williams CD. ERP II: best practices for successfully implementing an ERP upgrade, Communications of the ACM. 2006; 49:105-109.
16. Klaus H, Rosemann M, Gable GG. What is ERP? Information systems frontiers. 2000; 2:141-162.
17. Shehab E, Sharp M, Supramaniam L, Spedding TA. Enterprise resource planning: An integrative review, Business Process Management Journal. 2004; 10:359-386.
18. Barlow RE, Hunter LC. Reliability analysis of a one-unit system, Operations research. 1961; 9:200-208.
19. Natarajan R. A Reliability Problem with Spares and Multiple Repair Facilities, Operations Research. 1968; 16:1041-1057.
20. Kumar A, Saini M. Cost-Benefit Analysis of a Single-Unit System with Preventive Maintenance and Weibull Distribution for Failure and Repair Activities, Journal of Applied Mathematics, Statistics and Informatics. 2014; 10:5-19.
21. Kumar A, Malik S. Reliability Modelling of a Computer System with Priority to H/W Repair over Replacement of H/W and Up-gradation of S/W Subject to MOT and MRT," Jordan Journal of Mechanical & Industrial Engineering. 2014, 8.
22. Kanso A, Toeroe M, Khendek F. Comparing Redundancy Models for High Availability Middleware, Computing. 2014; 96:975-993.
23. Jain M, Rani S. Transient analysis of hardware and software systems with warm standbys and switching failures, International Journal of Mathematics in Operational Research. 2014; 6:1-28.
24. Schroeder B, Gibson GA. The Computer Failure Data Repository (CFDR)," Workshop on Reliability Analysis of System Failure Data (RAF'07) MSR Cambridge, UK, 2007.
25. Lee BC, Rhew SY. An Empirical Study on Adjustment Factors to Estimate Maintenance Cost of Applications Developed Using Components, Lecture Notes on Software Engineering, 2014, 2.
26. Goh YFLMXTN. A study of the non-linear adjustment for analogy basedsoftware cost estimation, Empir Software Eng 12, 2009.