An optimal software release model has been analyzed with the goal that total cost of software cannot exceed a given budget and requirements, for finding the maximum reliability of the software products. In this chapter, we first develop the non-homogeneous software error detection model to study the software reliability growth. The total expected cost of the software along with repair cost has been evaluated. The optimal release policies are also discussed from the manufacturer viewpoint to know the optimal release time of the software. By taking numerical examples, assumed related data sets are illustrated in order to facilitate the sensitivity analysis.

10.1 Introduction

With the advances in computer technology, the reliable software is greatly needed today than ever before. The performance measures of software based on reliability theory give us confidence about the effectiveness of the completed software. A company cannot achieve success if its systems are outdated, unreliable and unavailable. There is no doubt that the success of any innovative program depends upon the reliable and updated software system. Therefore in order to maintain the reputation and remain competitive, the software companies are looking for more reliable software to improve the quality of the systems.

Software testing has a great impact on the total software development cost. For repairing the detected software errors, the test resources such as manpower, time, tools and equipments are required. If any software is released in the market for operational use without perfect testing, the maintenance cost of it is increased due to its frequent failure. On the other hand, longer testing time increases the reliability but delays the release time as such the cost is also increased. An optimal software release time in this situation is a matter of vital importance to the software developer.

A number of NHPP software reliability growth models has been appeared in literature. Yamada et al. (1983) described S-shaped software reliability growth model for software error detection. Zhang and Pham (2006) predicted software failure rate

In recent years, the cost of developing software has become the major expenses in any computerized system. Leung (1992b) discussed the optimal software time with a given cost budget. Pillai and Nair (1997) described a model for software development effort and cost. Some researchers estimated a software cost model with error removal time, warranty cost and risk cost (cf. Zhang and Pham, 1998a; Zhang and Pham, 1998b). Pham (2006) analyzed the non-homogeneous Poisson process software reliability and cost models.

Various researchers have suggested the optimal software release policies considering reliability and cost factors. Yamada and Osaki (1986) proposed optimal software release policies for a non-homogeneous software error detection rate model. Again they (1987) gave optimal release policies with simultaneous cost and reliability requirements. Xie and Hong (1998) studied the sensitivity of software release time. An optimal release time for software considering cost, testing-effort and testing efficiency was considered by Huang and Lyu (2005).

Many software reliability growth models have been proposed to measure the time dependent behavior of fault isolation/detection/removal process. In this chapter, we develop software reliability growth model for a system with learning process. The rest of the chapter is organized into various sections as follows. Section 10.2 is devoted to the assumptions and notations related to the model. In section 10.3, we obtain the expected total cost of the software introducing repair cost. Optimal software release policies are discussed in section 10.4. Numerical results by taking illustration based on assumed data set are given in section 10.5. The concluding remarks related to model developed are given in section 10.6.

10.2 Model Description

We consider NHPP model for a software system subject to software failures caused by software faults in the system. The software fault detection process is assumed to follow a time dependent NHPP. According to model, the testers become
familiar with the software products, environments and software specifications. The model has the following distinct features:

- The failure detection phenomenon is modeled by NHPP.
- All faults in a program are mutually independent from the failure detection process.
- Fault detection rate is a learning function and it is expected that the learning process will grow with time.
- The error content function of the software is a random variable.
- The time between \((i-1)^{th}\) and \(i^{th}\) failures depends on the time to the \((i-1)^{th}\) failure.
- The probability of failure detection at any time is proportional to the current number of faults in the software. The proportionality of failure detection is constant.
- During the fault isolation/detection, no new error is introduced into the system.

Some notations used to describe the model are as follows:

\[
\begin{align*}
m(t) &: \text{ mean value function.} \\
a &: \text{ total number of errors in the software including the initial and introduced errors.} \\
b(t) &: \text{ error detection rate.} \\
\lambda(t) &: \text{ failure intensity function.} \\
C_0 &: \text{ setup cost of the software.} \\
C_1 &: \text{ testing cost of the software per unit time.} \\
C_2 &: \text{ cost per unit time of removing an error during testing.} \\
C_3 &: \text{ cost per unit time of removing an error detected during warranty period.} \\
C_4 &: \text{ cost associated with the loss due to the software failure.} \\
C_5(t) &: \text{ repair cost function.} \\
E\{C(t)\} &: \text{ total expected cost of the software.} \\
T &: \text{ testing time of the software.}
\end{align*}
\]
Chapter-10: NHPP Error Detection Software

\( T_w \): duration of warranty period.
\( T^* \): optimal release time of the software.
\( R(x/T) \): reliability function, where \( x \) is the operating time of the software.
\( r \): market interest rate.
\( \upsilon \): cost of rectification.
\( u_y \): expected time for removing an error during the testing period.
\( u_w \): expected time for removing an error during the warranty period.

Under these assumptions the SRGM based on NHPP can be formulated as

\[
\frac{dm(t)}{dt} = b(t)[a - m(t)] \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (10.1)
\]

where \( b(t) = \frac{b^2 t}{1 + bt} \).

On solving equation (10.1), the mean value function can be obtained as

\[
m(t) = a\left[1 - (1 + bt)e^{-bt}\right] \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (10.2)
\]

The failure intensity function is given by

\[
\lambda(t) = \frac{dm(t)}{dt} = ab^2te^{-bt}
\]

The corresponding reliability is,

\[
R\left(\frac{x}{t}\right) = \exp[-\{m(t + x) - m(t)\}]
\]

\[
= \exp[- a\{(1 + bt)e^{-bt} - (1 + b(t + x)e^{-b(t+x)}\}]]
\]

\[
\equiv e^Z
\]

where \( Z = -a\{(1 + bt)e^{-bt} - (1 + b(t + x)e^{-b(t+x)}\})).\)
10.3 Expected Total Cost of the Software

We consider the software life-cycle length from the time when the testing time of the software starts. The cost of the software before release and the costs of removing errors before and after release are known as software cost factors. Also, it can be described as how much investment is required in software testing. The cost elements are taken by including setup cost, cost of removing errors during testing and warranty period and repair cost of detected errors.

Therefore, the expected total cost of the software can be formulated as

$$E\{C(T)\} = C_0 + C_1 T^\delta + C_2 \lambda(T) u_w + C_3 [\lambda(T + T_w) - \lambda(T)] u_w + C_4 [1 - R(x/T)] + C_5 (T)$$

where

$$C_5 (T) = \int_0^T C_g(C) dC.$$

Here discounting process is considered as exponentially. The term $T^\delta$ shows a time discounting process with the testing time $T$. $C_5 (T)$ is the expected cost of each rectification over the life-cycle of the software. The rectification can be a repair or replacement. We consider that the cost of rectification is distributed exponentially with exponential parameter $\nu$ and is obtained as

$$g(C) = \nu e^{-\nu C}$$

On putting this value in equation (5), we get

$$C_5 (T) = \frac{1}{\nu} \left[ 1 - e^{-\nu T} (1 + \nu T) \right]$$

Now, we formulate the optimization problem as

$$\text{Min } E\{C(T)\} = C_0 + C_1 \int_T^{T+T_w} e^{-\eta t} dt + C_2 \lambda(T) \int_0^T e^{-\eta t} dt + C_3 u_w \int_T^{T+T_w} \lambda(t) e^{-\eta t} dt + C_4 [1 - R(x/T)] + \frac{1}{\nu} \left[ 1 - e^{-\nu T} (1 + \nu T) \right]$$

subject to $R\left( \frac{x}{T + T_w} \right) > R_0$.  

202
Here, $R_0$ is the minimum requirement of the software reliability.

Differentiating equation (10.7) with respect to $T$, we get

$$
\frac{\partial E\{C(T)\}}{\partial T} = C_1 e^{-rT} \left[ e^{-rT_T} - 1 \right] + C_2 u_y b^2 T e^{-T(b+r)} + C_3 u_w a b^2 e^{-T(b+r)}
$$

$$
\left[ e^{-T_T} (T + T_w) - T \right] - C_4 a b^2 \left[ e^{-bT} \left\{ e^{-b} (T + x) - T \right\} z \right] + T v e^{-uT}
$$

$$
\equiv X(T) \quad \quad \quad \quad \quad \quad \quad \quad \quad (10.8)
$$

Again differentiating equation (8) with respect to $T$, we get

$$
\frac{\partial^2 E\{C(T)\}}{\partial T^2} = -r C_1 e^{-rT} \left[ e^{-rT_T} - 1 \right] + C_2 u_y b^2 e^{-T(b+r)} \left[ 1 - T(b + r) \right] + C_3 u_w a b^2 e^{-T(b+r)}
$$

$$
\left[ e^{-T} (b + r) [1 - (b + r)(T + T_w)] + T(b + r) - 1 \right] - C_4 a b^2 e^{-bT}
$$

$$
\left[ ab^2 e^{-bT} \left\{ e^{-bx} (T + x) - T \right\} + e^{-bx} (1 - b(T + x)) + bT - 1 \right] + v e^{-uT} \left[ 1 - vT \right]
$$

$$
\equiv Y(T) \quad \quad \quad \quad \quad \quad \quad \quad \quad (10.9)
$$

### 10.4 Optimal Software Release Policies

The purpose of the software developers is to determine an optimum software release time satisfying both cost and reliability requirements. They need to know when software testing should be stopped so that they can minimize the related testing cost and meet the requirements of the software quality. The optimal software release policies can be established by minimizing the total expected software cost under the specified constraint so that the achieved software reliability by software testing is not less than a reliability objective.

There are three cases for minimizing the expected total cost of the software:

(i) if $Y(0) \geq Z$,

(ii) if $Y(\infty) < Z$,

(iii) if $Y(0) < Z$, $Y(T) \leq Z$ for $T \in (0, T_0)$ and $Y(T) > Z$ for $T \in (T_0, \infty)$ where $T_0 = U^{-1}(C)$. 

203
Case (i) \( Y(0) \geq Z \): In this case
if \( X(0) \geq 0 \), then \( T^* = 0 \);
if \( X(\infty) < 0 \), then \( T^* = \infty \);
if \( X(0) < 0 \), \( X(T) < 0 \) for \( T \in (0, T_1) \) and \( X(T) > 0 \) for \( T \in (T_1, \infty) \),
then
\[ T^* = T_1, \text{ where } T_1 = X^{-1}(0). \]

Case (ii) \( Y(\infty) < Z \): In this case
if \( X(0) \leq 0 \), then \( T^* = \infty \);
if \( X(\infty) > 0 \), then \( T^* = 0 \);
if \( X(0) > 0 \), \( X(T) > 0 \) for \( T \in (0, T_2) \), and \( X(T) < 0 \) for \( T \in (T_2, \infty) \),
then
\[ T^* = 0, \quad \text{if } E\{C(0)\} \leq E\{C(\infty)\} \]
\[ T^* = \infty, \quad \text{if } E\{C(0)\} > E\{C(\infty)\}. \]

Case (iii) \( Y(0) < Z \), \( Y(T) \leq Z \) for \( T \in (0, T_0) \), and \( Y(T) > Z \) for \( T \in (T_0, \infty) \).
where \( T_0 = U^{-1}(C) \). In this case
if \( X(0) \geq 0 \), then
\[ T^* = 0 \text{ if } E\{C(0)\} \leq E\{C(T_b)\} \]
\[ T^* = T_b \text{ if } E\{C(0)\} > E\{C(T_b)\}, \text{ where } T_b = \inf \{T > T_a: X(T) > 0\} \]
if \( X(0) < 0 \), then
\[ T^* = T_b', \text{ where } T_b' = X^{-1}(0). \]

10.5 Numerical Results

In this section, we provide numerical illustration to calculate the expected cost
\( E\{C(T)\} \) and reliability \( R(T) \) by varying the introducing faults ‘a’
and error detection rate ‘b’. The effects of these parameters on the expected cost and reliability have been
examined for default parameters \( C_0 = 40, C_1 = 60, C_2 = 250, C_3 = 300, C_4 = 350, r = .9, T_w = 1, u_w = .5, u_y = .5, b = .005, x = 1, \nu = .5 \).
For exploring the effect of different cost elements, we consider the following sets:
Chapter-10: NHPP Error Detection Software

**Set I.** : $C_0=40$, $C_1=60$, $C_2=250$, $C_3=300$, $C_4=350$.

**Set II.** : $C_0=50$, $C_1=60$, $C_2=250$, $C_3=300$, $C_4=350$.

**Set III.** : $C_0=40$, $C_1=80$, $C_2=250$, $C_3=300$, $C_4=350$.

**Set IV.** : $C_0=40$, $C_1=60$, $C_2=280$, $C_3=300$, $C_4=350$.

**Set V.** : $C_0=40$, $C_1=60$, $C_2=250$, $C_3=330$, $C_4=350$.

**Set VI.** : $C_0=40$, $C_1=60$, $C_2=250$, $C_3=300$, $C_4=400$.

Table 10.1 displays the results for the expected cost for different values of warranty period ($T_w$) for different data sets of costs and fixed values of other parameters as $u_w=.5$ and $u_y=.5$. It is seen that the expected cost first decreases sharply and after some time period it increases smoothly. Tables 10.2 and 10.3 illustrate the software cost by varying the parameters $u_y$ and $u_w$, respectively. We notice the similar pattern of the software cost as seen in table 10.1.

Figs 10.1(i)-10.1(iv) depict the variation of the total expected cost $E\{C(T)\}$ with testing time $T$ by varying parameters $a$, $b$, $r$ and $\eta$, respectively. Figs 10.1(i)-10.1(iv) reveal that $E\{C(T)\}$ first sharply decreases and after sometime it grows with respect to testing time $T$. In figs 10.1(i) and 10.1(ii), we notice that the expected cost first decreases linearly up to $T=4$ and after that it increases sharply. Figs 10.1(iii) and 10.1(iv) show the similar pattern up to $T=5$ and after that the cost increases linearly.

Figs 10.2(i) and 10.2(ii) illustrate the pattern of the reliability of the software by varying parameters $a$ and $b$, respectively. From figs 10.2(i) and 10.2(ii), we notice that the reliability initially decreases and after some time ($T=20$) it increases and finally it become almost constant for both parameters $a$ and $b$.

Over all we conclude that the total expected cost first decreases for lower values of $T$, and then it goes on increasing for further increment in $T$ followed by the almost constant trend for the increased values of $T$. This shows the fact that the expected total cost is greatly affected by the testing time $T$. The reason behind the high value of $E\{C(T)\}$ in the beginning is the initial testing cost, which is taken significantly high. All the parameters signify that more money is required for the maintenance if the software reliability growth occurs in the operational phase also.

205
10.6 Conclusion

The detection of the fault and then repairing it are very critical for achieving the pre-specified goal of very high quality software products. Manufacturer needs to know how much money should be spend in the testing of the software so that they can launch the software in the market with minimum cost, high reliability and desirable quality. In this chapter, we have studied this issue by developing a NHPP model for software reliability. The expected total costs incurred on detection and repairs determined are further employed to suggest the optimal software release policies. By knowing the effects of various parameters on the cost function, the software managers can take more reasonable and realistic decisions for software testing and repair processes.
### Table 10.1: Expected costs of software for different value of $t_w$

<table>
<thead>
<tr>
<th>$u_w$</th>
<th>$u_y$</th>
<th>$t_w$</th>
<th>$t$</th>
<th>Set I</th>
<th>Set II</th>
<th>Set III</th>
<th>Set IV</th>
<th>Set V</th>
<th>Set VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>.5</td>
<td>.8</td>
<td>0</td>
<td>74.38</td>
<td>84.38</td>
<td>85.79</td>
<td>74.38</td>
<td>74.38</td>
<td>74.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.31</td>
<td>54.31</td>
<td>44.43</td>
<td>44.33</td>
<td>44.31</td>
<td>44.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.44</td>
<td>56.44</td>
<td>46.44</td>
<td>46.46</td>
<td>46.44</td>
<td>47.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.40</td>
<td>58.40</td>
<td>48.40</td>
<td>48.42</td>
<td>48.40</td>
<td>49.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.19</td>
<td>60.19</td>
<td>50.19</td>
<td>50.21</td>
<td>50.19</td>
<td>51.33</td>
</tr>
<tr>
<td>.5</td>
<td>.5</td>
<td>1.0</td>
<td>0</td>
<td>79.73</td>
<td>89.73</td>
<td>92.92</td>
<td>79.73</td>
<td>79.73</td>
<td>79.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.37</td>
<td>54.37</td>
<td>44.51</td>
<td>44.39</td>
<td>44.37</td>
<td>44.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.44</td>
<td>56.44</td>
<td>46.44</td>
<td>46.46</td>
<td>46.44</td>
<td>47.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.40</td>
<td>58.40</td>
<td>48.40</td>
<td>48.42</td>
<td>48.40</td>
<td>49.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.19</td>
<td>60.19</td>
<td>50.19</td>
<td>50.21</td>
<td>50.19</td>
<td>51.33</td>
</tr>
<tr>
<td>.5</td>
<td>.5</td>
<td>1.2</td>
<td>0</td>
<td>84.20</td>
<td>94.20</td>
<td>98.88</td>
<td>84.20</td>
<td>84.20</td>
<td>84.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.42</td>
<td>54.42</td>
<td>44.58</td>
<td>44.44</td>
<td>44.42</td>
<td>44.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.44</td>
<td>56.44</td>
<td>46.44</td>
<td>46.46</td>
<td>46.44</td>
<td>47.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.40</td>
<td>58.40</td>
<td>48.40</td>
<td>48.42</td>
<td>48.40</td>
<td>49.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.19</td>
<td>60.19</td>
<td>50.19</td>
<td>50.21</td>
<td>50.19</td>
<td>51.33</td>
</tr>
</tbody>
</table>

Table 10.1: Expected costs of software for different value of $t_w$. 
### Table 10.2: Expected costs of software for different value of $u_y$.

<table>
<thead>
<tr>
<th>$u_w$</th>
<th>$u_y$</th>
<th>$t_w$</th>
<th>t</th>
<th>Set I</th>
<th>Set II</th>
<th>Set III</th>
<th>Set IV</th>
<th>Set V</th>
<th>Set VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>.3</td>
<td>1</td>
<td>0</td>
<td>79.73</td>
<td>89.73</td>
<td>92.92</td>
<td>79.73</td>
<td>79.73</td>
<td>79.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.30</td>
<td>54.30</td>
<td>44.44</td>
<td>44.31</td>
<td>44.29</td>
<td>44.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.36</td>
<td>56.36</td>
<td>46.37</td>
<td>46.38</td>
<td>46.36</td>
<td>46.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.32</td>
<td>58.32</td>
<td>48.32</td>
<td>48.33</td>
<td>48.32</td>
<td>49.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.11</td>
<td>60.11</td>
<td>50.11</td>
<td>50.12</td>
<td>50.11</td>
<td>51.25</td>
</tr>
<tr>
<td>.5</td>
<td>.6</td>
<td>1</td>
<td>0</td>
<td>79.73</td>
<td>89.73</td>
<td>92.92</td>
<td>79.73</td>
<td>79.73</td>
<td>79.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.40</td>
<td>54.40</td>
<td>44.55</td>
<td>44.43</td>
<td>44.40</td>
<td>44.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.48</td>
<td>56.48</td>
<td>46.48</td>
<td>46.51</td>
<td>46.48</td>
<td>47.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.43</td>
<td>58.43</td>
<td>48.43</td>
<td>48.46</td>
<td>48.43</td>
<td>49.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.22</td>
<td>60.22</td>
<td>50.22</td>
<td>50.25</td>
<td>50.22</td>
<td>51.37</td>
</tr>
<tr>
<td>.5</td>
<td>.9</td>
<td>1</td>
<td>0</td>
<td>79.73</td>
<td>89.73</td>
<td>92.92</td>
<td>79.73</td>
<td>79.73</td>
<td>79.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.51</td>
<td>54.51</td>
<td>44.66</td>
<td>44.55</td>
<td>44.51</td>
<td>44.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.59</td>
<td>56.59</td>
<td>46.60</td>
<td>46.63</td>
<td>46.59</td>
<td>47.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.55</td>
<td>58.55</td>
<td>48.55</td>
<td>48.59</td>
<td>48.55</td>
<td>49.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.34</td>
<td>60.34</td>
<td>50.34</td>
<td>50.38</td>
<td>50.34</td>
<td>51.48</td>
</tr>
<tr>
<td>$u_w$</td>
<td>$u_y$</td>
<td>$t_w$</td>
<td>t</td>
<td>Set I</td>
<td>Set II</td>
<td>Set III</td>
<td>Set IV</td>
<td>Set V</td>
<td>Set VI</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>----</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>.4</td>
<td>.5</td>
<td>1</td>
<td>0</td>
<td>79.74</td>
<td>89.74</td>
<td>92.93</td>
<td>79.74</td>
<td>79.74</td>
<td>79.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.37</td>
<td>54.37</td>
<td>44.51</td>
<td>44.39</td>
<td>44.37</td>
<td>44.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.44</td>
<td>56.44</td>
<td>46.44</td>
<td>46.46</td>
<td>46.44</td>
<td>47.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.40</td>
<td>58.40</td>
<td>48.40</td>
<td>48.42</td>
<td>48.40</td>
<td>49.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.19</td>
<td>60.19</td>
<td>50.19</td>
<td>50.21</td>
<td>50.19</td>
<td>51.33</td>
</tr>
<tr>
<td>.6</td>
<td>.5</td>
<td>1</td>
<td>0</td>
<td>79.72</td>
<td>89.72</td>
<td>92.91</td>
<td>79.72</td>
<td>79.72</td>
<td>79.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.37</td>
<td>54.37</td>
<td>44.51</td>
<td>44.39</td>
<td>44.37</td>
<td>44.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.44</td>
<td>56.44</td>
<td>46.44</td>
<td>46.46</td>
<td>46.44</td>
<td>47.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.40</td>
<td>58.40</td>
<td>48.40</td>
<td>48.42</td>
<td>48.40</td>
<td>49.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.19</td>
<td>60.19</td>
<td>50.19</td>
<td>50.21</td>
<td>50.19</td>
<td>51.33</td>
</tr>
<tr>
<td>.8</td>
<td>.5</td>
<td>1</td>
<td>0</td>
<td>79.71</td>
<td>89.71</td>
<td>92.89</td>
<td>79.71</td>
<td>79.70</td>
<td>79.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>44.36</td>
<td>54.36</td>
<td>44.51</td>
<td>44.39</td>
<td>44.36</td>
<td>44.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>46.44</td>
<td>56.44</td>
<td>46.44</td>
<td>46.46</td>
<td>46.44</td>
<td>47.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>48.40</td>
<td>58.40</td>
<td>48.40</td>
<td>48.42</td>
<td>48.40</td>
<td>49.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>50.19</td>
<td>60.19</td>
<td>50.19</td>
<td>50.21</td>
<td>50.19</td>
<td>51.33</td>
</tr>
</tbody>
</table>

Table 10.3: Expected costs of software for different value of $u_w$. 
Fig. 10.1: Expected total cost $EC(T)$ Vs $T$ for different values of (i) $a$ (ii) $b$ (iii) $r$ (iv) $v$. 
Fig. 10.2: Reliability $R(T)$ Vs $T$ for different values (i) $a$ and (ii) $b$. 

(i) 

(ii)