Staffing Software Maintenance Projects with Multiple Service Levels in SLA

M. Ghorbalipoor
Computer Engineering Department
Amirkabir University of Technology
Tehran Iran
ghorbalipoor@aut.ac.ir

M. R. Meybodi
Computer Engineering Department
Amirkabir University of Technology
Tehran Iran
mmeybodi@aut.ac.ir

Abstract: *Today outsourcing a software system's maintenance is common in software industry. Software maintenance organizations need to estimate the size of software maintenance team, in order to make decisions about acceptance or rejection of a project. In this paper we present a method to estimate the optimal staffing needed for a project that has multiple service levels with different priority and response time. The priority and response time determined with a SLA. We use results of queuing theory to model the maintenance environment with non-preemptive priority M/M/c queue. We also show that response time of an arbitrary request can not be less than a specified time*

Keywords: Outsourcing, Software Maintenance, Non-preemptive Priority M/M/c Queue.

1. Introduction

Software maintenance refers to software engineering activities pertaining to non-essential enhancements in software (enhancive, also known as perfective), adapting the software due to changing environmental requirements (adaptive), and the removal or mitigation of faults (corrective) [1].

Outsourcing the software maintenance is common in software industry. Software maintenance is well recognized as the most expensive and, perhaps, the longest phase in a software development life cycle [2].

A large number of software companies provide outsourced support and maintenance services of legacy software systems and applications for organizations around the world [3] and earn much revenue from this activity.

SLA as a part of contract describes the minimum performance criteria a service provider promises to meet while delivering a service. It also contains some penalties that will take effect if performance falls below the promised agreement. These penalties may be the financial repayment in the short term or lose of the contract in long term.

We need to have estimations of the necessary staffing levels for doing a software maintenance project.

If we assume that maintenance requests that come from client organization to maintenance provider organization have no priority to each other and be serviced via FIFS, we let the SLA condition as follows. In A percent of times, the response time must be less than t units of time. If we let $a = A/100$, with mathematical notations we have:

$$P(S \le t) \ge a \quad (1)$$

While S is a random variable that shows the response time.

For maintenance requests with different types while each type has its own priority and response time, we express the SLA conditions as follows. For requests of type i, in A_i percent of times the response time must be less than t_i units of time. If we let $a_i = A_i/100$ then with mathematical notations we have:

$$P(S_i \le t_i) \ge a_i \quad (2)$$

While S_i is a random variable that shows the response time of a request of type i.
We assume that requests of the type i have non-preemptive priority over requests of type j whenever $i < j$.

2. Assumptions

For modelling the maintenance environment we assume that maintenance requests (requests of type i) are entered to service provider organization with Poisson distribution with mean l (l_i). We also assume that efficiency of all members of project is equal and the service times for all requests have been distributed exponentially with mean m^{-1}.
While the distribution of Poisson for requests arrival has been observed in some cases [4], but some case studies reject it [5]. Ramaswamy [6] argues that this assumption is valid for error correction and adaptation types of maintenance projects.
We assume that requests with a same priority are serviced via FIFS while the requests with upper priority are serviced before lower priority requests. In addition, after a request is chosen for servicing, we do not have permission to cut its service (non-preemptive priority).

3. Queue Models

3.1 $M/M/c$ Queue

The first M indicates that arrival time distribution of requests is exponential (Memory-less) and second M indicates that service time distribution of requests is also exponential and c shows the number of personnel in maintenance project team(or servers).
In this model we assume that all requests have same priority and serviced via FIFS (First Input First Service).
If the number of requests in maintenance system are less than or equal to number of servers/personnel, no request will wait in queue. In contrary, when the number of requests is greater than number of servers, requests have to wait in queue.
In $M/M/c$ queue the length of queue is assumed infinity which is confirmable with our maintenance environment, because we need only to satisfy the "Relation (1)" and number of requests in queue is not important for us. Notice that mean service time for any request is m^{-1} and thus the mean service rate for any server/person becomes m.
Fig. 1 shows the $M/M/c$ model for maintenance environment.

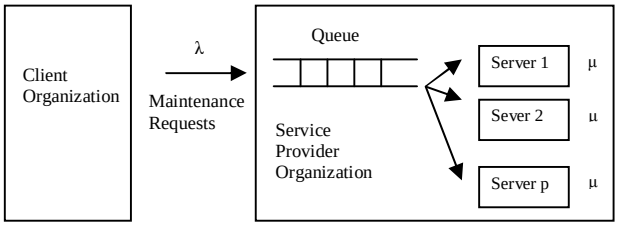

Fig. 1 M/M/c model for maintenance

Table 1 shows the list of symbols we use for modelling the maintenance environment as $M/M/c$.

Table 1: List of symbols for analyzing the $M/M/c$ queue

Symbol	Description
l	Mean rate that requests entering in the service provider organization
m	Mean rate that any server/person can service
c	Number of servers/personnel
$r = l/cm \le 1$	Traffic intensity of requests over servers
W	Steady-state waiting time of a request in queue.
B	Service time of a request
$S = W + B$	Steady-state sojourn time of a request in system
P_k	Probability of existence exactly k requests in steady-state system

We need to have $l < cm$ because otherwise we are not able to satisfy the SLA condition in "Relation (1)".
Probability that exactly n requests exist in system (waiting requests in queue +servicing requests) is equal to [7]:

$$P_0 = \frac{1}{\sum_{i=0}^{c-1} \frac{(r.c)^i}{i!} + \frac{(r.c)^c}{c!(1-r)}} \qquad r = l/cm$$

$$P_n = P_0 \frac{(r.c)^n}{n!} \qquad n = 1,\ldots,c \qquad (3)$$

$$P_n = P_0 \frac{r^n c^c}{c!} \qquad n = c, c+1,\ldots$$

Probability that an arbitrary request does not wait in queue is equal with probability that at most $c-1$ requests exists in system. It means that:

$$P(W=0)=P_0+P_1+...+P_{c-1}=\sum_{i=1}^{c-1} r_i \quad (4)$$

The probability that a new request has to wait is:

$$P(W>0)=P_c+P_{c+1}+...=P_0\frac{c^c}{c!}\sum_{i=c}^{\infty} r_i$$

$$=P_c\sum_{i=0}^{\infty} r^i=\frac{P_c}{1-r} \quad (5)$$

Computation of $P(W>0)$ through "Relation (5)" is not good way because we need to compute the factorial values and this decreases the computation speed when c is large. For an efficient computation of $P(W>0)$ we can use below relations [7]:

$$P(W>0)=\frac{r.B(c-1,c.r)}{1-r+r.B(c-1,c.r)} \quad (6)$$

$$B(m,r)=\frac{r.B(m-1,r)}{m+r.B(m-1,r)}\ m\geq 1 \quad (7)$$

$$B(0,r)=1$$

Distribution of waiting time and sojourn time of a request in $M/M/c$ queue is [7]:

$$P(W\leq t)=1-P(W>0)e^{-cm(1-r)t} \quad (8)$$

$$P(S\leq t)=1-e^{-m.t}+\frac{P(W>0)}{1-c+c.r}(e^{-m.t}-e^{-c.m(1-r)t}) \quad (9)$$

The "Relation (9)" is an important relation for us because it gives us the necessary tool for finding optimal staffing level for maintenance project that satisfies the SLA conditions.

If $c\to\infty$ then $P(W>0)\to 0$ and $e^{-c.m(1-r)t}\to 0$, with attention to "Relation (9)" we have:

$$\lim_{c\to\infty} P(S\leq t)=1-e^{-m.t}$$

$P(S\leq t)$ is a decreasing function with respect to c, so for great values of c we will have $P(S\leq t)<1-e^{-m.t}$ and hence

$$e^{-m.t}<1-P(S\leq t))$$

$$\Rightarrow t>\frac{\ln(1-P(S\leq t))}{-m} \quad (10)$$

The "Relation (10)" indicates to an important result:

"Response time of a request can not be less than a specified value, without attention to number of servers (personnel)".

Suppose the SLA condition be defined as "Relation (1)" with $P(S\leq t^*)\geq a^*$.From this we have:

$$1-P(S\leq t^*)\leq 1-a^*$$

$$\Rightarrow \frac{\ln(1-P(S\leq t^*))}{-m}\geq\frac{\ln(1-a^*)}{-m}$$

Combination with "Relation (10)" results that $t^*\geq\frac{\ln(1-a^*)}{-m}$. So we have the below corollary:

"The necessary condition for $P(S\leq t^*)\geq a^*$ to be meaningful is that we have $t^*\geq\frac{\ln(1-a^*)}{-m}$"

Suppose S_1 be the sojourn time of a request in $M/M/c$ queue with parameters l and m, and S_2 be the sojourn time of a request in $M/M/2c$ queue with parameters $2l$ and m. We would have

$$P(S_2\leq t)-P(S_1\leq t)>0 \quad (11)$$

This relation indicates that if we change arrival rate of requests from l to $2l$, it is not necessary to increase the number of servers/personnel from c to $2c$.We need personnel less than $2c$.

3.2 Non-preemptive Priority $M/M/c$ Queue

This model is similar to previous model. Here we have some different types of requests that a priority has been assigned to each type of the requests.

Requests with same priority are serviced via FIFS and requests with upper priority are serviced before lower priority requests. Higher priority requests cannot cut the service of the lower priority requests while servicing them.

Fig. 2 shows the Non-preemptive priority $M/M/c$ model for maintenance environment.

Notice that in non-preemptive priority $M/M/c$ queue, we have only one priority queue but for simplicity in representation we showed a queue for each type of requests in Fig. 2

Table 2 shows the symbols we use for modelling the maintenance environment as non-preemptive priority $M/M/c$ queue

Probability that a request has to wait in queue is similar to case of $M/M/c$ queue and is

computed through "Relation (6)" and "Relation (7)".

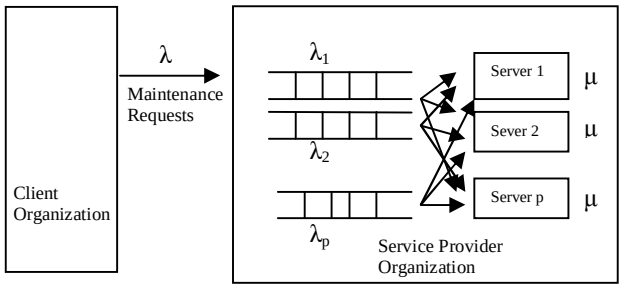

Fig. 2: Non-preemptive priority M/M/c queue

Table 2: List of symbols for analyzing the non-preemptive priority $M/M/c$ queue

Symbol	Description
l_i, $i=1,..,p$	Mean rate that requests of type i are entered into service provider Organization
$l=\sum_{i=1}^{p} l_i$	Mean rate that requests are entered in service provider organization
m	Mean rate that any server/person can service
c	Number of servers/personnel
W	Steady-state waiting time of a request in queue
W_i, $i=1,..., p$	Steady-state waiting time of a request of type i in the queue
B	Service time of a request
$S_i = W_i + B$, $i=1,..., p$	Steady-state sojourn time of a request of type i
$r_i = \frac{l_i}{cm}$, $i=1,..., p$	Traffic intensity of requests of type i over servers/personnel
$d_i = \sum_{j=1}^{i} r_j$, $i=1,..., p$	Sum of traffic intensity of the requests of type 1 to i over servers
$r = d_n = l/cm < 1$	Traffic intensity of requests over servers/personnel
P_k	Probability of existence exactly k requests in steady-state system

If let W_i denote the steady state waiting time in the queue for priority i requests, then for $i=1$ we have [8]:

$$P(W_1 \le t) = 1 - P(W > 0)e^{-cm(1-r_1)t} \quad (12)$$

But there is no simple way for finding the probability distribution of W_i for $i \ge 2$. In this paper we try to find a lower bound for $P(W_i \le t)$ $i \ge 2$ by using the below information from [8]:

$$E(W_i) = \frac{P(W > 0)}{cm(1-d_{i-1})(1-d_i)} \quad (13)$$

$$E(W_i^2) = \frac{2P(W > 0)(1-d_{i-1}d_i)}{(cm)^2(1-d_{i-1})^3(1-d_i)^2} \quad (14)$$

For finding this lower bound, we use the Markov's inequality that is defined as follows:

***Markov's inequality*:** let X be non- negative random variable and a , b be two constants greater than zero, then,

$$P(X \ge a) \le \frac{E(X^b)}{a^b} . \quad (15)$$

"Relation (13)", "Relation (14)" with "Relation (15)" can result that:

$$P(W_i > t) \le \min\left\{\frac{E(W_i)}{t}, \frac{E(W_i^2)}{t^2}\right\} \quad (16)$$

Notice that $P(W_i > t) = P(W_i \ge t)$, because W_i is continuous random variable.

The "Relation (16)" shows that there is an upper bound for $P(W_i > t)$ and consequently a lower bound for $P(W_i \le t)$. We use this lower bound as estimation for true $P(W_i \le t)$. It is quite probable that this bound be an underestimation of $P(W_i \le t)$ and consequently cause to overstaffing of a maintenance project, but in practice since often satisfying the SLA conditions for lower priority requests is ignored, we may hopefully have a closer estimation to number of servers/personnel.

We need to compute $P(S_i \le t)$ in order to understand that if the SLA condition in "Relation (2)" is satisfied. For case $i=1$ we have:

$$P(S_1 \le t) = 1 - e^{-m.t} + \frac{P(W > 0)}{1 - c + c.r_1}(e^{-m.t} - e^{-c.m(1-r_1)t}) \quad (17)$$

And for $i \ge 2$ we find an estimation of it:

$$P(S_i > t) = P(W_i + B > t)$$
$$= \int_{x=0}^{\infty} P(W_i + x > t) m.e^{-m.x} dx$$
$$= \int_{x=0}^{t} P(W_i > t - x) m.e^{-m.x} dx + \int_{t}^{\infty} m.e^{-m.x} dx$$

And after some simplifying we get it lesser than

$$\leq E(W_i^2)\int_{x=0}^{m}\frac{m.e^{-m.x}}{(t-x)^2}dx+$$

$$E(W_i)\int_{x=m}^{t}\frac{m.e^{-m.x}}{t-x}dx+e^{-m.t} \quad (18)$$

$$\leq \frac{E(W_i^2)}{(t-m)^2}(1-e^{-m.m})+$$

$$\frac{E(W_i)}{t-m}(e^{-m.m}-e^{-m.t})+e^{-m.t} \quad (19)$$

Integrals in "Relation (18)" can not be solved with usual methods. For finding an approximation of them, we can use numerical methods or Monte Carlo methods. Today powerful mathematical software's can also give a good approximation of them.

The "Relation (19)" shows a lower bound for $P(S_i \leq t)$.

We use the "Relation (18)" for an estimation of $P(S_i \leq t)$ and present an algorithm for finding an estimation of required servers/personnel for multiple service level projects that we discussed in section 4.2.

The last subject we discuss here is about being the meaningful or meaningless the conditions of SLA.

Suppose SLA conditions are as "Relation (2)", corollary of "Relation (10)" indicates that:

"The necessary condition for an SLA with multiple service level to be meaningful is that for all $1 \leq i \leq p$ *have the* $t_i > \frac{\ln(1-a_i)}{-m}$ *".*

4 Algorithms for Finding the Optimal Staffing Level for Software Maintenance Projects

In this section, we present some algorithms for finding the optimal staffing level for software maintenance projects with different conditions.

These algorithms are based on previous relations and results.

4.1 When Requests Have the Same Priority

In this case, we assume that requests have the same priority .Distribution time of requests arrival is exponential with parameter l and service time distribution is exponential with parameter m.

The algorithm for finding the minimum personnel satisfying the SLA condition in "Relation (1)" is as follows:

Algorithm 1: Algorithm for finding the minimum of Servers (personnel) when requests have the same priority

Input: l, m, t, a

Output: $c_{\min}$ (minimum personnel for satisfying the $P(S \leq t) \geq a$)

---

1) $c \leftarrow \lfloor l/m \rfloor + 1$ 2) $r \leftarrow l/cm$

3) Compute $B(c-1, c.r)$ through "Relation (7)"

4) $P(W>0) \leftarrow \frac{r.B(c-1,c.r)}{1-r+r.B(c-1,c.r)}$

5) Compute $P(S \leq t)$ through "Relation (9)"

6) if $P(S \leq t) < a$ then $c \leftarrow c+1$; goto 2 endif

7) $c_{\min} \leftarrow c$

4.2 When the maintenance requests have non-preemptive priority

Suppose we have p types of requests. The distribution time of requests arrival for the ith priority requests is exponential with parameter l_i and service time distribution is exponential with parameter m. We also suppose that the distribution time of all requests arrival is exponential with parameter $l = \sum_{i=1}^{p} l_i$.

The algorithm for finding estimation for minimum servers/personnel that satisfy the SLA condition in relation (1.2) is as follows:

Algorithm 2: Algorithm for finding estimation for the minimum servers/personnel when requests have non-preemptive priority

Input: $l, l_1, ..., l_p, m, t_1, ..., t_p, a_1, ..., a_p$

Output: $c_{\min}$ (an estimation of minimum personnel that satisfy the $P(S_i \leq t_i) \geq a_i$ for $1 \leq i \leq p$)

---

1) $c \leftarrow \lfloor l/m \rfloor + 1$ 2) $r \leftarrow l/cm$

3) Compute $B(c-1, c.r)$ through "Relation (7)"

4) $P(W>0) \leftarrow \frac{r.B(c-1,c.r)}{1-r+r.B(c-1,c.r)}$

5) $r_{i\ i=1,...,p} \leftarrow \frac{l_i}{cm}$ 6) $d_{i\ i=1,...,p} \leftarrow \sum_{j=1}^{i} r_j$

7) $E(W_i) \leftarrow \frac{P(W>0)}{cm(1-d_{i-1})(1-d_i)}$

$i = 2,..., p$

8) $E(W_i^2) \leftarrow \dfrac{2P(W>0)(1-d_{i-1}d_i)}{(cm)^2(1-d_{i-1})^3(1-d_i)^2}$

$i = 2,..., p$

9) Compute $P(S_1 \le t_1)$ through "Relation (17)"

10) $L_1 \leftarrow P(S_1 \le t_1)$

11) $m_i \leftarrow t_i - \dfrac{E(W_i^2)}{E(W_i)} \quad i = 2,..., p$

12) Compute $\int_{x=0}^{m_i} \dfrac{m.e^{-m.x}}{(t_i - x)^2} dx$ and

$\int_{x=m_i}^{t_i} \dfrac{m.e^{-m.x}}{t_i - x} dx$ for $i = 2,..., p$

13) $L_i \leftarrow 1 - E(W_i^2) \int_{x=0}^{m_i} \dfrac{m.e^{-m.x}}{(t_i - x)^2} dx$

$- E(W_i) \int_{x=m_i}^{t_i} \dfrac{m.e^{-m.x}}{t_i - x} dx - e^{-m.t}$

$i = 2,..., p$

14) if exists $1 \le i \le p$ such that

$a_i > L_i$ then $c \leftarrow c + 1$; goto 2;

end if

15) $c_{\min} \leftarrow c$

4.3 When the service times of requests is different

In this case in contrary to previous situation, each type of requests has its own service time distribution. Here we only can give an upper bound for the number of servers/personnel.

We use $M/M/c$ queue to find the minimum personnel for each type of requests, then the sum of the number of all personnel gives us an upper bound. Algorithm is as follows.

Algorithm 3: Finding an upper bound for number of servers/personnel when the service times of requests are different

Input: $l_1,..., l_p, m_1,..., m_p, t_1,..., t_p, a_1,..., a_p$

Output: $c_{\min}$ (upper bound for the number of Servers /personnel)

---

1) for i from 1 to p do Algorithm 1 with input l_i, m_i, t_i, m_i and get output $c_{\min(i)}$

2) $c_{\min} \leftarrow \sum_{i=1}^{p} c_{\min(i)}$

5. Conclusion

In this paper, we presented a method to estimate the optimal number of personnel needed to do a software maintenance project that has multiple service levels. Theses service levels are stipulated in one or some SLA(s) that determines the priority and response time of the requests. For the case that service times of requests were equal, we gave estimation for the number of personnel and for the case that service times of requests were different, we gave an upper bound for it. we showed that in order to reduce the maintenance costs when outsourcing the software maintenance projects it is better to offer them to service providers that do the maintenance works extensively.

References

[1] N. Chapin and et al, "Types of software evolution and software maintenance" *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 13, No. 1, pp. 3-30, 2001.

[2] A. Rana Ejaz, "Software maintenance outsourcing: Issues and strategies," *Computers & Electrical Engineering*,Vol. 32, No. 6, pp. *499-453, 2006*.

[3] J. Asundi and S. Sarkar, "Staffing Software Maintenance and Support Projects," *Proceedings of the 38th Hawaii International Conference on System Science*, 2005.

[4] H. Kung and C. Hsu, "Software Maintenance Life-Cycle Model," *Proceedings of the International Conference on Software Maintenance*, pp. 113-121, 1998.

[5] M. D. Penta and et al, "Modelling web maintenance centers through queue models," *Conference on Software Maintenance and Reengineering*, 2001.

[6] R. Ramaswamy, "How to staff business-critical maintenance projects," *IEEE Software*, Vol. 17, No. 3, pp. 90-94, May-June 2000.

[7] I. Adan and J. Resing, *Queuing Theory. 2001.* Available online from: http://www.win.tue.nl/~iadan/queueing.pdf/.

[8] O. Kella and U. Yechiali, "Waiting times in the non-preemptive priority M/M/c queue," *Commun. Statist. - Stochastic Models*, Vol. 1, No. 2, pp. 257–262, 1985.