Studying Supply and Demand of Software Maintenance and Evolution Services

Alain April, PhD
Department of Software Engineering, ÉTS University
1100 Notre-Dame West, Montreal, Canada
alain.april@etsmtl.ca

Abstract—Software maintenance and evolution constitutes an important part of the total cost of the life cycle of software. Some even argue this is the most important fraction of the cost. The added value of software maintenance and evolution is often not fully understood by the customer leading to a perception that software maintenance organizations are costly and inefficient. A common view of maintenance and evolution is that it merely fixes bugs. However, studies over the years have indicated that in many organizations the majority of the maintenance effort is devoted to value-added activities. To improve customer perceptions it is important to provide them with better insight into the activities performed by the maintenance organization and to document such performance with objective measures of software maintenance activities.

In this paper, software maintenance trend analysis is used as a basis for improvement. First, the differences between software maintenance activities and the Information System (IS) department development projects are described. Then a trend model is developed as a mean to manage the expectations of customers. To conclude, some remarks are made regarding the application of trend analysis by maintenance work categories for software maintenance managers.

Keywords: Software Maintenance, Software Maintenance improvement, trend analysis

I. INTRODUCTION

The software life cycle can be divided into two major parts: (1) initial software development, and (2) software maintenance. As has been well documented by Abran [2, 10] and Pfleeger [11], software maintenance is different from software development. It is different because of the following characteristics (Abran et al)[1]:

- The size and complexity of each maintenance work request are such that one or two resources can usually handle it;
- Maintenance work requests come in more or less randomly, and cannot be accounted for individually in the annual budget-planning process;
- Minor enhancements (adaptive) work requests in the enhancement category are reviewed with customers and can be assigned priorities;
- The maintenance workload is not managed using project management techniques, but rather with queue management techniques;
- Maintenance has a broader scope of configuration management with more operational considerations.

The software maintenance workload is user-services-oriented and system-responsibility-oriented. Priorities can be shifted around at any time, and corrective work requests, that address production failures, often take priority over other work in progress.

The four major aspects on which software maintenance focuses are (Pfleeger)[11]:
- Maintaining control over day-to-day functions;
- Maintaining control over system modification;
- Perfecting existing acceptable functions;
- Preventing performance from degrading.

An important aspect of software maintenance work performed on a system or environment might not be directly visible to the customer; therefore customers often feel that maintainers do not provide enough value versus the cost of their services. Such a negative perception is basically due to a lack of information made available to customers to explain all the services exchanged, and the many contributions of maintainers to the organization. Most software maintenance organizations have not progressed yet to the point of transparency, where they share detailed management information with their customers in terms they understand [4].

This is also based on a premise that the software maintenance processes are documented, mature and stable across the organization, and that they are well understood by the maintainers themselves.

Before addressing trend analysis, it is important to present the reader with the prerequisites to this measurement. The Software Engineering Institute (SEI) has identified that there is a basic set of common measures that can be used by all software organizations: size, effort, schedule and quality. It is once these basic measurements are operational that trend measurement should be envisaged. To enable the measurement of these four basic measures, it is recommended that the following software maintenance processes must be reviewed: (1) software maintenance activity definitions across all maintenance organizations, (2) the change request process, and (3) the time recording system. Most software maintenance organizations will need to initiate process improvement activities to implement, document and review these three basic processes before initiating productivity measurement.

During 2009, a process improvement activity was initiated in Integratik, an ERP development firm in Canada, to implement a maintenance request tracking process and information system. Each customer request sent to the software maintenance organization need to be recorded,
dispatched and tracked formally. For this process to be clear to every software maintenance employee involved, it was necessary to review and standardize the many software maintenance activities and services offered in this organization. This improvement project ensured that the demands for software maintenance services could be recorded, measured and analysed in an orderly fashion. A second process improvement activity was initiated to enhance the time-recording system to track maintenance personnel efforts according to: (1) “billable administrative activities” (also sometimes called productive activities) and (2) “non billable administrative activities” (also sometimes called nonproductive activities). This second process improvement project ensured that the “demand” for software maintenance services could be measured and analysed by ensuring that the time records accurately reflected the effort spent by each member of the software maintenance organization. With both the billable and non-billable perspectives being tracked, insights into productivity will be possible in the future.

II. BEFORE DRAFTING MAINTENANCE TRENDS

This section describes, in more detail, the prerequisites to trend measurement of software maintenance. The goal of section A is to define the most important aspects of software maintenance, in order to build the trend model in Section III.

A. Software Maintenance Activities

For an organization to mature, it is necessary that its processes (or set of activities) for carrying out the software maintenance activities be defined. Lientz and Swanson [9] were the first researchers to propose categories of software maintenance. Since then, standards for software maintenance (ISO/IEC 14764 [8] and IEEE Std. 1219-1998[6]) have documented the international consensus on maintenance work categories. These standards divide software maintenance into four of these work categories (as presented in Table I), which are defined in terms of: (1) the timing of the change (proactive or reactive), and (2) the goal of the change (correction or enhancement).

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CATEGORIES OF SOFTWARE MAINTENANCE WORK</strong></td>
</tr>
<tr>
<td><strong>Proactive</strong></td>
</tr>
<tr>
<td>Preventive</td>
</tr>
<tr>
<td>Reactive</td>
</tr>
</tbody>
</table>

Software maintenance organizations typically work on software systems developed internally, and they have access to the source code. When the software has been acquired from a third party a maintenance contract is required to formalize the third party maintenance agreement. In this case the software maintenance personnel becomes the main interface with the vendors, who in turn guarantee the software maintenance services. These additional maintenance interactions with third parties lead to more interfaces and to hand-over, making maintenance work even more difficult for software maintenance customers to understand.

Software Correction Dimension: The two main work categories identified in the ISO/IEC14764 software maintenance international standard have to do with corrections (preventive maintenance and corrective maintenance) to the software. These corrections are aimed at keeping the software at an agreed quality level in conformity with the established service levels. The first work category of corrections, preventive software maintenance, provides activities, after the initial delivery of the software, to prevent failure of software system by detecting and correcting latent errors before they become effective faults. The second work category, corrective software maintenance, comprises the reactive modifications needed for the elimination of an error condition in a software system that is impacting the operability of a production system. This service covers all associated coding, testing, change control, software distribution, documentation, job rerun and file recovery required to fix the error condition. Corrective maintenance is necessary and often caused by hidden faults in the software system. These faults, which will not have been uncovered by the many verification and validation processes of software development, should be dealt with initially by the organization supplying the software (as a form of project or contract warranty). Once the new software has been stabilized, the software maintenance organization will take over the responsibility of the software system to operate within the service levels agreed and established in the SLA (Service Level Agreement).

Software Enhancement Dimension: The two work categories in the enhancement dimension seem less familiar and are defined in terms of results rather than activities. Proactive enhancements (perfective maintenance and adaptive maintenance) are usually initiated by the IT organization to enhance the maintainability of the software and improve software maintenance effectiveness. This type of change does not change the information system functionality and should therefore be highlighted as such in the SLA. The functionality enhancements required by the customer do have a large impact on his perception of the software maintenance value and therefore must be tracked closely. This is mainly because they are not to be considered as small software maintenance work when the effort associated with implementing them requires more than 5 days of a maintainer’s time. The first work category, perfective maintenance, provides minor functional improvements, quality, maintainability and operability improvements for a specific system or the whole system portfolio with the objective of reducing the current level of resource consumption. The second work category in this dimension, adaptive maintenance, provides activities required to evolve the system to a change in the operating environment: hardware, operating system or volume, where no new or changed functionality is required. These ISO/IEC14764 definitions should be used to categorize each work request and track employee time against these categories. Management reports using these categories should accompany each piece of operational software. As well, the organization should ensure that all the software maintenance
personnel track their time (known as effort) accurately against these internationally accepted software maintenance
categories of work.

B. Change Request and Time tracking processes

At Integratik, four major sources of maintenance request
have been identified: software customers, software operators,
project managers completing their projects and requests for
reengineering studies. Because the size and complexity of
each work request to be assigned to a maintenance team must
be handled by one or two resources, it is only essential at the
outset of a work request that it be assessed and, when it is
outside of the maintenance scope in terms of either required
time and resources or levels of authority, be redirected to the

Call or Ticket to
Support Group (A,B & C)

Route request
to Account Manager
to be assessed as
Major Enhancement or
Project

CORRECTIVE
Interrupt work to
attend to failure,
insert in list of work
after repair

ADAPTIVE
insert in list of work
request, treat by priority

PREVENTIVE
insert in list of work
request, treat by priority

PERFECTIVE
insert in list of work
request, treat by priority

ADAPTIVE
insert in list of work
request, treat by priority

Fig. 1. A Change Request assignment process
development organization for review, prioritization, staffing
budgeting, planning and execution. This is often a source of
confusion in the IS department, since it is initially difficult to
form a consensus on how much of the enhancement effort is
to be carried out by maintainers, and what the threshold is for
a hand-over to a software project organization [4].

The procedure illustrated in Fig. 1 was used and a threshold
of 5 person days was selected as the maximum effort that the
software maintenance personnel could devote to a perfection
request (often called a minor enhancement). The United
Kingdom Software Metrics Association (UKSMA) also
recognizes this five-day limit: ‘The distinction between
maintenance activity of minor enhancements and development
activity of major enhancement is observed in practice to vary
between organizations. The authors are aware that in some
organizations a request as large as to require 80 workdays is
regarded as maintenance, while in others the limit is five days.
Initially it is proposed that the ISBSG and UKSMA will adopt
the convention that work requiring five days or less will be
regarded as maintenance activity.’ [7] This threshold is very
important in organizations, as it defines what software
development is, as opposed to software maintenance.

A work request larger than the approved threshold of 5
days is redirected to the IS department account managers for
budgeting review and planning, to assess whether it is part of
a major enhancement or will be bundled into small
development projects. The work request tracking system,
often automated using help desk commercial software (i.e.
Remedy/BMC Software, HP Service Center, etc.), supports
the creation, follow-up and closure of an individual work
request (also called a ticket). Both the Time Recording System
and the work request tracking system should be integrated to
ensure that effort is entered only once by the maintenance
personnel. The use of the work request procedures and toolset
should be agreed upon with customer management to ensure
that they are aware of individual work request priorities and
status. An audit of this process, system and list of
active/pending requests by customers should be prepared at
least once a year.

Since software maintenance service levels are based on
tracking effort for each service, it is essential that the IS
department Time Recording System (TRS) and reporting be
accurate. A review of the TRS associated with software
maintenance work requests is necessary to ensure
completeness and accuracy of the results. If it is found that
software maintenance does not account for close to 90% or
more of their effort/cost, a correction/improvement plan
should be put in place to ensure completeness and accuracy of
these records.

TABLE II

<table>
<thead>
<tr>
<th>Activity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative</td>
<td>General Admin., housekeeping, office work, communication, reporting</td>
</tr>
<tr>
<td>Break/ Non-productive</td>
<td>Lunch, tea, personal work, non-productive task, sickness</td>
</tr>
<tr>
<td>Classify and update work request</td>
<td>Preventive, corrective, perfective, adaptive or customer request for information</td>
</tr>
</tbody>
</table>
The solution to incomplete or deficient time tracking accuracy is not technically difficult: it is a software engineering management issue. It requires, on the one hand, the setting up of a standard chart of accounts (as illustrated in Table II) for the daily tasks performed on software maintenance work requests, and, on the other hand, a data collection procedure for all maintainers. To explore economic productivity in ways that lead to insights for the customer charge-back or improvement process, a reasonably granular chart of accounts is required. Details of time reporting on time spent per work request and maintenance category should be available on demand to both the IS department and the customer organization. Definitions of collected data and validation of that data are essential steps which the IS department must take to ensure data integrity.

III. MEASURING MAINTENANCE TRENDS

This section describes how maintenance trends were created in the Intégratik case study.

A. Studying the demand for maintenance services

The monthly demand for software maintenance can be expressed as the total of all requests in a given month (Correction work requests + Enhancement work requests). The software maintenance demand reporting was based on this approach in 2009 (presented in Table III).

<table>
<thead>
<tr>
<th>Work Request Demand</th>
<th>Corrections</th>
<th>Enhancements</th>
</tr>
</thead>
<tbody>
<tr>
<td>From last month</td>
<td>33</td>
<td>4</td>
</tr>
<tr>
<td>New this month</td>
<td>204</td>
<td>3</td>
</tr>
<tr>
<td>Solved this month</td>
<td>219</td>
<td>6</td>
</tr>
<tr>
<td>Carried forward</td>
<td>18</td>
<td>1</td>
</tr>
</tbody>
</table>

While these totals provide high-level information aggregated for all the systems in this organization, they do not provide enough information on maintenance work requests and they are not conducive to productivity analysis and comparisons. A first improvement is to identify the maintenance personnel effort associated with each request and report it (presented in Table IV). Monthly supply for software maintenance can be derived from the total effort, by work category, reported in a given month (Σ work request time reports).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
<td>218</td>
<td>209</td>
<td>180</td>
<td>607</td>
</tr>
<tr>
<td>Supply (days)</td>
<td>870</td>
<td>1111</td>
<td>1222</td>
<td>3203</td>
</tr>
</tbody>
</table>

Since distinct system have both different technical characteristics as well a customer driven demand peculiar to their own domains, it is also necessary to report software maintenance work requests by categories and by system for the same period of three months (presented in Table V):

<table>
<thead>
<tr>
<th>Software Maintenance work requests</th>
<th>System A</th>
<th>System B</th>
<th>System C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correction</td>
<td>59</td>
<td>35</td>
<td>34</td>
</tr>
<tr>
<td>Enhancement</td>
<td>95</td>
<td>101</td>
<td>96</td>
</tr>
<tr>
<td>Customer Query</td>
<td>59</td>
<td>70</td>
<td>56</td>
</tr>
<tr>
<td>Unclassified</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

This data highlights that most of the software maintenance effort, in Intégratik, is carried out in enhancements, corrections to production problems and customer query/support of service. One could proceed to do this for the entire IT system supported by the Intégratik. With this information, the QA manager could create a graph representing the service trends. (as illustrated by Fig. 2)

![Fig. 2. Intégratik maintenance service trends by systems (3 months)](image)
management of the software maintenance portfolio. Such a chart can also be produced by cost per category for budget view. Once the customers have worked with the detailed information provided by the IS department, there are additional measures that can be introduced. Lines of code and Function Point measures can be used as normalization factors to compare the customer systems between themselves and to the industry. However, obtaining these measures from COTS and SaaS can be quite difficult, if not impossible.

B. Studying the supply of maintenance services

Table IV presented that software maintenance personnel have recorded 3203 days of work effort for that specific period of time. This represents 16 work-years for this organization (assuming 200 work-days in a year, with 7.5 hours of billed work per day), which represents an average of 5.2 days of effort per work request.

Comparing and studying the percentage of distribution of work per category can provide insights into the overall distribution of effort in one organization relative to other maintenance organizations. It is important to understand how much time an average request takes per category for budgeting reasons. It is of interest to the IS department customers to have an idea of where the software maintenance hours are being spent relative to their demands. Measurement of the workload distribution is essential to the IS department managers in order to evaluate whether or not the trends are going in the direction that the ‘paying customer’ wants.

C. Studying trend analysis per system supported

With six to twelve months of work effort distribution per software maintenance category, managers at Intégratik should be able to start representing trends. In this experiment, we observed that the customer appreciates downward trends in the corrective requests, while upward trends in perfective requests (small enhancements) enhancements are also appreciated.

![Fig. 3. Ideal trends of work effort by category for system in maintenance](image)

Trends represented in Figure 3 demonstrate to the customers that progressively less effort is required to control existing software systems (e.g. corrections and tuning take less effort out of his budget), while more functionality and “value added” support is increasing. Figure 3 represents trends, which a typical customer would like to observe from his software maintenance supplier.

As stated earlier, for that analysis to be possible, the software maintenance personnel should take great care on the assignment of work requests to a specific category. An indirect benefit of measuring will be to collect historical data for the future estimation of software maintenance work. The following section describes where this data could be beneficial for Maintenance and Support management:

Software corrections: Whereas a fire fighting mentality was once predominant, there is now a more professional to presenting the software maintenance contributions. Comparison between software systems is also made possible.

The information gathered from analysing this category of work can be used to define quality objectives by system, for external supplier and software development teams before their handover of the software. Such quantitative figures can also be used to initiate preventive maintenance programs and justify proposed system reengineering efforts.

Enhancements: The ratio of adaptive maintenance to other categories of work can also be an indicator of mandatory enhancements and of the introduction of new products. It is an indication of the rate of introduction of changes and new technology in the organization. It can also lead to identifying suppliers who have the most impact on an organization’s resources (effort required by products).

Preventive, perfective and portability maintenance: The data collected for each category of work can be used to evaluate the quality and efficiency of the products and softwares. It can also be used to compare the costs and level of system tuning required by each system. Insights from such analysis can help to justify reengineering decisions and provide insights on targeted initiatives in this area of the software maintenance portfolio.

Customer queries: Too much support effort identified for a piece of software might point to a root cause in a previous life cycle phase which was shortened, lacking training, suffered from the poor quality of a third-party supplier, or received too little investment to obtain the expected stability. This can help maintainers in identifying software development weaknesses.

IV. CONCLUSION

Software maintenance trend analysis can only be envisaged by organizations that collect data systematically. Many basic definitions of software maintenance work categories and effort collection activities are required as prerequisites to this activity. A number of process improvement projects need to be initiated to ensure that the proper processes and basic information are in place to successfully conduct such a measurement. Progress towards detailed productivity analysis can be considered once the software maintenance organization has achieved a higher level of maturity [3].
ACKNOWLEDGMENT

This study could not have been carried out without the help of Claude Éthier, president of Intégratik Canada. Sincere thanks are extended to all the software maintenance managers who contributed to the software maintenance trend measurement project.

References