Analysis of Maintenance Work Categories Through Measurement.

ALAIN ABRAN HONG NGUYENKIM

Montréal Trust, Montréal,
Québec, Canada, H3A 3K9 (514-982-7175)

Abstract

This article presents empirical data from a two-year measurement effort in the maintenance environment of a Canadian organization. The findings reported here are based on a daily data-collection process including 2,152 work requests which required 11,365 days to complete. This set of empirical data confirms some of the findings of the opinion surveys in terms of the stability of the overall workload distribution in the maintenance area, while the analysis of any subset of data indicates that this overall average hides significant differences. This paper includes a discussion on the improved measurement program implemented, and illustrates how insights into the maintenance process are gained through various measurements.

Introduction

Many organizations that started to implement computer applications 20 to 30 years ago now have a major portfolio of applications to support their business operations. Surveys [3,6] indicate that 50% to 70% of an information systems (IS) budget is spent on maintenance activities. However, these surveys of IS managers have not been supported by empirical data and according to [6,10], very little empirical software maintenance studies were carried out in the '70s and '80s.

Software maintenance is often defined as including all activities associated with changing, modifying or otherwise altering existing software applications [10], or alternatively as work done on a software system after it becomes operational [2,9]. Within that frame of reference maintenance work is further divided into the following three categories of changes [1,8,11]:

(1) Corrective: changes to correct program failures, performance failures and implementation failures
(2) Adaptive: changes to adapt a program to changes in data requirements and processing environments
(3) Perfective: changes to enhance performance, improve cost effectiveness, efficiency and maintainability.

Software maintenance productivity has been defined as a simple output/input ratio, most typically thousands lines of code or thousands of function points maintained per work-month [2,4,5]. While interesting, such a ratio does not lead to causal relations in maintenance productivity studies, and our research motivation was to add to the body of knowledge in favor of more appropriate measure of maintenance productivity. This research is a study of the measurement work carried out in software maintenance at a Canadian financial institution in 1989 and 1990. This empirical research work can be classified as a field study wherein researchers collect data on several objects or groups of objects to identify significant differences, either at a single point in time or across time [10]. All of the data originated from one organization, therefore neither inter-organizational nor industry differences should introduce noise in the measurement process [2]. The drawback is that this may limit the applicability of the results. However, due to the lack of empirical data on software maintenance these results will clearly be of interest to researchers in that field and there is no *a priori* reason to believe that the maintenance measurement illustrated could not be applicable across organizations and industries.

The purpose of this presentation is to report on the adjustments required to the above mentioned maintenance categories framework definition and to present and analyze the data collected. The scope of this empirical research is therefore limited to a report on the

CH3047-8/91/0000/0104/$01.00

maintenance measurement program as implemented and to illustrate the initial benefits derived from this measurement program.

Definitions: Development and Maintenance

In the industry-research site under study, the definition of development work includes not only the development of new applications, but also major enhancements (additions and changes) to existing applications. In both cases the issue, or problem to be solved, is complex and requires a team effort. Usually a business case must first be prepared, and then must be approved by senior executives, and is usually handled within a project structure. These projects must be planned ahead of time, generally on an annual basis at budget time. The nature of the work carried out for major enhancements can be classified as a variation of the development life-cycle process.

The following pie-chart (Fig. 1) shows the distribution of the workload of this organization whose business units rely heavily on their computerized applications. Of the total budget, less than 10% is dedicated to building totally new applications from scratch, while 35% goes into the addition of new business functions within existing applications (classified as major enhancements in Fig. 1).

At the empirical site under study, the maintenance work is defined as non-related project work, and, even with this restrictive definition, it still represents the greatest share (55%) of the total budget.

Maintenance, as defined from now on, does not require a project management structure for its performance, and is handled very differently:

1- The size and complexity of each work request are such that it can usually be handled by one or two resources.

2- Work requests come in more or less randomly and cannot be accounted for individually in the annual budget-planning process.

3- Work requests are reviewed by operations committees which are responsible for assigning priorities. Priorities can be shifted around at any time, and any work request on a production problem takes priority over work in progress.

4- The operations committees must work within preset annual budgets, and they must manage their queue of work requests (requests outstanding, completed, in progress, etc.).

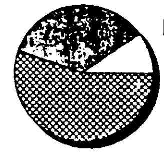

Figure 1: Research Site Workload (1989)

5- If additional funding is required during the year (for example, when there are too many requests in the queue or when the delays for initiating or completing work requests are too long), the executive steering committee is approached for funding approval to improve service levels, but not to fund specific requests.

6- The work request will require less than 60 person-days to complete (This cut-off varies from 30 days to 80 days in various other organizations).

Maintenance Measurement Program History.

The maintenance process at this industry-research site is recognized as a vital function of the corporation, and the maintenance area is run like an internal service bureau for the user community. The budget is approved annually by each of the business units, with a specific number of annual billable work-days for the maintenance of each application. These costs are then charged back, on a monthly basis, to each profit center. Maintenance work request queues are managed by joint user/I.S. operations committees.

The maintenance management processes have been formalized and improved over the past few years. A work request can only be initiated through a signed form, called a Request For Service (RFS). All work carried out that pertains to this Request For Service (RFS) is logged on against the RFS number in the Time Reporting System (TRS), which includes the effort required to conduct the impact analysis and prepare the estimate, and to analyze, program, test and implement the solution. This effort is entered daily by all staff into the computerized TRS system, and all time spent over half an hour must be reported. Daily, weekly and monthly controls ensure that all of the effort is recorded. There is also a variety of reports produced from this time-reporting system, including monthly billing to the user community with details of all time spent for every RFS for which there was activity during the current month.

Up until the end of 1988, the productivity measurement process was based on what could be called the metrics for the management of queues, such as:

- the number of outstanding requests;
- the average waiting time before being serviced;
- the estimated number of days in the queue;
- the number of requests completed;
- the number of requests in progress;
- comparison of estimates versus actual costs, etc.

While these standard maintenance metrics provide information on queue servicing performance levels, they do not provide enough information on the maintenance work product, and they are not conducive to productivity analysis and comparisons.

Senior I.S. management was therefore interested in looking at the feasibility of implementing additional measurement concepts within the *maintenance* areas. They were looking not for specific answers from the maintenance metrics, but rather for tools to manage and improve the maintenance process. They also needed metrics to measure the progress and benefits brought in by the introduction of new productivity tools, both in the maintenance area and on the downstream side of major investments in development productivity tools, such as the very expensive CASE technology.

This required, as the first step proper classification of the maintenance work. Lientz and Swanson's classification classification system was selected with its *Corrective*, *Adaptive* and *Perfective* maintenance categories. By definition, these work categories focus on changes made to the software applications. However, a significant number of work orders do not request changes but only information on the software components.

To these three categories, a fourth was therefore added, called *User Support*, to include the following types of activities:

- User requests for information on the particulars of application rules and behaviour;
- Requests for preliminary analysis;
- Requests for work estimates;
- Requests for application data information through an adhoc (one-time) report which would not be reused or implemented in the production environment.

It must be noted that this classification is not standard, either in academia [7] or in industry. Readers should therefore be careful when comparing this set of data with data originating in their organization or with survey data published in the literature.

Data pertaining to general administration and overhead activities are not included in this presentation, which is strictly limited to a consideration of maintenance work that can be attributed directly to a specific work request, and that can be billed to the user community as work carried out on their applications.

Maintenance Workload Distribution

Two years-worth of data have been collected and examined: a total of 2,152 work requests were classified and analyzed (Table 1), which accounted for a total of 11,332 days of work effort, or 57 work-years (This organization's yearly average of billable days per person is 200 days, a day representing 7 hours of actual billable work). This in turn represents, on average, 5.2 days of work effort per work request.

The maintenance workload distribution was analyzed by work category, and the results are presented in Table 2. From this table it can be seen that maintenance teams in 1989 did far more than correct defects and errors (36%). In fact, they spent 64% of their time in 1989 doing other types of maintenance work: they spent 29% adding new functionalities to the applications, 11% optimizing application performance and a quarter of their time (24%) answering user requests for information.

The distribution of work is fairly stable over the two-year period, specifically for the corrective and user-support types of maintenance work, while the shift in effort between the adaptive work (from 29% to 34%) and the perfective work (from 11% to 5%) reflects a shift of emphasis in this organization from a focus in computer cost reduction in 1989 to a focus on adding functional enhancements to facilitate business operations.

	1989	1990
No. of work requests	1,013	1,139
Effort (days)	5,209	6,123

Table 1: Quantity of Data Analyzed

Work category	1989	1990
Corrective	36%	35%
Adaptive	29%	34%
Perfective	11%	5%
User Support	24%	25%
Non Corrective	64%	65%

Table 2: Maintenance Workload Distribution

No other hard data sets were found for comparison purposes. However, some figures were available from industry surveys. In Feb. 1991, Zvegintzov [12] published the following comparison (Table 3) of data obtained from three surveys: Lientz & Swanson surveyed 487 organizations in 1980, R. K. Ball surveyed participants at the 1987 Annual Meeting and Conference of the Software Maintenance Association and S. Dekleva surveyed participants at the 1990 Annual meeting of the same professional association.

In order to compare the above figures with this set of empirical data, the following equivalences were established: "enhancements" become adaptive, "adaptation", "tuning", "documentation" and "re-engineering" are classified as perfective maintenance, and "user support" and "other" become user support activities. The 1990 Abran's data set has also been restated to take into account the 35% of all IS efforts for major enhancements: corrective (21%), adaptive (21%) + enhancements (39%), perfective (3%) and user support (15%). The results of this reclassification are shown in Table 4.

The Abran's data set, based on real data but from a single organization, illustrates one instance of real measurement of the maintenance workload distribution. This data set is remarkably close to the Lientz & Swanson 1980, specifically for the corrective and adaptive categories while the ratios are reversed in the perfective and user support categories.

It must be noted that the user support category had not been identified as such in the Lientz & Swanson survey. The Ball and Dekleva survey results concur with the user support category, but differ significantly in the adaptive and perfective categories which could be caused by definitional discrepancies and the re-classification schema selected.

Work Category	Lientz & Swanson 1980	Ball 1987	Dekleva 1990
Corrections	22%	17%	16%
Enhancements	59%	39%	43%
Adaptations	6%	9%	8%
Tuning	4%	5%	5%
Documentation	6%	5%	6%
Re-engineering	N/A	10%	9%
User support	N/A	12%	12%
Other	3%	3%	1%
Non corrective	78%	83%	84%

Table 3: Zvegintzov's Comparison Table of Maintenance Effort

Quarterly Analysis

Analysis of the quarterly figures for 1989 (Fig. 2) indicates that the workload distribution is fairly constant throughout the year, especially for the first, second and fourth quarters. If the perfective and user support categories are grouped together, they represent approximately 35% for the three quarters. Only in the third quarter is there a decrease of a few percentage points in the corrective maintenance category, offset by a corresponding increase in adaptive maintenance work.

Work Category	Lientz & Swanson 1980	Ball 1987	Dekleva 1990	Abran 1990
Corrective	22%	17%	16%	21%
Adaptive	59%	39%	43%	60%
Perfective	16%	29%	28%	3%
User support	3%	15%	13%	15%
Non corrective	78%	83%	84%	79%

Table 4: Data Sets Restated

Figure 3 provides a tentative explanation for this variation in work distribution. In this figure, we can see that the workload for three of the four 1989 quarters is around the 1,400-day mark, while there are fewer than a 1,000 days of workload for the third quarter (July, August and September). In fact, this corresponds to peak summer holidays for both the I.S. staff and the user community. From the same figure, it can also be seen that two work categories (corrective and user support) have been cut almost in half, while the other two (adaptive and perfective) have remained almost the same in terms of the number of days spent on each.

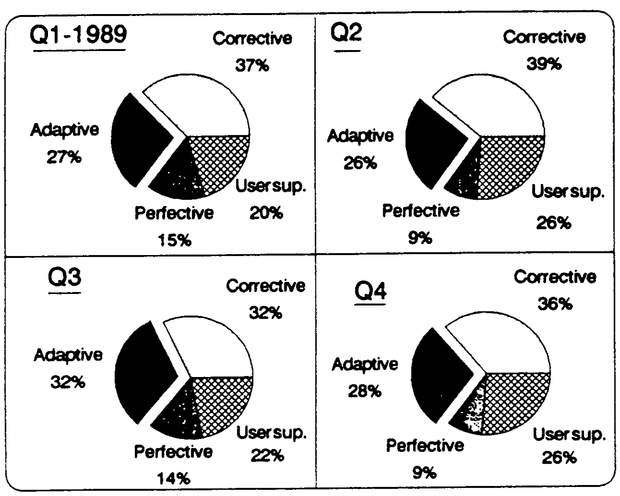

Figure 2: 1989 Quarterly Workload Distribution

The various maintenance managers have provided the following reasons for this phenomenon:

- During the summer months, there is less user data input (decreased workload) in the applications, leading to a decrease in user-induced errors;
- During the summer months, there is almost no project implementation in the production environment, and therefore fewer sources of application destabilizing factors;
- There are fewer requests for information from the user community;
- Finally, there are fewer programmers working on both development and application support, and also, therefore, fewer sources of programmer-induced destabilizing factors.

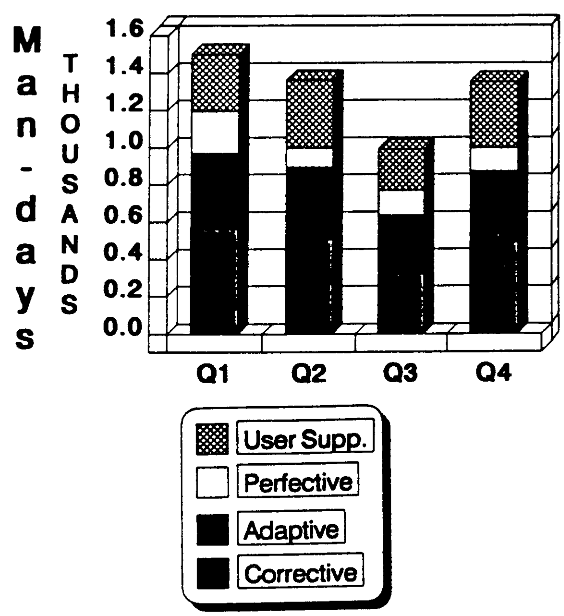

Figure 3: 1989 Quarterly Effort Volume

Workload Distribution Shift 1989-90

The shift in work effort noted in Table 2 is illustrated partially by Table 5, which shows a significant increase in work requests in the adaptive category (from 20% to 33%), a decrease in the perfective category (from 9% to 6%), and in the user support category (from 44% to 35%).

It must be understood that, except for corrective maintenance, the number of work requests is strictly user-driven, and reflects, up to a point, their priorities. For example, in 1989 there was a major impetus to cut computer chargeback costs, while in 1990 the emphasis was shifted towards adaptive maintenance through a significant increase in requests (from 20% to 33%) to add new (but small) functionalities to existing applications.

Work category	1989	1990
Corrective	28%	26%
Adaptive	20%	33%
Perfective	9%	6%
User Support	44%	35%
Non Corrective	72%	74%

Table 5: Work Requests Distribution

However, there is not a one-to-one relationship between the number of work requests and the effort required. The average number of days per work request varies according to the category of maintenance work, as illustrated in Table 6.

Table 6 also shows that while the "All Categories" average did not change very much over the 2-year period (from 5.14 days to 5.38), there are significant differences in the average number of days by work category, as well as over this period within the same category (for example, the adaptive maintenance average dropped from 7.51 days to 5.58, while the user support average went up from 2.80 days to 3.80). When both factors (number of work requests and average number of days per type of work request) are taken into consideration, the shift in total work effort within the work categories of Table 2 can be explained.

From Table 2 and Table 5, it can be seen that while 44% of the work requests fall within the user support category, they account for only 24% of the workload, or, on average, only 2.8 days. This is almost half of the overall average of 5.2 days; this can easily be explained since for most of these requests there is no programming or testing involved - it is purely analytical work. It could also be an indicator of the analytical work required before initiating any work in the other categories where programming changes are required (to functions, data or JCL). It might also be an indicator of the fixed costs in a particular maintenance work request.

Work category	1989	1990
Corrective	6.73	7.36
Adaptive	7.51	5.58
Perfective	6.63	4.33
User Support	2.80	3.80
All Categories	5.14	5.38

Table 6: Average No. of Days by Work Request

Trend Analysis

The two years of historical data, by quarter, are represented graphically in the following two figures. Figure 4 illustrates the relative workload distribution by maintenance category over the 24-month period.

Over this two-year period, the largest proportion of maintenance work fell into the corrective category to keep the computerized applications operational, and into the adaptive category to enhance the applications in order to cope with the changing needs of business and of users. The effort required for adaptive action and user support grew steadily. For the corrective category, the annual peak in the first quarter corresponds to peak processing volumes, in addition to special year-end procedures.

The next figure (Figure 5) was prepared with the cumulative historical data to smooth out the various abrupt and seasonal changes noted previously in Fig. 4. These cumulative data provide a better indication of the trends at the industry-research site under study. Two maintenance categories show downward trends (corrective and perfective), which represents positive improvement in terms of a slight decrease in the relative number of resources required to support applications that keep increasing in size and complexity.

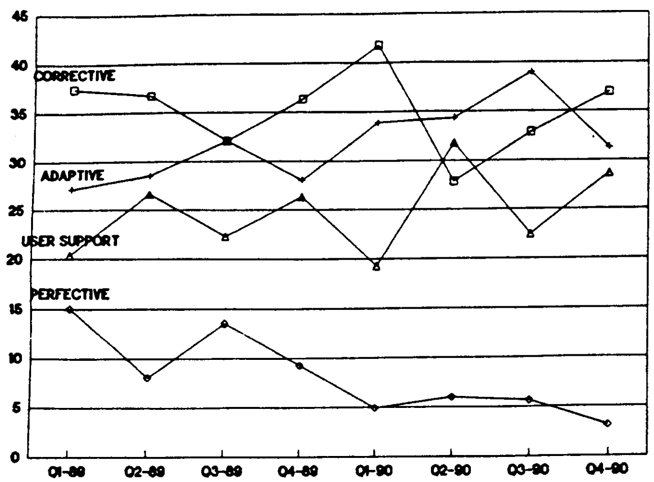

Figure 4: Relative Distribution (%) Over 8 Quarters (Qi)

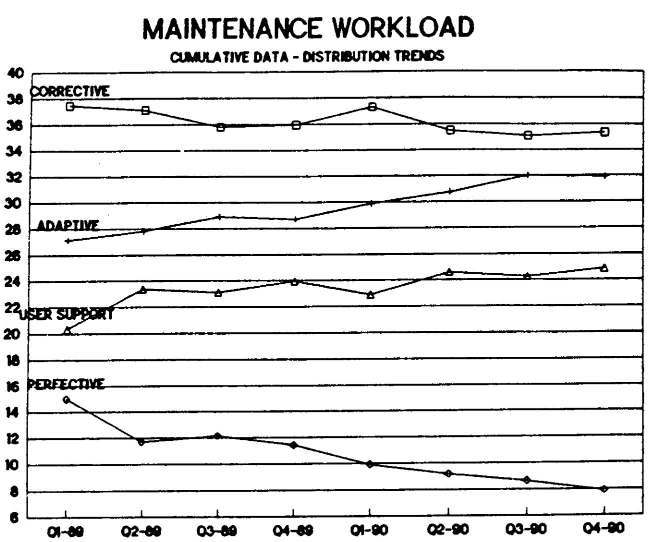

Figure 5: Cumulative Distribution (%) Over 8 Quarters

The other two categories (adaptive and user support) show upward trends. The increase in the adaptive category is encouraging and illustrates the ability of this organization to dedicate a greater share of its maintenance staff to work on functional enhancements to these applications in response to changing business requirements. The upward trend in the last category, user support, is more difficult to interpret and will need further analysis. Preliminary information indicates that a significant turnover in the user areas have had to be offset by additional support from the software maintenance teams in terms of additional training, coaching and analytical support in system analysis.

Analysis by Application.

While the previous figures demonstrate some stability in the maintenance workload distribution at the overall level (for all applications combined), the next research question was to verify whether or not this stability was also present at the application level. The results are presented in Table 7, broken down by major applications (over 1 million lines of code), packages, and small applications bundled together.

From Table 7 it can be seen that there is considerable variation from one application to another. Each major application has a different effort distribution, with application C showing a greater variation from the other three and from their combined average. Due to their importance relative to the overall work effort (4,946 days out of 6,123 days = 81%), the major applications have considerable influence on the overall average. Their combined work effort hides the much greater variations in work effort spent on maintaining either packages or small applications.

This leads us to believe that the type and size of software applications might have a significant impact on the work effort distribution (large applications, small applications, packages, and their relative effort distributions). It should also be noted that the set of data analyzed does not contain any information on non-mainframe-based applications.

The M.I.S. applications analyzed in Table 7 are briefly described in the following paragraphs.

Major application A: This application has over one million lines of code and was developed in the early '80s using structured methodologies and a Database Management System (Network Database). Maintenance activities are performed in a structured way and the system documentation is kept up-to-date. There is a

Application	Total days	1989	1989	1989	1989	Total days	1990	1990	1990	1990
		Corr.	Adap	Perf	User		Corr	Adap	Perf	User
Major A	1205	34%	32%	12%	22%	1740	26%	49%	4%	21%
Major B	991	43%	29%	14%	27%	772	30%	33%	7%	31%
Major C	1571	50%	14%	14%	22%	1882	46%	25%	1%	28%
Major D	454	17%	51%	6%	26%	552	22%	46%	7%	25%
Subtotal	4,221	40%	27%	9%	24%	4,946	34%	37%	3%	26%
Packages	614	20%	30%	31%	19%	753	59%	12%	11%	18%
Small Appli-cations	407	15%	51%	2%	32%	424	9%	46%	13%	32%
Subtotal	1,021	18%	38%	20%	24%	1,177	41%	12%	23%	23%
TOTAL	5209	36%	29%	11%	24%	6123	35%	34%	5%	25%

Table 7: Maintenance Work Distribution By Application

considerable amount of development work being carried out by different development teams concurrently.

Major application B: This application is mostly batch, with a few online processes. It closely matches (within 5%) the subtotal averages for each work category for both 1989 and 1990 data.

Major application C: This was initially a package-based application which was acquired in the early '70s, and which has been completely overhauled since then. No structured methodology has been used in this application, and the maintenance team has considerable experience in maintaining it. The 1989 workload distribution is not entirely accurate due to an underrepresentation of work requests in the adaptive category: there were some problems in this work category in counting the function points, and, in the opinion of the researchers, this generated an overrepresentation in other categories. However, the data collection process was improved in in 1990, and reflects a more accurate work distribution for the second year for this major application.

Major application D: This is a business application package designed in the '70s and implemented in the mid '80s, with a significant amount of modification having been carried out using structured methodology. The data collected is considered very accurate, even though it differs substantially from the averages. This application is considered bug-free compared to other applications, and it performs extremely well, leading to an underrepresentation of work effort in the corrective maintenance category.

Standard packages: these are essentially package-based applications, with smaller workloads and less data individually. Their work distribution varies considerably, depending on the specifics of the package implementation and degree of modification.

Other small applications: these applications are run in a service bureau, outside the corporate data center. The information collected is incomplete, and not necessarily accurate.

Concluding remarks

The results of this field study confirm some of the findings of the opinion surveys in terms of the stability of the overall workload distribution in the maintenance area, while providing researchers some interesting insights into the maintenance process itself. The emphasis in this last section is put on the benefits observed in this industry-research site, derived as direct benefits from this maintenance measurement program in terms of a much better understanding of the maintenance function and of the costs attached to each type of maintenance. At budget time, for example, the maintenance managers can come to the user community with a lot more information on the services provided and the goods delivered. Discussion can then focus on sets, or subsets, of information, which can be verified at the lowest level of primitives (the work requests), and not on a unique global number for the whole maintenance process.

Corrective maintenance

Where the quality, or lack of quality, of an application had previously been to some extent a matter of hearsay, the corrective maintenance ratio of the application can now be used to compare it to other applications. Questions are raised on applications with very high ratios in this category. The time spent in this category is considered "untouchable" at budget time, a fixed expense of the application, and the bare minimum to keep an application running. The information collected in this category has been an essential component in developing quality indices for each application. These quality indices will be monitored over the next few years. The numbers are used to build business cases to justify preventive maintenance programs, and later to monitor the realized benefits. This information is considered critical for future quality improvement programs, and will be used to promote and justify preventive maintenance concepts. It will also help to quantify the cost avoidance factor in quality improvement programs.

Adaptive maintenance

The ratio of adaptive maintenance by application is an indication of the amount users are willing to invest in each application in order to obtain *additional information* (business information) or *additional functions*. This is, in our opinion, not a reflection of the weaknesses of an application, but rather an indication of the users willingness to invest money to enhance their own operations through the leverage of additional computerized functions within their existing applications.

Perfective maintenance

This maintenance category accounts for only 11% of the workload, on average. However, there are major differences from one application to another and these could be monitored. It has been observed that applications based on software packages have a much higher ratio of perfective maintenance (from 50% to 75%, as compared to the 11% average). This could be explained by the very low ratio of corrective and adaptive maintenance. These packages have a proven track record, and must be almost foolproof (e.g. general ledger, payroll, accounts receivable packages). However, they do not seem to use computer resources efficiently and the transaction cost is relatively high resource consumption as compared to internal applications, therefore requiring a high level of system tuning throughout the maintenance life-cycle.

User Support

The 24% of the time spent on non-programming-related work requests came as a surprise, both in terms of the amount of effort expended and in terms of the number of requests (44%). Measurements in this category of maintenance activities has helped define this type of work as a specific business function provided to the user community, at their request and under their total control. It also indicates that the user community both needs this information to manage their businesses, and is willing to pay I.S. specialists' rates to get this information.

For example, a request for information on the details of a computational algorithm, based either on the system documentation or on the program code, will provide them with the appropriate, and accurate, business information that they could not otherwise obtain through their own channels. It is now recognized in this organization that maintenance teams not only correct bugs, but also provide the user community with "business information", that is both timely and accurate.

The user support function has even been incorporated in the I.S. mission statement as a "provider of business information" to the user community.

Acknowledgements

This research was carried out jointly by Montréal Trust and the Organizational Research Division of the Canadian Workplace Automation Research Centre/Communications Canada. This joint project was made possible under the terms of the National Scientific Staff Exchange Program. The opinions expressed in this report are solely those of the authors. The authors would like to thank the anonymous referees and professor Sasa Deklava for the considered comments and helpful suggestions.

References

[1] L.J. Arthur, *Software Evolution. The Software Maintenance Challenge*, John Wiley & Sons Ltd, England, 1988.

[2] G.K. Gill and C.F. Kemerer, "Productivity Impacts of Software Complexity and Developer Experience", *MIT Industrial Liaison Program*, Report 3-40-90, 1990.

[3] R. Harrison, *Maintenance Giant Sleeps Undisturbed in Federal Data Centers*, Computerworld, March 9, 1987.

[4] International Function Point Users Group (IFPUG), *Function Points as an Asset - Reporting to Management*, IFPUG, Ohio, 1990.

[5] J.R. Johnson, *The Software Factory - Managing Software Development and Maintenance*, 2nd edition, QED Information Sciences, 1991.

[6] B. P. Lientz and E.B. Swanson, *Software Maintenance Management*, Addison-Wesley, Reading, MA. 1980.

[7] N. Madhav and S. Sankar, "Application of Formal Specification to Software Maintenance", *Proceedings Conference on Software Maintenance - 1990*, San Diego, Nov. 26-29, 1990.

[8] J. Martin and C. McClure, *Software Maintenance - The Problem and its Solutions*, Prentice-Hall, 1983.

[9] G. Parikh and N. Zvegintzov, *Tutorial on Software Maintenance*, IEEE Computer Science Press, Silver Spring, MD, 1983.

[10] S. Sharpe, D.A. Haworth and D. Hale, "Characteristics of Empirical Software Maintenance Studies: 1980-1989", *Journal of Software Maintenance Research and Practice*, Vol. 3, 1-15, 1991.

[11] E.B. Swanson, "The Dimensions of Maintenance", *IEEE 2nd International Conf. on Software Engineering*, Proceedings, San Francisco, Oct. 1976.

[12] N. Zvegintzov, "Real Maintenance Statistics", *Software Maintenance News*, Vol. 9, no. 2, Feb. 1991.