IMPLEMENTING A MEASUREMENT PROGRAM IN SOFTWARE MAINTENANCE - AN EXPERIENCE REPORT BASED ON BASILI'S APPROACH

Jean-Marc Desharnais
SELAM
7415, rue Beaubien Est, Suite 509
Anjou (Québec), Canada
H1M 3R5
desharnais.jean-marc@uqam.ca

France Paré
Régie des rentes du Québec
P.O. Box 5200
Québec (Québec), Canada
G1K 7S9
france.pare@rrq.gouv.qc.ca

Marcela Maya
Université du Québec à Montréal
P.O. Box 8888 (Centre-Ville)
Montréal (Québec), Canada
H3C 3P8
maya.marcela@uqam.ca

Denis St-Pierre
SELAM
7415, rue Beaubien Est, Suite 509
Anjou (Québec), Canada
H1M 3R5
dstpierr@crim.ca

Introduction

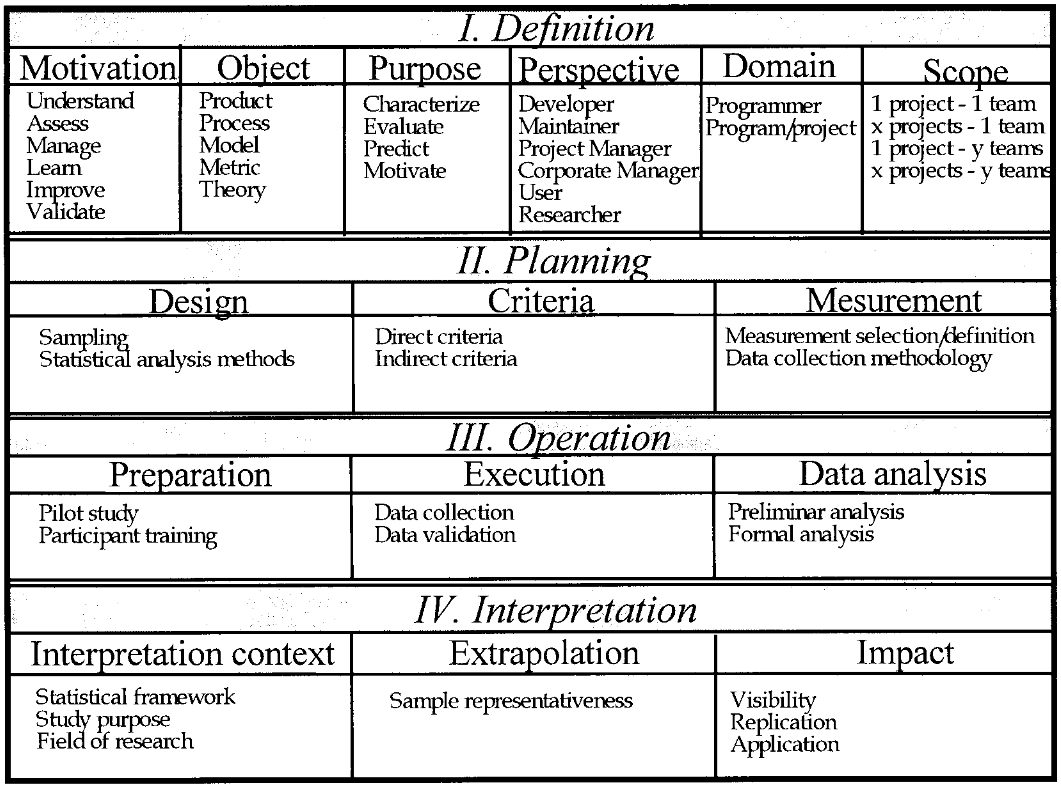
Software maintenance refers to all those activities, both technical and managerial, that take place after a software product becomes operational. Software maintenance is an area of vital importance to industry since most organizations now have a major application portfolio that supports their business operations. Software maintenance ensures that these applications continue to meet organizational and business objectives in an effective way, thereby keeping the organization competitive.

Some studies have been published on the workload distribution of maintenance activities. However, most of these studies are founded on surveys (Lientz and Swanson, 1980; Ball, 1987 and Dekleva, 1990), and few are based on actual collected data (Abran and Nguyenkim, 1993). The principal reason for this lack of published data is that the industry does not measure or collect maintenance-type data in a timely and accurate fashion (Abran and Nguyenkim, 1993). This article presents the initial results of a measurement program in software maintenance implemented at a Government of Quebec (Canada) agency. The measurement program and its initial results are presented in a structured manner using a framework introduced by Basili *et al.* (Basili, Selby and Hutchens, 1986) and used successfully by Bourque and Côté (1991) and Maya, Abran et Bourque (1996). First, a brief description of the framework itself is presented. Then, the measurement program is described following the structure of Basili's framework: definition, planning and operation, and interpretation of results.

1. Basili's framework

Basili *et al.* (1986) developed an experimentation framework "to help structure the experimentation process and to provide a classification scheme for understanding and evaluating experimental studies." They also recommend the framework as "a mechanism to facilitate the definition, planning and operation, and interpretation of past and future studies." Although this framework was proposed to structure the process and the presentation of *experimental* studies, it is also suitable for other kinds of projects which need a mechanism to facilitate their definition, planning and operation, and the interpretation of results.

Basili's framework has a four-phase structure (Figure 1). It begins with a definition phase where we develop our intuitive understanding of the problem we wish to solve into a precise specification of an experiment or project. This phase has six components, namely motivation, object, purpose, perspective, domain and scope.



I. Definition

Motivation	Object	Purpose	Perspective	Domain	Scope
Understand Assess Manage Learn Improve Validate	Product Process Model Metric Theory	Characterize Evaluate Predict Motivate	Developer Maintainer Project Manager Corporate Manager User Researcher	Programmer Program/project	1 project - 1 team x projects - 1 team 1 project - y teams x projects - y teams

II. Planning

Design	Criteria	Mesurement
Sampling Statistical analysis methods	Direct criteria Indirect criteria	Measurement selection/definition Data collection methodology

III. Operation

Preparation	Execution	Data analysis
Pilot study Participant training	Data collection Data validation	Preliminar analysis Formal analysis

IV. Interpretation

Interpretation context	Extrapolation	Impact
Statistical framework Study purpose Field of research	Sample representativeness	Visibility Replication Application

Figure 1: Basili's framework (Basili *et al.*, 1986)

The planning phase is next and consists in the selection of the experimental design and of the appropriate measures. During the design step, we select the various data that will be included in the sample. We then choose the statistical techniques that will be used to analyze this data, as well as the factors (direct and indirect criteria) that will be measured. Finally, we select measurements that will quantify these factors and we establish the data collection procedures and tools.

Subsequently, in the operation phase, the actual execution of the experiment and the problems encountered are discussed. Data are collected, validated and analyzed using the statistical techniques chosen during the design step.

Lastly, the results of the experiment are explained in the interpretation phase. First, the results are interpreted in the context of the statistical techniques that were used to obtain them. Then, the results are interpreted again, first in the context of the experiment's purpose and then in the context of the literature and of existing knowledge in the research field. If the samples studied are representative and the results are positive, extending the results to other environments and problems might be possible. The remainder of the article presents the measurement program following the phases and steps of Basili's framework. It is important to note that some minor changes have been made to Basili's framework, mainly in the interpretation phase, in order to adapt it to the particular context of the project presented here.

2. Definition

Two years ago, the Pension Plan Agency (PPA) of the Government of Quebec (Canada) started to work on the implementation of a measurement program for both software development and software maintenance. The Agency managers were interested in improving the development and maintenance processes. Among other things, they wished to improve the estimation process for development projects, to improve the budgeting process for maintenance tasks and to evaluate their performance in software development and maintenance. A plan and a schedule for a measurement program were presented and approved in April of 1996. The first phase of the program, which is the topic of this article, started in June of 1996 and focused on maintenance. The second phase will address development later.

Referring back to Basili's framework, the first phase of the measurement program at the PPA can be defined as follows. Its *motivation* is to understand and improve the PPA maintenance process and to evaluate PPA performance in software maintenance over years, across software applications and in comparison with that of other organizations. Its *purpose* or objective is to produce a budgeting model which will serve to establish the budget for maintenance tasks and to compare it with other models already established by other organizations in the same business area. The main entity or *object* being addressed by this phase of the program is therefore the software maintenance budgeting as seen from the manager's point of view (*perspective*). To reach the objectives of the program, all the PPA corporate applications will be measured, as well as the maintenance requests completed for these applications (*Scope*).

3. Planning

Design

To ensure that the results of the measurement program are as comprehensive as possible, it was decided that all the PPA corporate applications and sub-applications be measured (*Sampling).* These applications count for roughly 90% of the PPA software maintenance effort. We also planned to collect the effort data concerning all maintenance requests completed for these applications during the last three years.

To analyze this data, three types of statistical tools were selected (*statistical analysis methods*): descriptive statistical tools (average, median, standard deviation, etc.), correlation and linear regressions, and control charts (individual values and variations over time).

Criteria

This part of Basili's framework consists in selecting the criteria or factors related to the project purpose that will be measured using the selected sample. As specified in the program definition, we are interested in the budgeting of software maintenance. The maintenance process has many inputs and outputs, which in turn have many characteristics that can be measured (see Figure 2). For the purposes of this particular program, it was decided that five characteristics be measured: the functional size and the lines of code of the applications, the effort required to complete each maintenance request for these applications, the type of maintenance work (maintenance category) and some environmental characteristics. Figure 2 shows the selected characteristics in italics.

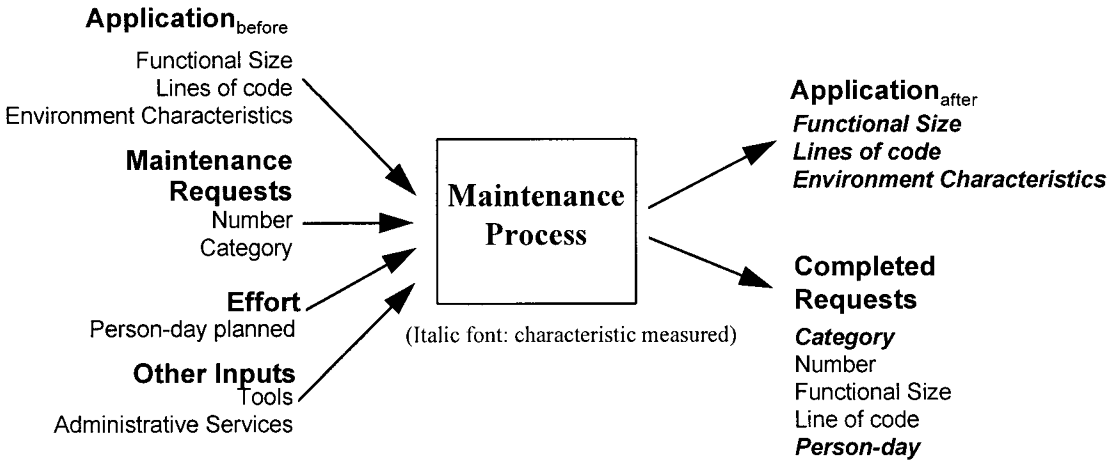


Figure 2: Maintenance process

Measurement

The measures selected to quantify the criteria described in the previous section are the following:

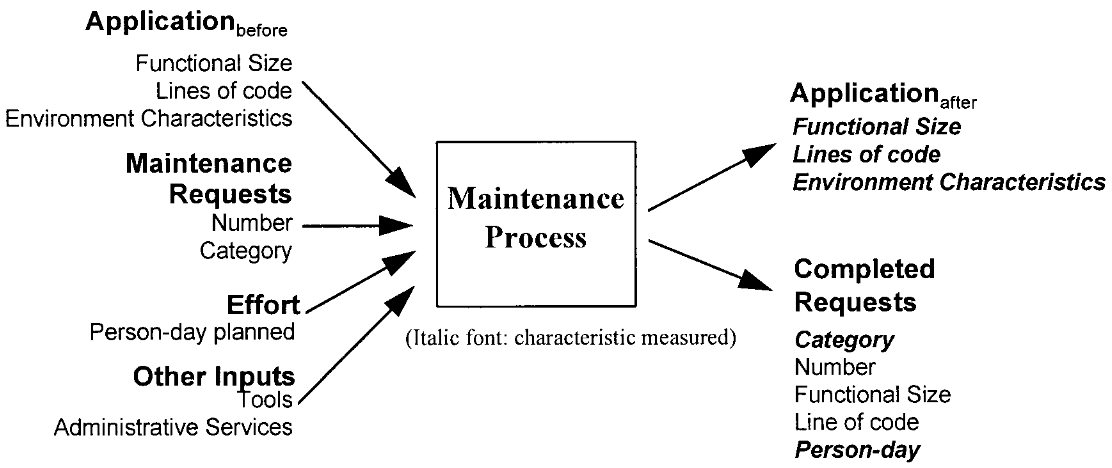
- Functional size: The chosen measure was Function Point Analysis (IFPUG release 4.0, 1994). Function Point (FP) is currently the functional size metric most often used and it continues to gain adherents in the MIS field. FP has proven to be successful for building productivity models and for estimating project costs.
- Lines of Code (LOC): Total number of LOC excluding comments and spaces.. This was used only to check the relationship between the FP count and the number of LOC.
- Effort required to complete a maintenance request: The effort was measured in person-day. The

collect the effort data concerning all maintenance requests completed for these applications during the last three years.

To analyze this data, three types of statistical tools were selected (*statistical analysis methods*): descriptive statistical tools (average, median, standard deviation, etc.), correlation and linear regressions, and control charts (individual values and variations over time).

Criteria

This part of Basili's framework consists in selecting the criteria or factors related to the project purpose that will be measured using the selected sample. As specified in the program definition, we are interested in the budgeting of software maintenance. The maintenance process has many inputs and outputs, which in turn have many characteristics that can be measured (see Figure 2). For the purposes of this particular program, it was decided that five characteristics be measured: the functional size and the lines of code of the applications, the effort required to complete each maintenance request for these applications, the type of maintenance work (maintenance category) and some environmental characteristics. Figure 2 shows the selected characteristics in italics.



(Italic font: characteristic measured)

Figure 2: Maintenance process

Measurement

The measures selected to quantify the criteria described in the previous section are the following:

- Functional size: The chosen measure was Function Point Analysis (IFPUG release 4.0, 1994). Function Point (FP) is currently the functional size metric most often used and it continues to gain adherents in the MIS field. FP has proven to be successful for building productivity models and for estimating project costs.
- Lines of Code (LOC): Total number of LOC excluding comments and spaces.. This was used only to check the relationship between the FP count and the number of LOC.
- Effort required to complete a maintenance request: The effort was measured in person-day. The

PPA has a computerized time reporting system which keeps track of the identification number of each maintenance request, the application concerned, the actual effort (person-day) used to complete the maintenance requests, the type of maintenance work performed (maintenance category), etc. This system was therefore used to obtain the effort per maintenance request.

- Maintenance categories: A scheme of five maintenance categories was used by the PPA to classify the nature of maintenance work requests. After a literature review and verification in the field of the appropriateness of this classification scheme, it was decided that a new scheme be introduced that is better adapted to the PPA reality and more adequate for making comparisons with industry. The retained maintenance categories are shown in Table 1.

Category	Description
Adaptive	Modifications to adapt a software to changes in data requirements and processing environments (Abran and Nguyenkim, 1993).
Corrective	The reactive modification of a software product performed after delivery to correct discovered faults. The modification repairs code to satisfy functional requirements (ISO/IEC, 1996).
Preventive	Maintenance performed on the basis of predetermined criteria with the intention of reducing the probability of failure of the application.
Perfective	The modification of a software product after delivery to improve performance or maintainability (ISO/IEC, 1996).
User Support	Responding to user demands outside adaptive, corrective, preventive or perfective (Abran and Nguyenkim, 1993)

Table 1: Maintenance categories

- Environmental characteristics: A questionnaire to be completed by the manager in charge of each application was prepared. This questionnaire asked for information about:
  - Application identification
  - Technical constraints (response time, security, number of users, platforms)
  - Maintenance tools and techniques (development methodology, CASE tools)
  - Factors related to personnel (number of programmers, experience)

4. Operation

Preparation

Before conducting the actual data collection, several activities had to be performed, most of them related to the preparation of the FP count. For example: engagement of FP specialists to conduct the FP counting of the application portfolio; review of the application's documentation in order to determine whether or not it is necessary to have the application specialists present in the counting sessions and, if so, to arrange these meetings; review of the FP rules in order to clarify their proper interpretation according to the PPA. Another important activity that had to be performed in this phase of the project was to establish a procedure to convert the maintenance categories of the historical time recording database to the new categories.

Execution

Four FP specialists carried out the FP counts in June of 1996. The high quality of the application documentation, for both data and transactions, made it possible to count all applications without the intervention of the application specialists. The only task required by the application specialists had to perform was to answer the environmental questionnaires. This was an important factor which made the first part of the measurement program a success since the managers didn't have to worry about missing their deadlines because their application specialists were away in FP counting sessions.

Also in June 1996, the effort data concerning the applications being measured was extracted from the PPA time reporting system and the conversion of the maintenance categories was carried out. The final effort dataset contained all the maintenance requests completed from April 1993 to June 1996.

Once the FP counting was finished, a *data validation* process was undertaken. For the FP, the validation method proposed by Desharnais and Morris (1996) was used. This method proposes two types of validation: external validation (relation between functional size and effort, verification of the consistent use of the counting rules, etc.); and internal validation (verification of the scope of the FP count and of the boundary of the application being measured, comparison of the FP counts and the industry average, etc.). Concerning the effort data and the environment questionnaire, basically two elements were validated: The data completeness and the identification and the verification of the outliers.

Data Analysis

As mentioned in the definition of the project, the final users of the measurement program results are the PPA maintenance management team, including the IS general manager, the maintenance manager, the maintenance team leaders and, finally, the analysts. We therefore generated a variety of graphs and statistical data depending on the targeted audience. Some examples of the data analysis results are presented in the following paragraphs.

Figure 3 shows the relationship between the total FPs of each application and the FPs for only one of the function types, the Internal Logical Files (ILFs). These two variables have a good linear relationship. This result is consistent with some published empirical studies which suggest that, as a result of the strong linear relationship between these variables, the FPs for the ILFs could be used to validate the counting results and to make an early estimate of the total FP count of an application (Desharnais and Morris, 1996a, 1996b).

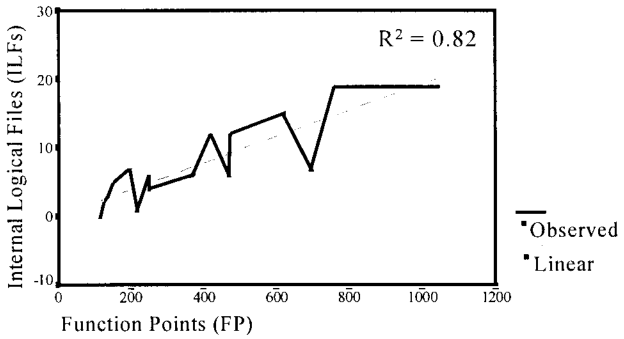


Figure 3: FP vs ILFs

Figures 4 and 5 illustrate the analysis carried out on the effort required to maintain the PPA application portfolio during the last three years. Figure 4 shows the effort of each PPA application between April of 1993 and June of 1996. Figure 5 again shows the effort by application, but this time normalized by 100 FP (Unit cost[1]). For example, figure 4 shows that application G required much more effort than the other applications. However, the unit cost of this application is not the highest (figure 5). Looking only at the total effort by application can therefore be misleading in terms of productivity and benchmarking analysis, and no decision should be taken based only on this variable.

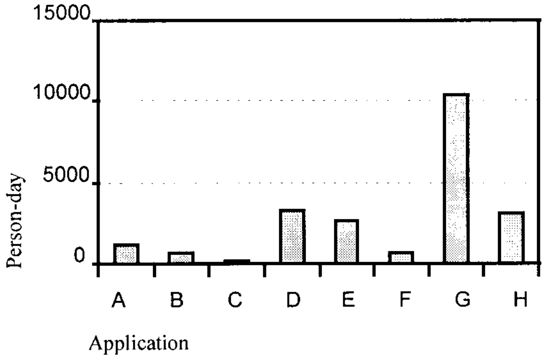


Figure 4: Person-day by application

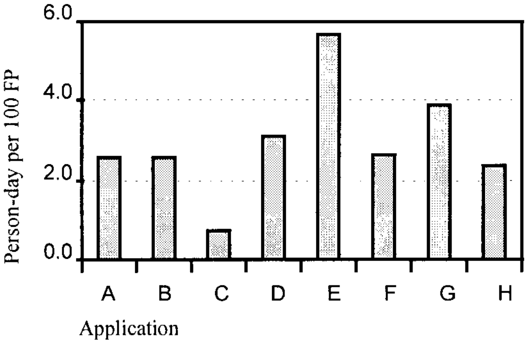


Figure 5: Unit cost by application

Charts showing the evolution and trend of a specific variable can be very useful in maintenance. For example, Figure 6 presents the evolution of the unit cost by maintenance category for application G. We can see that the effort spent on corrective maintenance decreases, while the effort spent on adaptive maintenance increases. These trends are encouraging for the maintenance team as they illustrate their ability to reduce the faults in the software and to dedicate more time to work on functional enhancements in response to changing business requirements. These types of graphs were prepared for each application measured as well as for the whole organization.

Figure 7 shows the behavior of the unit cost from one year to another. For the financial years 1994 and 1995, we can see that the unit cost tends to increase in May and January and to decrease in June and July and in December. The maintenance manager provided the following reason for this phenomenon: during the summer months and Christmas holidays, the number of maintenance requests completed diminishes because there are fewer people working, and therefore fewer person-day per 100 FP are worked. In January and September, the consequence of this phenomenon appears: the backlog of maintenance requests has to be processed and the number of maintenance requests increases.

[1] Since we only have the FP counts done in June 1996, to calculate the unit cost of previous years we estimated the FP from the number of LOC which we did have available.

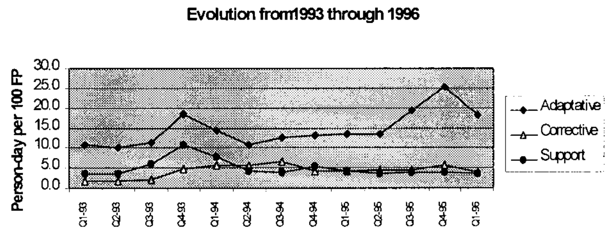


Figure 6:
Evolution of unit cost by maintenance category and quarter

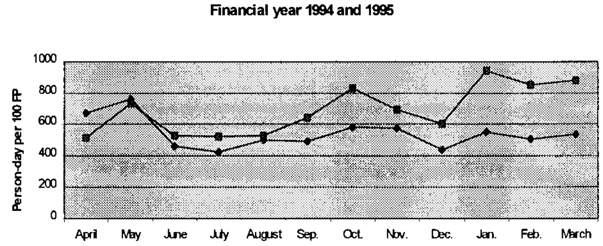


Figure 7:
Evolution of unit cost by financial year

Table 2 shows an example of the types of tables that were prepared as input for the budgeting model. This table presents the effort per application per maintenance category. Columns A to H show the total effort by application and by maintenance category during the year 1995. The last column presents the distribution of the effort by maintenance category as a percentage. This distribution means, for example, that adaptive maintenance represents 66% of the total effort for this year, while perfective maintenance represents only 0.5%.

Maintenance	Total effort by application (1995)								
Categories	A	B	C	D	E	F	G	H	%
Adaptive	49.2	14.2	9.4	72.8	206.2	35.7	91.2	40.5	66,4%
Corrective	11.3	58.1	1.6	5.8	19.0	10.2	30.6	11.9	17,6%
Perfective	0.4	2.1	0.4	0.7	2.2	0.0	0.4	0.3	0,5%
Preventive	0.0	0.0	0.0	0.2	0.0	0.0	2.2	2.8	1,3%
Support	7.8	12.6	3.5	9.0	19.7	7.3	26.5	7.3	14,2%
FP	474	239	252	1041	471	252	2664	1296	

Table 2: Unit cost by application and by maintenance category (1995)

5. Interpretation

Interpretation context

Two contexts were considered in interpreting the results of the measurement program: internal to the PPA and external. Regarding the internal interpretation context, the project results provided the maintenance managers with insight into the maintenance process and with a better understanding of the maintenance function. They now have empirical data which show the maintenance workload distribution by maintenance categories, the nature, evolution and trend of the maintenance workload, the cost attached to each type of maintenance, etc. The budgeting model will provide the managers with a toolset to budget services to be delivered.

A critical issue for the success of a measurement program is an adequate interpretation of the various data and graphs. It is very important to have a good knowledge of the maintenance environment and of the organization's environment in order to understand the real meaning of the data and, as a result, FP counts.

I. Definition

Motivation	Object	Purpose	Perspective	Scope
Understand and improve the maintenance process. Evaluate the performance in maintenance throughout time and between applications and other organizations	Maintenance budgeting	Produce a budgeting model and compare this model with others already established.	Mainteance manager	- Many applications - Many mainteance requests

II. Planning

Design	Criteria	Mesurement
Sampling: 90% of the application portfolio; maintenace requests completed during the last three years **Statistical analysis methods:** descriptive statistic, correlation, linear regressions, and control charts	- Applications functional size - Effort required to complete each maintenance request for these applications - Maintenance category - Environmental characteristics	- Functional size: Function points - Effort: person-day - Maintenance category: adaptive, corrective perfective, preventive and support - Ernvironment characteristics: questionnaire

III. Operation

Preparation	Execution	Data analysis
- Preparation of the function point counting sessions - Procedure to convert the actual maintenance categories to the new ones.	**Data collection:** - Carried by four FPspecialists in June 1996. - FP for all corporate applications (90% of porfolio) - Effort for maintenance requests completed between April 1993 to June 1996. **Data validation:** - FP: Procedure by Desharnais and Morris (1996) - Others: completeness and oultilers	- Maintenace requests vs effort - Evolution and trend of unit cost - Unit cost by application - Unit cost by maintenance category - Budgeting model

IV. Interpretation

Interpretation context	Extrapolation
Internal context: Insights into the maintenace process and better understanding of the maintenance function: mainteance workload distribution; nature,evolution and trend of the maintenace workload; cost attached to each type of mainteance; budgeting model. Knowledge of the maintenace environment is important **External context:** Comparison with other organization is diffcult	Good data representativeness, therefore good quality of resutls

Figure 8: Summary of the measurement program

This article also demonstrates the relevance of Basili's framework for structuring and presenting an industrial project. The use of this framework encourages the proper statement of the purpose and objectives of the project, the proper definition of what is going to be measured, and how and why, and validation of the data before jumping to the analysis and to the interpretation of the results in widening contexts.

References

Abran, A. and Nguyenkim, H. (1993), 'Measurement of the Maintenance Process from a Demand-based Perspective', *Software Maintenance: Research and Practice,* Vol. 5, 1993, pp. 63-90.

Ball, R. K. (1987) in Zvegintzov, N. 'Real maintenance statistics', *Software Maintenance News*, Vol. 9, No. 2, !991.

Basili, V. R., Selby, R. W. and Hutchens, D.H. (1986), 'Experimentation in Software Engineering', *IEEE Transactions on Software Engineering,* Vol. SE-12, No. 7, July 1986, pp. 733-743.

Bourque, P. and Côté, V. (1991), 'An Experiment in Software Sizing with Structured Analysis Metrics', *Journal of Systems and Software*, 15, pp. 159-172.

Bourque, P., Maya, M. Abran, A. (1996). 'A Sizing Measure for Adaptive Maintenance Work Products', *IFPUG 1996 Spring Conference*, Atlanta, 1996.

Dekleva, S. (1990), *1990 Annual Software Maintenance Survey,* Survey conducted and compiled for the Software Maintenance Association, P.O. Box 12004 no. 297, Vallejo, CA 94590.

Desharnais, J. M, and Morris, P.(1996a), Validation Process in Software Engineering: an example with Function Points, SES 96, Montreal.

Desharnais, J. M. and Morris, P. (1996b), Validation of Function Points - An Experimental Perspective, *IFPUG 1996 Spring Conference*, Atlanta, 1996.

IEEE, 1990, *IEEE STD 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1991, 83 pages.

IEEE, 1992, *IEEE STD 1219-1992 - IEEE Standard for Software Maintenance*, The Institute of Electrical and Electronics Engineers, Inc., New York, NY, 1992, 42 pages.

IFPUG (International Function Point Users Group) (1994), *Function Point Counting Practices Manual Release 4.0,* IFPUG, Westerville, OH, 293 pages.

ISO/IEC, 1996, *ISO/IEC JTC1/SC7/WG7 - ISO/IEC International Standard - Information Technology - Software Maintenance*, Project 07.37, Working Draft 1-2, August 2, 1996, 36 pages.

Lientz, B. P. and Swanson, E. B, (1980), *Software Maintenance Management.* Addisson-Wesley, Reading, MA.

take the right decisions.

Regarding the external context, one of the objectives of the project and one of the reasons for choosing FP[2] as a measurement technique was to have the capacity to compare the results with other organizations in the same business domain. However, the data available for this comparison, another agency of the Government of Quebec in the same business area (here called organization X), presents the following difficulties:

- The rules used by organization X to count the ILFs are quite different from the IFPUG rules. They decided to count each low level physical file as one ILF;
- The PPA applications are more integrated than those of organization X. Business functions used by several applications are implemented in only one application (PPA), instead of having the same business functions implemented in each application (organization X). Therefore, an organization that has less integrated applications (in this case organization X) could have a relatively higher number of FP.
- Regarding the effort data, two problems were identify at organization X: (1) there is not a validation process and (2) for some applications there were no efforts for many months because many applications are developed and maintained by the end users.

Extrapolation

The representativeness of the sample is a determining factor for the significance and importance of the results. The sample used to study the maintenance process and to build the budgeting model is very representative of the reality of the industrial site. The functional size of 90% of the application portfolio was measured using the standard FP counting rules, and more than three years of effort data were collected. This is a very representative sample for a first phase of a measurement program, and allowed us to produce reliable results.

6. Conclusion

A complete framework for the measurement program presented in this article is presented in Figure 8.

Even though the purpose of the project was not completely achieved (the comparison with the industry could not be made) the PPA managers were satisfied with the results. Two factors were crucial to the quality of the results: The quality of the applications documentation and the reliability of the effort data.

There is still more work to be done in the future to improve the measurement program in maintenance, for example: measurement of the functional size of the applications not included in the sample; measurement of the functional size of each maintenance request; analysis of the impact of the high level of integration of the applications in the total value of FP; implementation of an automated tool for metric calculation, storage and analysis; establishment of procedures to update the

[2] It is stated that FP are independent of the technical development and implementation processes. As a result, comparisons between organizations are possible.