SOFTWARE MANAGEMENT BASED ON SOFTWARE DELIVERABLES

ALAIN ABRAN
Montreal Trust
1800 Mc Gill College Av.
Montreal, Quebec H3A 3K9
Tel: (514) 982-7175 Fax: 982-7069

PIERRE N. ROBILLARD
Ecole Polytechnique de Montrtal
P.O. Box 6079, Succ. A
Montreal, Quebec H3C 3A7
Tel: (514) 340-4238 Fax: 340-4174

ABSTRACT

With Function Point metrics the deliverables of the software development process can be measured and, by combining the Function Point metrics with other units of measurements, various ratios can be derived to determine costs and productivity. This ability to measure the software development productivity enhances Information Systems corporate image and facilitates the move towards fixed-price projects, with the ability to make profits for re-investment in productivity programs. It brings Information Systems management back into the mainstream of the corporate culture centered aroung productivity measures and profit center concepts.

RESUME

Les metriques des points de fonction permettent de mesurer de façon objective les biens livrables produits par le processus du developpement de logiciel. En les combinant à d'autres métriques il est possible de determiner les coûts de revient et les niveaux de productivite du developpement informatique. Cette abilité à mesurer la productivite ameliore la crédibilité corporative des gestionnaires de systèmes d'information. Ceci permet Cgalement de gérer des projets à prixe fiie et de ré-investir les surplus dans des programmes d'amélioration de productivite. Cela permet au developpement informatique de s'enligner avec le courant principal de la culture corporative centrée sur les concepts des mesures de productivite et des centres de profit.

INTRODUCTION

Software developers need to implement the same basic measurement systems that are commonplace in the business and engineering fields. The inability to measure the output and the productivity of the software development process has given information systems developers a very poor image as managers in corporate boardrooms.

With Function Point metrics we have implemented key economic indices to manage the software development process. The Function Point metrics, combined with our previous types of measurements, provide us with tools to manage the productivity of the software development business unit like any other business unit in the corporation.

We will present management ratios based on Function Point metrics and see how they can be used to better understand the software development process. We will then highlight some of the major benefits to be derived from a software measurement program.

1- BASIC FUNCTION POINT METRICS CONCEPTS

In software development, we are doing a good job at measuring our costs and even at charging them out to the user community. We are doing very poorly, however, at measuring our development output and our productivity.

Some of us have very sophisticated standard accounting systems to keep track of salaries, CPU costs, diskspace costs, telecommunications, capital costs, amortization costs, overheads, etc. But how do we measure our output?

Unfortunately, the most common measure is the man-day! Who has not heard a department head proudly report his department's output during the fiscal year in terms of so many man-days? The next most common metrics, and even then it is used only by a handful of software departments, is the "Lines of Code" metrics (LOC). Even after forty years of software development, this metrics it is still not recognized as a valid metrics for software development productivity: it is not consistent across technologies, it cannot be trusted to measure productivity and it does not have identifiable relationships with the applications delivered. Nevertherless, until quite recently, it was still the only system output measure available.

The Function Point metrics were developed by Allan Albrecht from IBM. First published in 1979, they were revised in 1984. They are now used internationally by over 500 corporations and are generating considerable interest in the marketplace.

Fortunately, the new Function Point metrics, while far from being perfect, have addressed many of the basic deficiencies of the previous software measurement systems. For management information systems, they have proven to be independent of hardware technologies, programming languages and software development methodologies. Although they are still in the development stage, there already exists a normalized set of rules that are sanctioned by an industry-wide international organization that is monitoring their evolution.

Most importantly, this metrics provides a valid normalized measure of the system's deliverables that is meaningful to our customer base. In fact, the foundation of the Function Point metrics is the measurement of these deliverables from the user's viewpoint (see Figure 1).

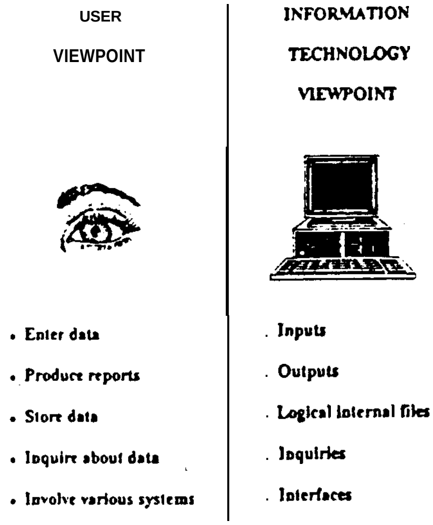


Figure 1

The measurement of the software product strictly from the user's viewpoint is a very different approach. The Function Point metrics provide a standard measure of the different types of functions requested by the user, and enable us to quantify these functions as well as analyze the characteristics of the software products delivered. This offers us major insights into the structure of the products as well as giving us a better undertanding of the software delivery process. While in the past our analyses were limited to productivity factors, we can now take into account this new dimension, the structural characteristics of the products delivered.

Function Point metrics can measure very early in the Development Life Cycle the number of functions requested and, at the end of the project, the number of functionalities actually delivered as compared to the initial estimates. They may also be used to identify, to measure and to negotiate changes throughout the project life cycle.

The Function Point metrics are designed to measure only the deliverables. By combining them with other units of measurement, additional measures, and ratios for determining costs and for analyzing productivity, may be derived. The Function Point metrics combined with productivity factors measured by other project management techniques will provide us, therefore, with the necessary tools to conduct productivity comparisons using Function Point as the reference metrics. Function Points provide the missing link in our productivity equation:

$$\begin{aligned} \text{PRODUCTIVITY RATIO} &= \\ & \frac{\text{QUANTITY OF DELIVERABLES}}{\text{TOTAL COSTS}} \\ &= \frac{\text{QUANTITY OF FUNCTION POINTS}}{\text{TOTAL COSTS}} \\ &= \text{UNIT COST PER FUNCTION POINT} \\ &= \text{UNIT COST PER DELIVERABLE} \end{aligned}$$

At Montreal Trust we implemented these metrics in 1987, and since then we have been aggressively pursuing and implementing concepts derived from the FUNCTION POINTS metrics and have linked them to our existing standard measurements of management and software engineering processes.

The Function Point metrics have allowed us to progressively put in place an integrated set of software measurement metrics that would be considered a standard management process in any manufacturing concern (but quite a feat for a software development shop!).

2- ANALYSIS OF THE DELIVERABLES

The first key unit measure is the total number of function points implemented in the PRODUCTION environment over the previous twelve months. By combining the project measures, we can derive the average unit cost per function point at the departmental level, or at any sub-level for which we have enough data:
See Figures 2-A and 2-B.

- by hardware platform;

- by data base platform;

- by development methodology;

FUNCTION POINTS INSTALLED BY TYPE
NEW APPLICATIONS

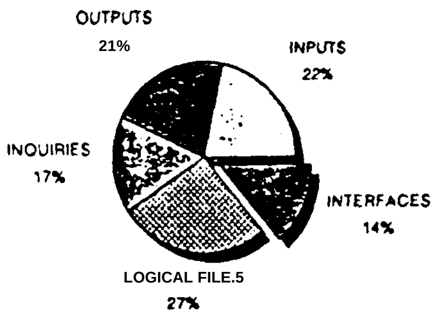
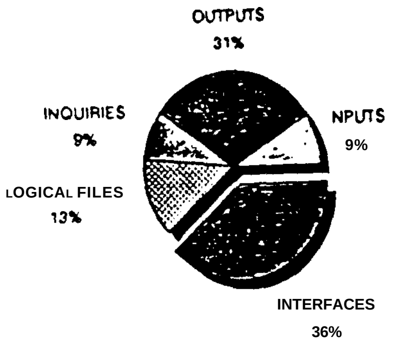


Figure 2-A and 2-B

FUNCTION POINTS INSTALLED BY TYPE
MAJOR ENHANCEMENTS



These metrics allow yearly comparisons of the number of software functions delivered. The Function Point count is time-independent and does not require annual adjustments for the output measurement process. These Function Point metrics constitute the yardstick against which we measure the impact of the introduction of changes (tools and techniques) to our software delivery process.

The next set of ratios developed is the relative distribution of function types per project. These project ratios and graphs offer valuable insights both for I.S. management and for our customers, into the structure of the systems either requested or delivered. They provide a vivid picture of both the functionalities delivered and those delayed or postponed to later phases. Newly developed applications differ significantly from mature applications, and, the respective projects will show quite different point distributions (see Figures 3-A and 3-B) In fact, with enough historical data, we should soon be able to come up with time-series distributions of applications based on their life-cycle in production (0 to + 15 years).

PROJECTS NET UNIT COST
PER FUNCTION POINT

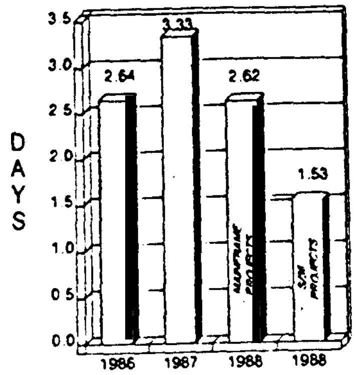


Figure 3-A

PROJECTS
AVERAGE COST

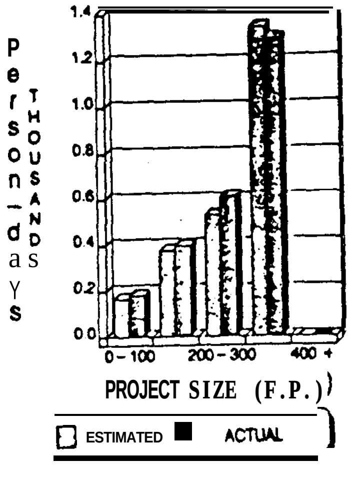


Figure 3-B

3- PRODUCTIVITY ANALYSIS

In addition to Function Point metrics, we collect a significant amount of data on a variety of productivity factors for each project:

- project scope
- development environment
- production environment
- staff experience
- project management, etc.

Altogether, we collect information on nearly fifty productivity factors; however, the productivity factor metrics are far from being as well defined as the Function Point metrics. We have, therefore, experienced significant difficulty in analyzing the impact of these productivity metrics within our own development environment and in coming up with factors specific to this environment. In the meantime, we are closely monitoring the industry-research results published at the International Function Point Users Group meetings.

4- INDUSTRY RATIOS COMPARISONS

A key factor in Montreal Trust's decision to adopt Function Point metrics was that it be in the public domain and not tied to the methodology of a particular software vendor. This would guarantee the availability of industry-wide historical data against which we could compare our productivity performance.

In June 1988, for example, Computerworld magazine published a special issue on productivity in which Mr. Capers Jones presented a set of productivity metrics based on Function Points in addition to data for the top 5% of U.S. companies.

Refer to Figures 4-A and 4-B 6 for Montreal Trust ratio and available industry data.

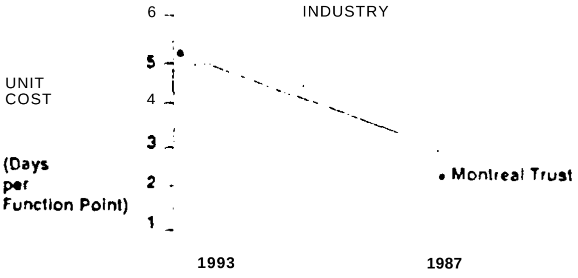
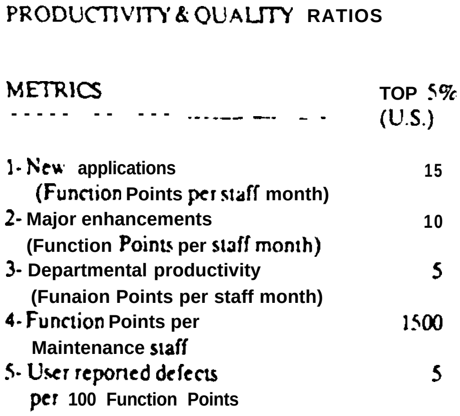


Figure 4-A



PRODUCTIVITY & QUALITY RATIOS

METRICS	TOP 5% (U.S.)
1- New applications (Function Points per staff month)	15
2- Major enhancements (Function Points per staff month)	10
3- Departmental productivity (Funaion Points per staff month)	5
4- Function Points per Maintenance staff	1500
5- User reported defects per 100 Function Points	5

Ref. Capers Jones. Computerworld, June 1988

Figure 4-B

5- SOFTWARE DEVELOPMENT MEASUREMENTS: BENEFITS

In the business and engineering fields, measurements enable us:

- to understand processes,
- to evaluate tasks and products,
- to control and improve processes,
- to predict product and process results.

While we are the first to admit that it requires a considerable amount of time and energy to implement and maintain a software development measurement program (but much less than that required for any accounting system), we already have derived some significant benefits.

5.1- CORPORATE CREDIBILITY

The management of the software development process with a measurement system that relates to the corporate culture and the corporate business is generating serious interest at the senior management level of the corporation. It gives us significant credibility as a corporate business unit that can contribute and be managed like any other business unit, and provides senior executives with the confidence that there is in place within the software development division a management system to monitor productivity.

5.2- IMPROVED DIALOGUE WITH BUSINESS USERS

At the business unit level, the use of Function Point metrics has been of major benefit to us in the way we manage our dialogue with the user community, which is based on concepts that they can understand:

- the number of functions requested and delivered,
- the types of functions delivered,
- the relations between the computerized functions and their business functions.

By identifying, and measuring, the functions requested and delivered, we are now in a much better position to demonstrate to our user community the complexity and the scope of their requests.

For example, when users request the addition of a function to an existing software application, we can now measure and demonstrate to them, using a clearly defined and understood metric process, that they are not only requesting a few free-standing input screens and a few reports, but also a specific number of functionalities related to their business data elements and, most importantly, that their business requirements also include: a quantifiable number of functions related to interfaces with existing components of the application and this is the specific price to pay for these requested interfaces.

The proper use of the Function Point metrics conveys these messages very accurately and has allowed us to alter the dynamics of managing the users' interaction with the systems professionals during the project management process: we are now in a position to focus the discussions on the number of functions to be delivered, as well as on their distribution. In the past, project estimates would be based on a traditional list of I.S. tasks and activities and productivity factors, and presented as such to the users to document project costs. This approach often placed us into the defensive position of having to justify our processes and discuss complex technical issues with the users. Very often this led to increased uneasiness on their part.

With the benefits of a well-established measurement process and a historical basis for the unit cost calculation, the project estimates presented now to the business customers relate only to the number of functions requested. There is now a very clear relationship between the project cost and its scope. When the estimates are higher than the users' expectations, we can now focus the discussions on the scope of the project and the areas where priorities have to be assigned as well as on the types of functionalities that can be postponed to later phases of the project. These are the areas that are under the users' control and in which they have to make business decisions. We have, in effect, handed back to them the type of project decisions that they are best equipped to make.

Discussions on I.S. development processes stay within the I.S. group and it is our sole responsibility to improve these processes and meet our business obligations.

5.3- FIXED-PRICE PROJECT'S

The implementation of Function Point metrics has also enabled us to meet a critical requirement of our business community: a fixed-price policy.

Each project is based on a cost-benefit analysis and there are corporate mechanisms in place to verify that the projected benefits are realized. Therefore, there is considerable pressure on the users to achieve this, and when they give the go-ahead for a project, the last thing they want is a cost overrun. Accordingly, we have geared our I.S. development organization towards a fixed-cost project pricing policy. At the present time, we are confident that with the management mechanisms in place we can meet our commitments. We may still bid too high or too low on some projects, but on a yearly average basis we expect to perform well.

However, we consider this to be only an interim phase. We are actively pursuing additional analyses to better understand and control our development processes. Also, we plan to invest in improving our processes in order to substantially reduce our costs. Our objective is not merely to lower our costs, but to meet our estimates. We are aiming to reduce our own risk as well as the risk to our customers.

CONCLUSION

The use of Function Point metrics has, for the first time, made it possible for us to measure the deliverables of the software development process. It is the common denominator that enables the calculation of productivity ratios across applications, project teams and technologies; it is also time independent. For the first time, projects can be compared objectively and productivity analyses can be prepared with a solid foundation for inter-project comparisons.

Through the use of the management information derived from Function Point measurements applied across the board to all development projects, we have been successful in shifting our focus from cost management by man/days to management by number of deliverables, their characteristics and their associated cost structure based on our in-house historical costs curve. This has also enabled us to quantify the productivity ratios to be reached when new tools and techniques are introduced, and to audit the results of the productivity improvements once they are fully implemented.

Although very promising, Function Point metrics is still a young science and up to now it has been used mostly at the individual project level. Very few corporations have attempted to introduce them as their main measurement system. Therefore, very little has been published on setting up corporate economics ratios (although there is currently work being done in one of the working committees of the International Function Point Users Group).

This has not come easily, however, and we have invested a great deal of time and energy in this process. When expertise was not available in the marketplace, we went ahead anyway and did the required research and development. We fully believe that it has the necessary foundation to be applied to a wider range of applications and processes. Whenever we have applied its concepts to our traditional areas of software work, we have been successful in deriving the economics ratios to enable us to perform the appropriate type of management analysis.

The implementation of Function Point metrics has brought us back into the mainstream of the corporate culture and enhanced our corporate image as good managers.

References:

AJ. Albrecht et J.E. Gaffney, “Software function, source lines of code and development effort prediction: A software science validation:, IEEE Transactions on Software Engineering, vol SE-9, pp. 639-648, Nov 1983

J.M. Desharnais, :‘Analyse statistique de la productivité des projets de développement en informatique à partir de la technique des points de fonction”, Master degree thesis, Université du Québec à Montréal, Dec. 1988

Capers Jones, "Building a better metric”, Computerworld Extra, June 20, 1988, pp. 38