Function Points Usage in Contracts – Considerations and Guidelines

Luca Santillo

Independent Consultant, GUFPI-ISMA Board Member,

luca.santillo@gmail.com

Abstract. Since function point-like metrics cannot measure quality or technical aspects, the amount of Function Points is not sufficient to estimate a software project true cost. For long-term contracts, the average "cost per Function Point" could be extremely under- or over-estimated – from one project to another. Unfortunately, the seek for easiness and speed has privileged the spreading of fast project pricing mechanism – e.g fixed price per Function Point. A group of experts from the Italian Software Metrics Association issued a set of guidelines for customers and developers to agree upon the best usage of Function Points as a variable in negotiation and auditing of software development. The main factors affecting productivity are traced, as software reuse, change requests and quality, while trying to keep easiness and application flexibility.

Keywords: function point, contract, software, price, cost, guidelines.

1 Introduction

Function Points have been spreading in the software field since late 70's as an effective measure for functional software size [1]. Since such measure is not linked to any quality or technical aspect, it should be clear to practitioners that the only amount of Function Points for a software project is not sufficient to evaluate the right effort, duration and finally cost for the given project. For long-term contracts, this means that an average cost (price) per Function Point can be found to be extremely under- or over-estimated with respect to the real values of software development efforts, from project to project. Unfortunately, the seek for easiness and speed in software measurement and supply has privileged the spreading of fast pricing mechanism – the simplest and most dangerous being the fixed price per Function Point – in software development and enhancement projects.

In June 2006, GUFPI-ISMA (Gruppo Utenti Function Point Italia – Italian Software Metrics Association) issued a public domain technical report [2] comprising a set of guidelines for customers and developers to be able to agree upon the best usage of software measures, namely Function Points, as a variable of contractual negotiation and supplying auditing for ad hoc software development. A large number of practitioners – from both the customers' and the developers' side in Italy – and subject matter experts (including the author) did provide the national ICT community with a up-to-date analysis of do's and dont's, supply classes and adjustment aspects

for usage of function point-like metrics in contracts. The main factors affecting productivity are traced in the guidelines, as software reuse, change requests and quality. Still, an attempt has being made by the guidelines' authors to provide easiness of use and flexibility in the application of the proposed approaches.

Following over ten years of Function Point usage in large-agreement environments in Italy, this first version of the GUFPI-ISMA guidelines summarizes the experience of practitioners and organizations in the field. This work aims to outline the guidelines' contents and concepts for the international audience, while stimulating the discussion for further improvements of the documented approaches.

2 Guidelines' Contents

Besides a proper introduction, where the main purpose, scope and involved roles are presented in detail with a domain-specific glossary of terms, the guidelines by GUFPI-ISMA consist of a main section, entitled "Function Points in Contracts for Ad Hoc (Custom) Software", and a series of appendices for in-depth presentation of specific subjects and hints for practical application. The main section of the guidelines tackles four areas:

- **Items to consider for a Function Point-based contract**
  - Supply classes
  - Lifecycle models and functional measurement moments
  - Supply scope
  - Software requirements types and detail levels
  - Change requests in the software lifecycle (incl. interrupted projects),
  - Pricing mechanism for interrupted projects
  - Price impacting factors
  - Base contractual schemes
- **Measures management in a custom software supply contract**
  - Call for tenders
  - Software requirement analysis
  - Software product implementation and final inspection
  - Software assets size update
  - Final definition of contractual pricing value
- **Some application models for described items**
  - Fixed price model for specified products/solutions
  - Variable price model for not-fully specified products/solutions
  - Variable price model for products/solutions to be specified
- **Pricing mechanism for custom software products supply**
  - Fair price definition
  - Quality of external benchmarking data
  - From effort to cost to price
  - The issue of moving average value

Appendices are provided about the following main topics:

- **Software functional size estimation methods**
  - Extrapolated measures

  - Sampled measures
  - Average complexity method
  - KISS method
  - Early & Quick Function Point
- **Examples of software process models and selection criteria**
- **Examples of effort percentages for software process models phases**
- **Software contractual value impacting factors**
  - Productivity adjustment factors
  - Contractual Functional Measure (MFC)
  - Reuse impact on contractual value
  - Replication impact on contractual value
  - Change request impact on contractual value
  - Quality measurement

The latest items comprise the most innovative concepts provided by the guidelines for a practical application of the concepts outlined in the main section, with respect to the current status of contractual usage of functional size, at least at the national level. Since describing every item in the guidelines is beyond the scope of this work, next sections addresses some of the concepts introduced or clarified by the guidelines.

3 Guidelines' Concepts and Clarifications (excerpt)

Supply classes. It is clarified that, in their current version, only the supply classes of "ad hoc (custom) software development" and "ad hoc (custom) software enhancement" are addressed by the guidelines. In this context, several types of maintenance activities are described, to help practitioners distinguish the proper domain of application. Hints or explanations are provided on excluded types, as "non functional enhancement" or "corrective maintenance".

Lifecycle models and functional measurement moments. Common models are included and described, as waterfall (with/without prototypes), incremental, iterative. So-called agile or extreme models are excluded in the current version, since they seem hard to apply in a contractual framework at the moment. For each included model, a table is provided describing when functional measurement can/should be applied within the given model process, with a rationale and estimated accuracy. Table 1 reports the example for the waterfall lifecycle model.

Software requirements types and detail levels. Following ISO classification, requirements are classified as *functional*, *technical* or *quality* requirements. On detail levels, it is clarified that the method applied to obtain the size must be explicated, since in certain stages for multi-year contracts no standard methods can actually be applied to high level project descriptions. Some estimation methods or techniques are provided in a related appendix (see section 4 for an excerpt).

Table 1. Determining Function Point measures in the waterfall lifecycle model.

When	How	Why	Accuracy (error)
Feasibility Study Completed (System User Requirements)	Functional size estimation methods	Provide essential information for decision making, feasibility test and project planning	Likely ±20-20%
Logical Design Completed (Architectural Design)	Standard methods (FP)	Provide the baseline for any further variation, change request or scope creep	Often <10%
On demand, along the lifecycle	Estimation or standard methods	Provide support to change request measurement and management	Depending on detail level
At completion	Standard methods (FP)	Provide contractual check of requested vs. delivered	Often <5%

Change requests. Since change requests can cause relevant delays or even failure of the project, this topic is described and addressed in detail by the guidelines, in two steps. For "interrupted" projects, a formula is provided for pricing approach to functionality that has been already analysed, designed, and or implemented when the project is stopped, as follows:

$$\text{Reduced Price} = \text{FP} \times (\text{Unit Price} \times \text{Performed phases cost percentage}) , \quad (1)$$

where the parenthesis stress the fact that conceptually the price adjustment is applied to the unit price, based on the percentage of the work effort performed before the project has been stopped, rather than to the Function Point amount.

A further step for more accurate determination of change request impact is provided in a related appendix (see section 4 for an excerpt).

Price impacting factors. A sample of impacting factors, suggested by several research and industrial models, is provided. A first classification of the factors is suggested based on product, process, technology, and staff domains. Clarification is explicated on the fact that such list is not exhaustive, and that different models can denote one factor with different terminology, or on the opposite the same terminology in different models could refer to different factors from a conceptual point of view. The listed factors are then mapped on a Relevance/Measurability diagram, with scales from low to high impact relevance or strength on pricing values and ease or capability of measurement. Such diagram can help project/program managers to identify the most significant factors to measure, manage and (possibly) control. Figure 1 reports the cited diagram (a caption describing the low/average/high levels for relevance and measurability is provided in the guidelines – hereby not reported).

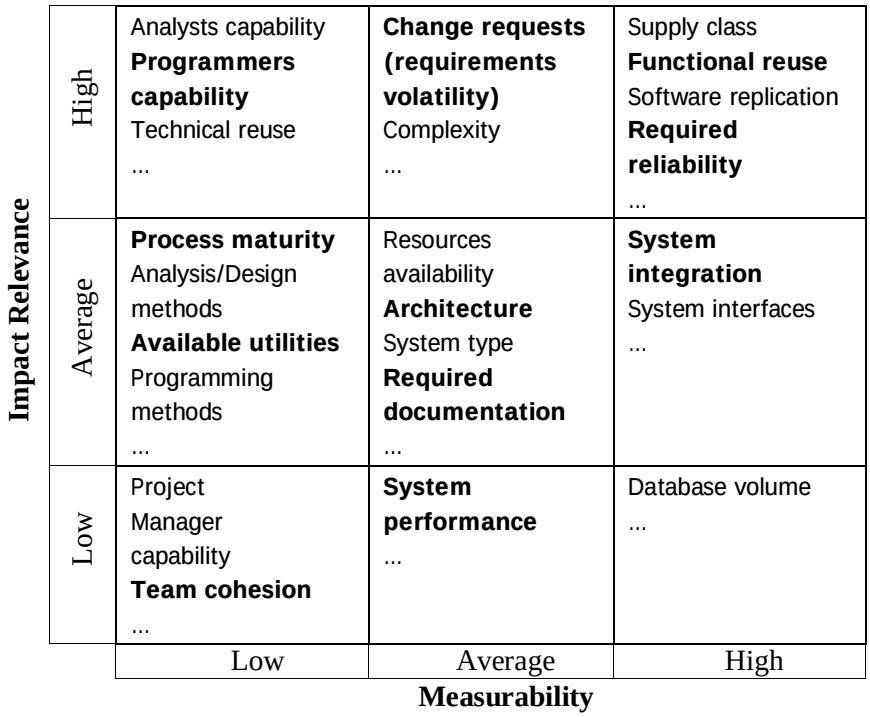

Fig. 1. Impacting factors map, with respect to the capability to measure them and the relative impact they can have on the pricing mechanism (excerpt sample).

Base contractual schemes. Four models are identified and briefly described for contractual frameworks:

- Body Rental
- Time & Material
- Measure-based
- Product-based

For each type, a definition, risk elements, pricing definition mechanism, change request treatment and warranty possibilities are provided.

Body Rental and Time & Material types are not further developed or addressed in the guidelines. For the latter types, application models are suggested in a specific subsection of the guidelines (see section 4.1 for a brief excerpt).

Measures management. Another aspect discussed in-depth by the guidelines is the estimation or measurement activities related to the generic steps that can be identified in any contractual framework development process. For instance, main stages of any "contract process" are listed as:

- Pre-contract start-up (needs definition and call for tenders)
- Contract execution (several phases, typically related to the software process itself)
- Contract closure (assets update and economical check-out)

For each stage, estimation or exact measurement of the functional size of software products are proposed where applicable.

4 Guidelines' Suggested Approaches (excerpt)

This section reports on some of the approaches suggested by the guidelines to address the main aspects and concepts previously mentioned.

4.1 Some (contract) application models for described items

The models considered are:
- Fixed price model for specified products/solutions
- Variable price model for not-fully specified products/solutions
- Variable price model for products/solutions to be specified.

For each model, the following aspects are discussed in the guidelines, obviously with different results from one model to another:
- general consideration
- initial definition of prices
- modalities for software metrics auditing
- modalities for change request management
- modalities for definition review of prices
- possible extensions to the main model.

The most developed model is the third one (variable price for products/solutions to be specified), of particular interests for mid to long-term contracts, where the initial setup of the contract is simply not able to define in advance all the projects and – for each project – the exact requirements that will be specified during the contract execution (usually, spanning over a time period much longer than one year).

In the mentioned model, the initial definition of prices section suggests to consider a derived (indirect) measure, denoted as Contractual Functional Measure (MFC, Misura Funzionale Contrattuale) as the product measure to relate to the unit price in the contract. Two variations are possible to combine the relevant impacting factors:
- project class-based approach
- adjustment factors multiplication-based approach

In the project class-based approach, the contract should define a (reduced) set of project types (classes) based on the most common and realistic combinations of impacting factors, and a corresponding set of unit prices per function point per each project class.

In the adjustment factors multiplication-based approach, in a similar way derived by COCOMO-like models [3], the contract should define a set of impacting factors and a scale of adjustment values per each factor over the pricing value.

(In reality, it is possible to prove that the two approaches are equivalent one to each other, under general conditions.)

Figure 2 and Figure 3 show the overall models for initial price definition for a given project under the discussed contract model (variable price fro products to be specified).

Besides injection of, respectively, project class values and factor adjustment values, both models include reuse and replication concepts to derive the Contractual Functional Measure. Reuse and replication are further described in corresponding appendices in the guidelines; further aspects, as intrinsic complexity, could be addresses in future versions of the guidelines, provided that double consideration is avoided – any factor explicitly considered in the MFC measure must not be also considered to derive a project class or as an pricing adjustment factor.

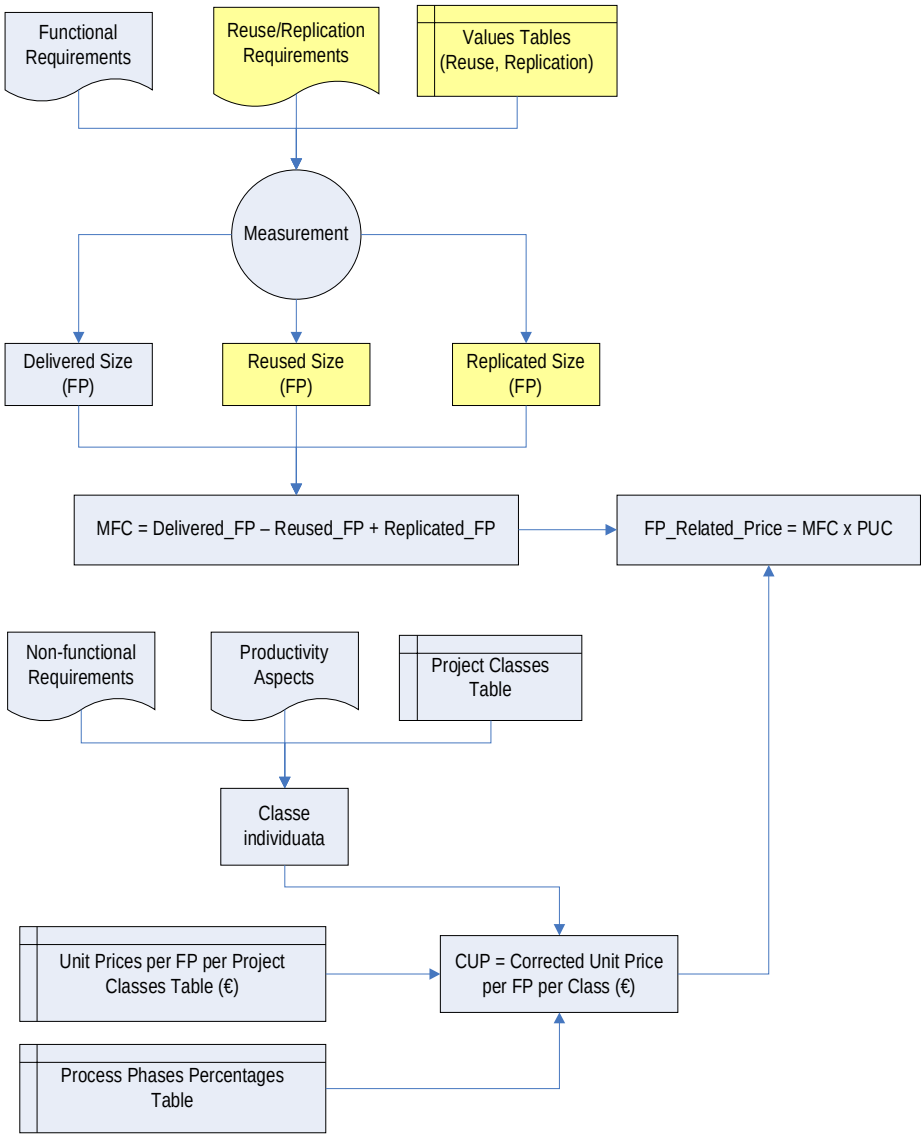

Fig. 2. Contractual Functional Measure (MFC) diagram, based on project classes.

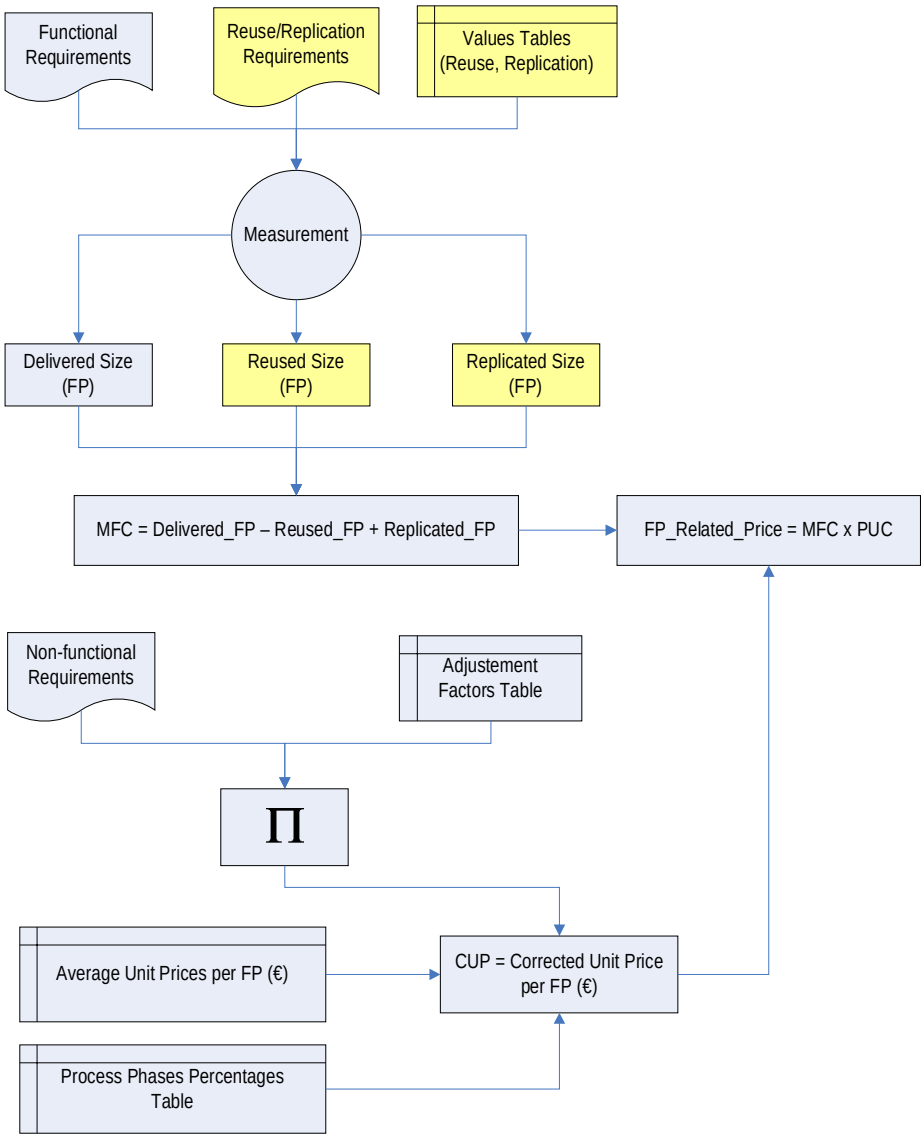

Fig. 3. Contractual Functional Measure (MFC) diagram, based on adjustment factors. The upper portion in as in Figure 2. P stands for the "product of *n* given factors values".

4.2 Pricing mechanism for custom software products supply

A discussion is provided in the guidelines about the issue of a "fair price definition", where practitioners need to fix an initial price for a given project class, or the average price per Function Point. Three main questions are discussed, namely:

- the quality of external benchmarking data
- the translation from effort to cost to price
- the issue of moving average value

Some hints are provided in the guidelines on such questions.

4.3 Software functional size estimation methods (Appendix)

Several techniques are reported, to obtain an early approximation of functional size, when needed. A description of six levels accuracy definitions for size measures is also provided, based on ISBSG publications [4]. A set of references is set, pointing to detailed descriptions of the suggested techniques (e.g. [5] for Early & Quick FP).

4.4 Examples of software process models and selection criteria (Appendix)

A list of phases, standard deliverables, and phase closure criteria is provided, for five development models (traditional/full, traditional/reduced, unique phase, knowledge-based, object oriented. Such lists can serve as a template for real contract definition.

4.5 Examples of effort percentages for software process phases (Appendix)

Based on public sources spanning over the last five years, some effort percentages values are reported, to help customers and developers negotiate their own values in real contract definition.

4.6 Software contractual value impacting factors (Appendix)

Several aspects are discussed in-depth in this part of the guidelines. For instance, a list of productivity factors, together with a possible value scale is proposed deriving from COCOMO models [3]. Contractual Functional Measure, Reuse and Replication impact factors are further discussed, since they play a primary role in the models previously reported (e.g. refer to Fig. 2 and 3).

On Change Request, a specific section provides a method and a formula to determine the size of a change request in terms of what is added, changed, and deleted with respect to the initial baseline measure. The change request measurement can serve as both a way to correct the Contractual Functional Size, if approved, in the final pricing mechanism of the given contract, or as an indicator of the overall requirements volatility, to explain delays or rework efforts, and to derive a specific productivity factor.

Finally, an initial selection of quality aspects is provided, as a trace for practitioners to consider, and for further evolution of the guidelines on such aspects.

5 Conclusions

As pointed out by the previous sections and excerpts, several distinct aspects of the usage of Function Points in contractual frameworks are discussed and proposed for suggested solutions in the current first version of GUFPI-ISMA guidelines. The main challenge of such work is to merge such aspects in an organic and systematic view, while keeping ease of use and flexibility for practitioners to manage different types of contracts in a consistent way. Undoubtedly, a long path is to be covered to reach such goal. Nonetheless, thanks to the contributions of so many subjects among the authors and reviewers of the guidelines, this first global attempt at national level for Italy helps in making explicit what to consider as base elements for a proper usage of functional size measures in software contracts, and what to avoid (first release "do's and don't's"). A first positive, practical result of this work should come from the inclusion of many concepts and suggested approaches into another set of guidelines on product and services quality in public administration by CNIPA (the National Centre for Information Technology in Public Administration in Italy) [6], which is under study at the moment this paper is issued. Such study, together with industry tests and further experts contributions, will lead to a an improved version of the GUFPI-ISMA guidelines in the future. Finally, a comparison with other national bodies frameworks on the subject of contractual usage of software metrics would result in a enriched and more consistent set of globally accepted practices.

Acknowledgments. The author wishes to thank all GUFPI-ISMA members, researchers and coordinators who helped in providing his and others' contributions, and in merging them in a consistent way in the guidelines. A complete list of authors and reviewers of the guidelines hereby briefly discussed is reported in [2].

References

1. IFPUG: Function Point Counting Practices Manual, Release 4.2.1, International Function Point Users Group (2006)
2. GUFPI-ISMA: LGC-FP – Linee Guida per l'Uso Contrattuale dei Function Point, Versione 1.0, Documento Tecnico 2006/01, Gruppo Utenti Function Point Italia – Italian Software Metrics Association, Italy (2006)
3. Boehm, B.W. et al.: Software Cost Estimation with COCOMO II, Prentice-Hall (2000)
4. ISBSG: Practical Project Estimation, 2nd edition", International Software Benchmarking Standards Group, Australia (2005)
5. Santillo, L., Conte, M., Meli, R.: Early & Quick Function Point – misurare di più con meno, in: GUFPI-ISMA (ed.): Metriche del software, Esperienze e ricerche, pp. 196-210, Franco Angeli, Italy (2006)
6. CNIPA: Linee guida sulla qualità dei beni e dei servizi ICT per la definizione e il governo dei contratti della Pubblica Amministrazione, National Centre for Information Technology in Public Administration, Italy (2004)