The ABC Model: Coming "Back To The Future" in (ICT) Contracts

Luigi Buglione[1,2]

[1] DXC Technology, Via Paolo Di Dono 3a, Rome, 00142, Italy
[2] GUFPI-ISMA, Corso Giovanni Agnelli 42, Turin, 10137, Italy

Abstract

One of the main problems in current ICT projects is to determine the economic value of project activities using mostly (or solely) the deliverables produced, considering such projects as something repeatable and applying the 'economies of scale' principles. But looking to the inside of those projects, most of them are "artisanal" projects, unique to a specific customer for a specific need and many variables should be taken into account in order to provide in an estimation the right 'quantity' to be produced (also in terms of outcomes), effort and costs (to be translated into a final price) considering all the activities in scope to such project, not only those strictly devoted to produce the project deliverables. This paper will discuss the current situation and a simple but effective solution to such issues using benchmarking and data management best practices, overcoming also bad outsourcing practices, called the 'ABC Model', providing an example with objective evidence.

Keywords

Contract Management, ABC Model, Requirements, Unit of Measure, Productivity.

1. Introduction

Each economic market has its own rules and peculiarities and thus it's difficult to apply the same, exact rules, principles, and best practices to another kind of business. Even if what said seems to be a 'golden rule', in ICT projects seems from many years that – whatever the kind of project, context, technology applied, economic conditions according to the country where such project will be realized, and more – there could be the possibility to apply 'economies of scale' to projects different each other only because the final main deliverable is a software product. Moving from the customers' need of asking more objective ways to determine the amount of work (that's effort, no matter if measured by man-days or man-hours) instead of using only (or mostly) effort and costs estimations by experience, the usage of measures was introduced in ICT Contract Management before using Lines of Code (LOCs) and then Function Points (FPs – several ISO standards currently active and available) as the 'objective quantity' that should have been used for giving to those 'units' an economical value (e.g. [1]). The more the units counted, the more the costs for a project and vice versa. This led to a huge variability of the unit price worldwide according to several factors, not assuring neither the customers nor the providers about a right balance about effort and costs because the debate was moved on the (technical) way to count or not some LOCs or FPs. The result was (as still often is) a delay in the delivery of such deliverables and/or a reduction of the effective quality in use perceived by the final end users when they use them in a service. Now, in 2023, all the elements to overcome these issues are available, applying the "**ABC Model**". This paper is structured as follows: Section 2 will describe the initial problem and what an improper application of the 'economies of scale' principle caused. Section 3 proposes the solution, describing the ABC Model and its four steps to be performed for normalizing the contract management in ICT contracts. Section 4 will provide an example of such application, showing the effective, practical, and global application of such simple and common-sense based approach. Section 5 will draw some conclusions and suggestions for next steps.

Proceedings Acronym: IWSM Mensura 2023, September 14–15, 2023, Rome, Italy
luigi.buglione@dxc.com (L. Buglione);

2. The Problem (Yesterday and Today)

The starting point in IBM for asking in 1975 to Dr. Allan Albrecht to create what was named Function Point Analysis (FPA) in 1979 was to overcome some drawbacks when using Lines of Code (LOCs) from a technical and economic viewpoint. Till the '70s, LOCs were used as the main measure in ICT projects for 'sizing' a software, being a sort of quantitative unit of measure, but unfortunately not sizing the number of functionalities to produce, but only the number of executable LOCs for producing such functionalities. But they were easy to be counted. Thus, their "productivity paradox", as stressed a few years later by Capers Jones, was that paradoxically the more the LOCs with an older technology/programming language, the more the effort and final price related to that software project but the better in terms of productivity (LOC/effort) and unit cost (LOC/monetary unit) that using a more modern technology/programming language that would reduce the overall effort and costs but would paradoxically increase the unit price and the unit cost, being less the formal number of LOCs.

That's why Function Points (FPs) started to make a step beyond this issue, because they cancel this paradox because the unit cost (FP/monetary unit) and the productivity (FP/effort) – being the number of FPs the same even such functionalities were (approximately) implemented with different technologies. This example was presented here [2] for clarifying with numbers this first achievement.

But what at that time were not considered enough was a couple of things:

- FPs and LOCs are <u>product</u> measures, <u>not</u> measuring the whole <u>project</u> (that's a larger container that the products/outputs/deliverables, that can be more than the solely software product, e.g. a user manual), as logically shown in [3];
- User Requirements (URs) are not only **FURs** (Functional User Requirements) – that's the basis for sizing FPs - but can be classified also as **NFRs** (Non-Functional Requirements) and **Project constraints**, not contributing to the FP count (whatever methodology, according to ISO/IEC 14143-x family principles and rules) but contributing in the typical way to consider the 'productivity' and 'unit costs' formulas. This classification was proposed before by the 'ABC Schema' [4] and a few later the COSMIC/IFPUG Glossary of Non-Functional Requirements [5] (Figure 1).

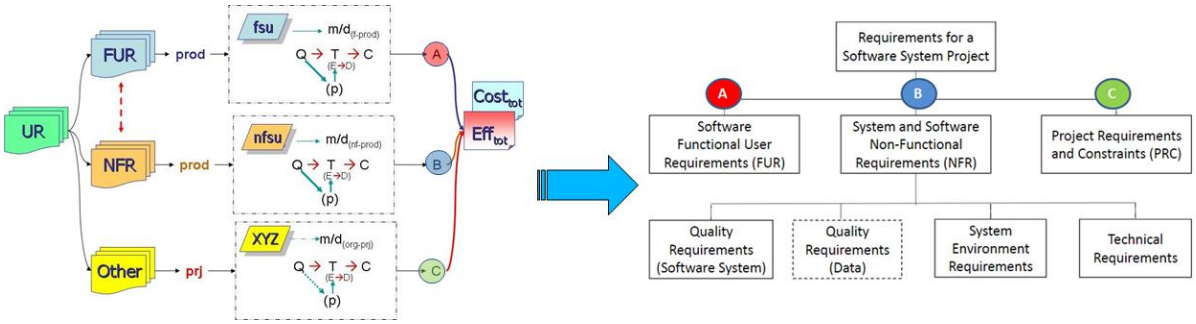

Figure 1: The ABC Schema [4] and the COSMIC/IFPUG Classification of Requirements [5]

Considering those couple of 'issues' mentioned before, what happened along time was that FP (that's simply a quantity) became a 'price' even if each software project – that's original each time –changed requirements and thus functionalities and the final deliverables and its inner, related quality – cannot be considered as a sort of 'repeatable product' with a sort of 'standard' cost (for the providers) and price (for the customer). The "price per FP" were including ALL the project costs, <u>not only</u> the ones related to the deployment of FURs but also what related to non-functional requirements (NFRs) and project-related activities (PRJ) plus all the fixed costs (e.g. travels, licenses costs, etc.), simplifying too much the way to calculate project economics. In fact, as in Figure 2, in a typical development project the most of the effort (A-type; FURs) is deployed by the project roles costing less (e.g. analysts-programmers-testers), followed by the B-type effort (NFRs) performed by professionals costing a bit more than the A-type ones (e.g. systemists, DBAs,

architects, …), with less working hours but costing more, till the C-type effort (PRJs) performed again by professionals (e.g. project managers, service managers, measurement specialists, team leaders, …) costing more than the previous two groups [6].

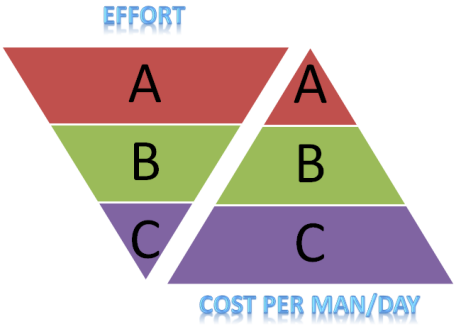

Figure 2: Effort and Cost/M-d Triangles

Again, the two more diffused FSM-based standard methods, IFPUG (ISO/IEC 20926) and COSMIC (ISO/IEC 19761) were thought to be in a 1:1 relation applying their own unit of measures, creating further confusion in the market among different units of measures expressing different efforts and costs. Some sources would like to propose the "price per FP" as paid a lot of USD/FP (e.g. [7][8]) in some cases more than 1,000 USD/each (as in [9][10]), while in Italy, that's one of the countries where FPA (IFPUG and COSMIC) are typically asked in Public Administration contracts and bids and with more certified counters, has raised down till 80 EUR/FP or less. Even if it's possible to averagely produce 2 FPs/m-d, if the price/FP would be 80-100 EUR/FP, the risk for providers is to go under cost (it'd mean that an average daily rate should be c.a. 200EUR/day), that's lower than the internal daily cost for a typical provider company, thus generating a loss and not a revenue or a tie. This would lead to the potential engagement of less expert people in the producers/providers' teams, reducing the final product quality and/or creating possible issues about the expected delivery dates, that could be not respected. And in this current 'digital age' the end users would be the stakeholders suffering more from low quality ICT systems not properly working with continuity. Outsourcing projects to low-cost countries is not necessarily a solution for overcoming economic issues (see [11] for a deeper discussion with examples). A further observation for ICT projects is about the case of the '*zero FP*' projects (no FURs are included in the project scope, only NFRs and PRJs) where paradoxically a price per FP would be equal to *null*, also in the case where NFR/PRJ requirements would be expressed – as in the past – as 'value adjustment factors' (zero FPs multiplied by any number – even if high - returns always…zero!).

Thus, the basic question is still: are we sure that a "price per FP" is the proper way to continue to follow for pricing ICT contracts? The question is very relevant because several companies – no matter is customers or providers – are concretely thinking that contractual issues would be related to measures and not to the (bad) way measures were (and are still) applied to contracts…functional product measures cannot size the entire project scope and daily costs cannot be the same in different countries for any type of software…development and maintenance and enhancement projects have different productivity levels…thus which could be a possible general solution valid for ALL projects, not only in the ICT domain?

3. The Solution (Tomorrow)

As said, a project plan is based on an affordable effort and duration estimated based on a solid measurement program. Paying a project by the expected amount of 'units' – where each project is new and not the exact copy for another previous one – cannot be the solution. The solution, from our viewpoint and experience, as in Figure 3 can be expressed by the "**ABC Model**", that's simply a way to "*come back to the future*", with a 4-step flow proposing a series of yet existing techniques and criteria but in an organized and logical way. Note: "ABC" is not an acronym bit a title for expressing a simple, basic but effective way to manage estimations citing the three first letters of the alphabet.

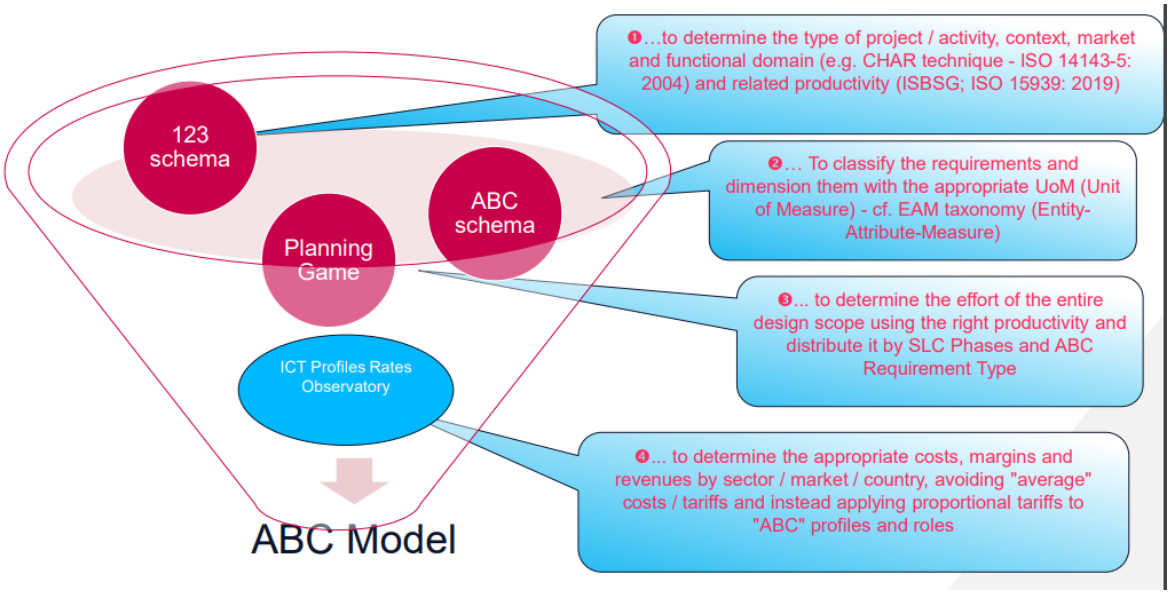

Figure 3: The ABC Model and its four (4) steps

The **Q→T→C** (Quantity → Time → Cost) logical flow from the "ABC Schema" moves from the quantification of user requirements (UR) into **measures** by the three streams (A-FUR, B-NFR, C-PRJ) that – divided by the proper nominal productivity value(s) – will drive to derive the final project effort. The whole project **effort**, after creating the project plan and schedule, will drive to the right calculation of costs, margins, and final prices according to the real project schedule and the other project constraints to be considered for the specific case [6]. Thus, linking directly quantities to costs/prices is not the right solution, being 'Time' the variable that should drive to a proper calculation of the final costs/prices.

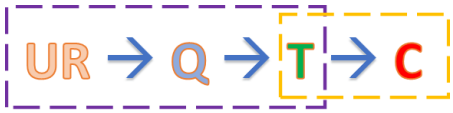

Figure 4: The Q→T→C flow: from UR to T (Time) and from T (Time) to C (Costs/Prices)

4. An example

4.1. Initial Assumptions

When applying the steps from the "**ABC Model**", here some assumptions:

- **Step #1 – Determine the type of project/activity/context…**:
  - Project Phase: Development [12][13]
  - Functional Domain: Management Information System (MIS) [14]
  - Technology/Programming Language(s): COBOL or PowerBI/PowerApps (to be compared)
- **Step#2 – Classify Requirements and choose the proper UoM(s)**:
  - IFPUG FPs are the main quantitative driver for using 'nominal' productivity values from ISBSG Development & Enhancement (D&E) repository [13] sizing the product (*entity*) functional (*attribute*) size according to the EAM (Entity-Attribute-Measure) taxonomy [15];
  - Product Functional Size: 1000 IFPUG FPs v4.3.1 (*unadjusted*)
- **Step#3 – Determine the effort of the whole design scope using the right productivities per SLC phase and ABC requirement type**:

- Nominal productivities:
  - COBOL: c.a. 0.6 FP/m-day
  - PowerBI/PowerApps c.a. 4.5 FP/m-day
- Average cost per day for a mixed team
  - Italy: c.a. 300€/day (e.g. for a project worked in Italy/Europe – of course let's change values when the project would be requested to be deployed in India, Brazil or other countries with a different economy). A more precise exercise about costs and prices could be done using the '*Planning Game*' technique, as in [12].

- **Step#4 – Determine the appropriate costs, margins and revenues (by context):**
  - Price/FP: 100 EUR/FP

Note: the example is using values referred to IFPUG FPA v4.3.1, since currently the FSM method mostly used in Italy in Public Administration contracts and Euros (€) as the related currency. But the same exercise can be done with any other FSM method, using its own productivity/PDR specific data and other currencies. To present the three hypotheses to be compared, let's assume the 2023 daily team mix costs for COBOL [16] and PowerBI/PowerApps [17] projects, respectively an average team mix cost of 345€/m-d and 303€/m-d. Here the detailed calculation, splitting the overall project effort by requirement types, according to the 'ABC Schema'.

Table 1
COBOL Programmers – Cost/m-d (2023) (elaboration from [16]) → 'A' Requirement-type effort

Median (USD)/year	USD/€	Median (€)/year	working days/year	Median (€)/day
82,547	0.91	75,118	220	**341**

Table 2
COBOL – Efforts and Daily Costs (by requirement types/professionals):

Req.Type	Effort (m/d)	%Effort (ABC)	DailyCost (€)	Tot Costs (€)	Avg Cost (€/day)
A (FUR)	1,667	68%	**341**	386,467	
B (NFR)	1,667	20%	350	116.667	
C (PRJ)	1,667	12%	360	72.000	
				575.133	**345**

Table 3
PowerBI/PowerApps Developers – Cost/m-d (2023) (elaboration from [17]) →

Median (GBP)/year	GBP/€	Median (€)/year	working days/year	Median (€)/day
52,500	1.17	61,425	220	**303**

Table 4
PowerBI/PowerApps – Efforts and Daily Costs (by requirement types/professionals):

Req.Type	Effort (m/d)	%Effort (ABC)	DailyCost (€)	Tot Costs (€)	Avg Cost (€/day)
A (FUR)	222	55%	279	42,191	
B (NFR)	222	32%	350	15,556	
C (PRJ)	222	13%	360	9,600	
				67,346	**303**

4.2. Three Hypothesis

4.2.1. Hp1 - Cost driver: only FPs

The first hypothesis is to use only FPs as the solely cost driver.

Table 5
Cost driver: only FPs

Funct Size (FP)	Tech/PL	N_Prod (FP/m-d)	Exp_Eff (m-d)	Price/FP (€/FP)	Overall Price (€*FP)	Exp_Price /m-d	Market Price / (€) m-d	Difference / (€/m-d)
1000	COBOL	0.6	1,667	100	**100,000**	60	345	**-285**
1000	PowerBI/PowerApps	4.5	222	100	**100,000**	450	303	**+147**

The difference per m-d would reveal a negative difference for a COBOL implementation, while an extra-cost for a PowerBI/PowerApps. Note that productivities cannot be assumed always but must be set up using organizations' historical data. Assuming a 'standard' productivity level, whatever the technology/PLs applied would lead to potentially large over/underestimates.

4.2.2. Hp2 - Cost driver: only m-d (avg daily rate)

Second hypothesis, as typically applied in many ICT contracts, is about the application of payments by a standard, average daily rate for the whole team mix, no matter the distribution of effort by requirement types, as in Figure 2:

Table 6
Cost driver: only m-d using an average daily rate

Funct Size (FP)	Tech/PL	N_Prod (FP/m-d)	Exp_Eff (m-d)	Avg Daily Price (€/m-d)	Overall Price (€*FP)	Exp_Price / (€/m-d)	Market Price / (€) m-d	Difference / (€/m-d)
1000	COBOL	0.6	1,667	300	**500,000**	300	345	**-45**
1000	PowerBI/PowerApps	4.5	222	300	**66,667**	300	303	**-3**

The difference per m-d in this case would be reduced, in both cases with negative values, but with smaller variabilities than in Hp1. The overall price for COBOL seems to be very high for a high number of man-days. Here the attention point would be to better explore the effort distribution by requirement types (and related FTEs), as in Hp3.

4.2.3. Hp3 - Cost driver: m-d (balanced daily rates using the ABC schema)

Last hypothesis uses a distribution of effort (and related daily costs) balanced by the ABC requirement types:

Table 7
Cost driver: m-d using a balanced daily rate (ABC)

Funct Size (FP)	Tech/PL	N_Prod (FP/m-d)	Exp_Eff (m-d)	% Effort (€/m-d)	Avg Daily Price (€/m-d)	Overall Price (€*FP)	Avg (€/m-d)	Exp__Price / (€/m-d)	Difference / (€/m-d)
1000	COBOL	0.6	1,667	A=68%; B=20%; C=12%	A=341 B=350 C=360	**500,000**	345	345	**0**
1000	PowerBI/PowerApps	4.5	222	A=55%; B=32%; C=13%	A=279 B=350 C=350	**67,346**	303	303	**0**

This last scenario, considering the context (country, economical parameters, business sector, etc.) seems to be the best one, using one or more UoMs as the input for determining efforts and costs and not directly 'quantities' as the main drivers, not being such kind of projects 'repeatable-at-all', thus applying a sort of 'economy of scale' mechanism.

As said in the FP arena, FPs are a sort of "square meters", but they cannot have a 'standard cost/price'…it'd be like to affirm that the cost/square meter for a floor would be the same when choosing between different materials (e.g. marble, tiles, parquet, …). In our analogy, the materials are the ICT project technologies, and they strongly impact on the final project efforts and costs.

Another variable to mention – of course – is the expected duration for the project and many other variables related to the project entity, not the product and not again only about its functional side. Even if the price per FP would be very high (e.g. 1000+ €/FP) in order to satisfy the economic side, the effort related to NFRs and PRJ related requirements (the B/C types in the ABC schema) would represent a significant effort making difficult to accomplish established deadlines. Finally, professionals dealing with some old fashion/legacy PLs can/could cost per day more than more modern and productive PLs and it could create a further paradox when paying a project "per FP" and not "per m/d".

Thus, why still using a "price per FP" and not – "*coming back to the future*" a price/m-day (or price per m/hrs) derived from a proper usage of measures and related historical productivities for a certain software domain that will drive to the right effort approximation to be used for planning and scheduling the project? Of course, the daily team cost should be applied according to the place where it will be worked (e.g. India, Brasil, etc.), not looking to other countries' economies and daily tariffs.

5. Conclusions and Next Steps

"*You cannot control what you cannot measure*" is a well-known motto by Tom Demarco [18] and should be very clear from the beginning for a project estimator what to measure, possibly applying the **EAM** (Entity-Attribute-Measure) taxonomy [19] and creating a measurement plan containing measures what could properly monitor the different projects aspects and sides. "**Quantity → Time (Effort and Duration) → Cost/Prices**" is the logical flow that should be always followed and since typical ICT projects are not producing 'standard' pieces of software, thus it's not possible to follow any 'economy of scale' rule. The final cost and related price must take into account not only the WHAT is requested to be produced but of course also the HOW, that's fundamental and not be skipped. The "**ABC Model**", called in such way because supposed to be simple and effective as remembering the first three letters of the alphabet, can be a way to normalize the way (ICT) projects are currently managed, stopping the application of a fixed price/FP if FPs would size and be related to the whole project scope and effort. It's applicable to any kind of project, in any country, in any moment. What should be priced is the needed effort and the value brought out from a project, not only or mostly by its size (that's only one of the inputs for producing the final value to users and other stakeholders).

The drivers on which betting more for improving good contract management practice will be:

- proper and continual project data gathering that will assure updated productivity data over time (not assuming external, uncontrolled data), as asked also by the ISO 15939 measurement process [19];
- proper application of nominal productivity values from trustable sources referrable to the right software domain (e.g. applying the CHAR technique from the ISO 14143-5 standard [14]);
- updated tariffs for paying one man-day (or man-hour) in the country/ies where the project will take place, fundamental for making it sustainable and allowing it won't fail over time. Time (effort) is the target to be properly estimated and paid, not the quantity (FP) when a project like ours <u>is not</u> a mass production.

According to the ITIL4 glossary [20], 'value' is "*the perceived benefits, usefulness, and importance of something*", including both the quantitative as well as the qualitative side from several viewpoints. From the C-level viewpoint, an aggregation mechanism for a single, synthetic number for taking decisions could desirable. Therefore, the next step will be the implementation of the **QEST nD** model [21] with two possible adaptations using a balanced set of measures with:

- three (3) dimensions: A (FUR-based/product), B (NFR-based/product), C (PRJ/project-organizational)
- four (4) dimensions: the ITIL4 Four Dimensions (Organizations & People; Information & Technology; Partners & Suppliers; Value Streams & Processes).

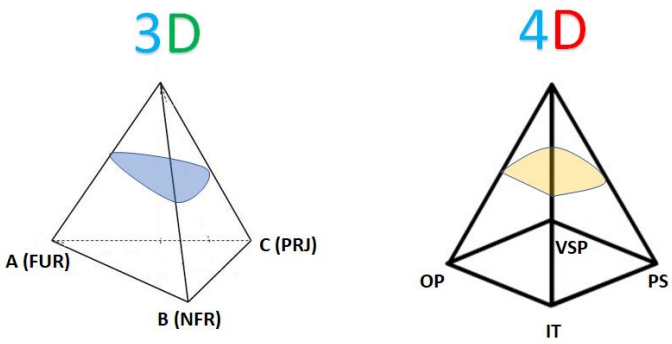

Figure 5: QEST nD models with two possible dimensions: 3D (left) or 4D (right)

"*Everything should be made as simple as possible, but not simpler*" (Albert Einstein)

References

[1] TAXUD, Tender Specifications - PART C – TECHNICAL ANNEX, Invitation to tender TAXUD/2020/OP/0001, Specification, development, maintenance and support of customs and taxation IT systems (SOFT-DEV), TAXUD/2020/OP/0001 – SOFT-DEV, October 2020, URL: https://shorturl.at/wJYZ8

[2] Jones, C., What Are Function Points? SPR website, URL: https://shorturl.at/ejSV8

[3] Buglione L., Abran A., ICEBERG: a different look at Software Project Management, Proceedings of IWSM-MENSURA 2002, Magdeburg (Germany), Oct 7-9, 2002, URL: https://tinyurl.com/yt8me7e8

[4] Buglione L., The Next Frontier: Measuring and Evaluating the Non-Functional Productivity, MetricViews, August 2012, URL: https://tinyurl.com/5feyurrv

[5] Ben Cnaan., Symons C., Glossary for Non-Functional Requirements, IFPUG/COSMIC white paper, September 2015, URL: https://tinyurl.com/2rj4cy3c

[6] Buglione L., Dekkers C., Advancements in Software Development Productivity: The FP-based 'Productivity Paradox', MetricViews, Dec 2021, pp. 23-31, URL: https://tinyurl.com/mj7ara68

[7] Czarnacka-Chrobot B., What Is the Cost of One IFPUG Method Function Point? Case Study, 11th Int. Conference on Software Engineering Research and Practice (SERP'12), 2012 World Congress in Computer Science, Computer Engineering & Applied Computing (WORLDCOMP'12), Las Vegas, Nevada, USA, CSREA Press, URL: https://shorturl.at/anxAF

[8] Van Heeringen H., Analysis of Project Costs per Function Point, ISBSG Short Paper, 2023, URL: https://shorturl.at/nuCIV

[9] Jones, C., Software Economics and Function Point Metrics: Thirty years of IFPUG Progress, White Paper, April 2017, URL: https://tinyurl.com/mr3yuuve

[10] Grist Project Management, Fixed price per unit delivered contracts, URL: https://shorturl.at/xMQ17, Accessed on: July 28, 2023

[11] ---, The Inner Costs of Outsourcing, CIO Journal, September 2020, URL: https://tinyurl.com/36h7d5n6

[12] Buglione L., The '123 Schema': why Nominal Productivity & PDR are changing with different kind of projects, ISBSG Webinar, Dec 2021, URL: https://tinyurl.com/mv7wbj8t

[13] ISBSG, Development & Enhancement (D&E) repository, 2022 edition, URL: https://www.isbsg.org/data-subscription-2/

[14] ISO/IEC, TR 14143-5:2004 (R2019), Information technology — Software measurement — Functional size measurement — Part 5: Determination of functional domains for use with functional size measurement
[15] Buglione L., Ebert C., Estimation, Encyclopedia of Software Engineering, Taylor & Francis Publisher, June 2012, ISBN: 978-1-4200-5977-9
[16] Salary.com, COBOL Programmer Salary, URL: https://shorturl.at/luBI1, Accessed on: July 28, 2023
[17] IT JobsWatch, Microsoft PowerBI Developer UK, URL: https://shorturl.at/txNX3, Accessed on: July 28, 2023
[18] Demarco T., Controlling Software Projects: Management, Measurement & Estimation, Yourdon Press, 1982, ISBN 9780917072321
[19] ISO/IEC, IS 15939:2017 (R2022), Systems and software engineering — Measurement process
[20] Axelos/Peoplecert, ITIL Foundation: ITIL4 Edition, 2022, ISBN 978-9925600007
[21] Buglione L., Abran A., QEST nD: n-dimensional extension and generalisation of a software performance measurement model, Advances in Engineering Software, Vol.33, Issue 1, Jan 2002, pp.1-7, https://doi.org/10.1016/S0965-9978(01)00050-3

Appendix A – Acronyms & Main Terms

Acronym	Full Term
123 Schema	Three sub-projects/lifecycle phases within a whole service project scope: (1) Development; (2) Operation; (3) Maintenance – as defined in [4]
ABC	(The first three letters of the alphabet) – meaning: something fundamental and basic
ABC Schema	Requirement taxonomy introduced in [2], also referenced in [7]
ABC Model	4-steps model for improving the management of a contract (ICT and not)
CHAR	Characteristics of FUR relevant to FSM (technique from ISO/IEC 14143-5:2004)
COSMIC	Common Software Measurement International Consortium (www.cosmic-sizing.org)
D&E	Development & Enhancement (see ISBSG repositories: www.isbsg.org)
FP	Function Point
FPA	Function Point Analysis
FUR	Functional User Requirement
FSM	Functional Size Measurement
GBP	Great Britain Pound
EAM	Entity-Attribute-Measure
EUR	Euro (€)
ICT	Information & Communication Technology
IFPUG	International Function Point Users Group (www.ifpug.org)
IEC	International Electrotechnical Commission (www.iec.ch)
ISO	International Organization for Standardization (www.iso.org)
LOC	Line(s) of Code
m-d	Man-day
m-hrs	Man-hour
MIS	Management Information System
NFR	Non-Functional Requirement ('B' requirement-type in the 'ABC Schema')
PL	Programming Language
PRJ	Project ('C' requirement-type in the 'ABC Schema')
Q→T→C	Quantity → Time → Cost
SLC	Software Life Cycle
UoM	Unit of Measure
UR	User Requirement
USD	United States Dollar