Luigi Buglione

Project Size Unit (PSU)

Measurement Manual

Version 1.21
(PSU-MM-1.21e)

November 2007

How to reference this document:
Luigi Buglione, *Project Size Unit (PSU) – Measurement Manual, version 1.21,* PSU-MM-1.21e, November 2007

For more information about PSU and other Software Measurement & Quality issues, please visit:
< http://www.geocities.com/lbu_measure > or contact L.Buglione by email at luigi.Buglione@computer.org

First Printing: November 2007

Table of Contents

1 Document Information

1.1 Executive Summary

This document describes **PSU** (**P**roject **S**ize **U**nit), a project management technique that allows to associate a measure of size to the project effort estimated by experience/analogy. It can be used yet from the Bid phase, because its main inputs are the initial Customer requirements and the first planning and related WBS by the Project Manager, to be refined during next SLC phases. Thus, it can be identified as an "early sizing" technique.

1.2 History

Revision	Date	Changes since last revision
1.00	31/08/2005	• First issue
1.01	05/10/2005	• Fixed typo errors, improve readability (whole document) • Clarified possible ambiguities between RHLR and Tasks usage (Sct.3, 4) • Improved the calculation example (Section 4)
1.2	27/08/2007	• Fixed some typo errors • Usage of PSU within Agile Projects (Section 2.4) • Automating PSU (Section 3.4) • Different PSU_{QM} calculation for new and closed projects (Section 4) • Size comparison between PSU v1.01 and v1.2 (Section 4.9) • Using PSU with Agile projects (Section 4.10) • Setting up PSU in Your Organization (Section 5)
1.21	01/11/2007	• Fixed some typo errors

1.3 Acronyms

Acronym	Description
ANOVA	Analysis of Variance
CFPS	Certified Function Point Specialist
CMM / CMMI	Capability Maturity Model / CMM Integration (www.sei.cmu.edu/cmmi/)
COCOMO	Cost Construction Model (http://sunset.usc.edu/research/COCOMOII)
COSMIC	Common Software International Consortium (www.cosmicon.com)
E/F	Early/Fast
EI/EO/EQ	External Input / External Output / External inQuiry
EIF	External Interface File
F/Q/T	Functional / Quality / Technical (referred to the nature of a requirement)
FFP	Full Function Points
FP	Function Points
FPA	Function Point Analysis
FSM	Functional Size Measurement
FSMM	FSM Method
FUR	Functional User Requirement
GSC	General System Characteristic
GUFPI-ISMA	Gruppo Utenti Function Point Italia – Italian Software Metrics Association (www.gufpi-isma.org)
H/M/L	High/Medium/Low (referred to tasks complexity)
HF	Homogeneity Factor
HLR	High-Level Requirement
ICT	Information & Communication Technology
IFPUG	International Function Point User Group (www.ifpug.org)
ILF	Internal Logical File
ISBSG	International Standard Benchmarking Software Group (www.isbsg.org)

KPA	Key Process Area
LOC	Lines of Code
M/Q/T	Management / Quality /Technical (referred to the nature of a task)
MMRE	Magnitude of MRE
MRE	Mean Relative Error
NESMA	Netherlands Software Metrics Association
PHD	Project Historical Database
PMBOK	Project Management Body of Knowledge (www.pmi.org)
PSU	Project Size Unit (http://www.geocities.com/lbu_measure/psu/psu.htm)
PSU_{qm}	PSU for Quality-Management Tasks
PSU_t	PSU for Technical Tasks
RHLR	Refined HLR
SLC	Software Life Cycle
SPICE	Software Process Improvement Capability dEtermination (www.isospice.com)
UCP	Use Case Points
UR	User Requirement
VAF	Value Adjustment Factor

1.4 References

[ALBR79]	Albrecht A.J., *Measuring Application Development Productivity*, Proceedings of the IBM Applications Development Symposium, GUIDE/SHARE, October 14-17, 1979, Monterey, CA, pp. 83-92
[ALBR84]	Albrecht A.J., *AD/M Productivity Measurement and Estimate Validation*, IBM Corp., NY, 1984
[BOEH81]	Boehm B., *Software Engineering Economics*, Englewood Cliffs N.J., Prentice-Hall Inc., 1981, ISBN 0138221227
[BOEH00]	Boehm B.W., Horowitz E., Madachy R., Reifer D., Clark B.K., Steece B., Brown A.W., Chulani S & Abts C., *Software Cost Estimation with COCOMOII*, Prentice Hall, 2000, ISBN 0130266922
[BUGL03a]	Buglione L., *Misurare il Software. Quantità, qualità, standard e miglioramento di processo nell'ICT*, 2° edizione, FrancoAngeli, FA724.20, ISBN 88-464-4634-8, Maggio 2003, URL: http://www.geocities.com/lbu_measure/libri/mis.htm
[BUGL03b]	Buglione L., *Dimensionare il software: qual è il giusto metro?* White Paper, 11/10/2003, Bloom!, URL: http://www.bloom.it/buglione1.htm
[BUGL05]	Buglione L,, *PSU e Metriche Funzionali per il Dimensionamento del Software: Concorrenti o Alleati?*, Bloom.it, 11/02/2005, URL: http://www.bloom.it/buglione2.htm
[BUGL06]	Buglione L., *Calculation Feature in Project Management tools - Requirements*, version 1.0, PSU-AU-1.00e, December 2006; URL: http://www.geocities.com/lbu_measure/psu/psu-au-100e.pdf
[BUGL07a]	Buglione L., *Meglio Agili o Veloci? Alcune riflessioni sulle stime nei progetti XP*, XPM.it, February 2007, URL: www.xpm.it
[BUGL07b]	Buglione L. & Abran A., *Improving Estimations in Agile Projects: issues and avenues*, Proceedings of the 4th Software Measurement European Forum (SMEF 2007), Rome (Italy), May 9-11 2007, ISBN 9-788870-909425, pp.265-274, URL: http://www.dpo.it/smef2007/papers/day2/212.pdf
[BUGL07c]	Buglione L., *Some thoughts on Productivity in ICT Projects*, version 1.0, WP-2007-01, White Paper, July 1 2007; URL: http://www.geocities.com/lbu_measure/fpa/fsm-prod-100e.pdf
[COHN05]	Cohn M., *Agile Estimating and Planning,* Prentice Hall, 2005, ISBN 0131479415
[CONT86]	Conte S., Dunsmore H. & Shen V.Y., *Software Engineering Metrics and Models*, Benjamin Cummings: Manlo Park, CA, 1986, ASIN 0805321624
[IFPU03]	IFPUG, *Framework for Functional Sizing*, Version 1.0, September 2003*),* International Function Point User Group, Westerville, Ohio, January 2004, URL: http://www.ifpug.org
[IFPU04]	IFPUG, *Function Points Counting Practices Manual (release 4.2),* International Function Point User Group, Westerville, Ohio, January 2004, URL: http://www.ifpug.org
[ISBS07]	ISBSG, *ISBSG Repository R10 Field Description*, 2007, URL: http://www.isbsg.org
[ISO95]	ISO/IEC JTC1/SC7/WG7 N72, *International Standard 12207 - Information Technology : Software Life Cycle Processes*, 22/02/95, URL: http://www.iso.ch
[ISO01]	ISO/IEC 9126-1:2001, *Software Engineering-Product Quality-Part 1: Quality Model*: ISO and IEC, 2001
[ISO02a]	ISO/IEC 20968:2002, *Software Engineering-MK II Function Point Analysis- Counting Practices Manual*: ISO and IEC, 2002
[ISO02b]	ISO/IEC JTC1/SC7/WG10, IS 15504-5, *Software Engineering - Process Assessment - Part 5: An Exemplar Assessment Model,* 2002
[ISO03]	ISO/IEC 19761:2003, *Software Engineering-Cosmic FFP-A functional Size Measurement Method*: ISO and IEC, 2003
[ISO05]	ISO/IEC, IS 24570:2005 - Software engineering -- NESMA functional size measurement method version 2.1 -- Definitions and counting guidelines for the application of Function Point Analysis, International Organization for Standardization, 2005
[ISO07]	ISO/IEC14143-1:1998 (R2007), *Information Technology-Software Measurement-Functional Size Measurement-*

	Part 1: Definitions of Concepts: International Organization for Standardization, February 2007
[JONE97]	Jones C., *What are Function Points?*, Software Productivity Research Inc., 1997, URL: http://www.spr.com/products/function.htm
[MELI97]	Meli R., *Punti Funzione Anticipati: un nuovo metodo di stima per i progetti software*, Proceedings of the 8th ESCOM Conference, Berlin, May 26-28, 1997, URL: http://www.dpo.it
[PAUL93]	Paulk M.C., Weber C.V., Garcia S.M., CHRISSIS M.B. & Bush M., *Key Practices of the Capability Maturity Model Version 1.1*, Software Engineering Institute/Carnegie Mellon University, CMU/SEI-93-TR-25, February 1993, URL: http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf
[PMI04]	PMI, *A Guide to the Project Management Body of Knowledge* (PMBOK), 2004 Edition, Project Management Institute, 2004, URL: http://www.pmi.org
[SANT05]	Santillo L., Lombardi S., Natale D., *Advances in statistical analysis from the ISBSG benchmarking database*, Proceedings of SMEF2005 (2nd Software European Measurement Forum), Rome (Italy), 16-18 March 2005, URL: http://www.dpo.it/smef2005/filez/g105_gufpi.pdf
[SEI02]	CMMI Product Development Team, *CMMI for Systems Engineering / Software Engineering / Integrated Product and Process Development,* Version 1.1, CMMI-SE/SW/IPPD v1.1, Continuous Representation, CMU/SEI-2002-TR-003, Technical Report, Software Engineering Institute, December 2001, URL: http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr003.pdf
[SEI06]	CMMI Product Development Team, *CMMI for Development Version 1.2,* CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute, August 2006, URL: http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf
[SYMO98]	Symons C., *RAPS: Rapid Application Portfolio Sizing*, Software Measurement Services Ltd, 1998
[STSC01]	STSC, *CMMI-SE/SW v1.1 to Sw-CMM v1.1 Mapping*, USAF Software Technology Support Center, 2001. URL: http://www.stsc.hill.af.mil/consulting/cmmi/cmmiseswippdv11.pdf

2 Introduction[1]

One of the aspects of greater interest for a Project Manager is to have the possibility to determine the needed effort for developing a project as soon as possible and with the greater level of confidence. A certain amount of developed projects base the estimation on the experiential factor; the capability of the PM to foresee possible risks in the project in the most accurate, but exclusively qualitative manner, by analogy with similar experiences or implementation is absolutely a practice largely adopted.

By the way, also a well recognized guide as the Project Management Body of Knowledge (PMBOK) [PMI04] in the "core" processes, in particular the "*Activity Duration Estimation"* (6.4), identifies among the possible techniques for estimating the duration of a project firstly the "*expert judgement*", secondly the "*analogous estimating*" and only at the third place a quantitative criteria (*quantitatively based durations*), given by the multiplication of a whatever technical counting unit by the average productivity level. In conclusion, PMBOK proposes an effort *buffer* to be taken into account in order to face eventual project risks.

Measuring an entity, whatever it will be, should be anyway more and more guided by objective and not subjective evaluations. At least, it should be translated in an objective language what for its inherent nature is not, at the aim to properly manage it. "*You cannot control what you cannot measure*" cites the well-known sentence by Tom De Marco [BUGL03a]. Thus, it would be preferable to choose the third way among the ones proposed in the PMBOK.

During last 25 years the Software Engineering community has addressed a lot of effort and attention to the estimation issue. The diffusion and application of models based on the regression analysis such as COCOMO [BOEH81] [BOEH00] can properly represent its relevance, where the relationship between effort and size is as follows:

$$effort = f(size)$$

So the starting point for calculating the effort is the size of a project. Eventual variants for the *f* functions are out of the scope of this paper; please consider [CONT86] as a reference text for the estimation issues in a software development project[2].

2.1 Sizing a software project: which units?

Function Points [ALBR79][ALBR84] with its several variants and evolutions (referred as FSM – Functional Size Measurement methods[3]) represent surely the family of sizing techniques more feasible and with a growing diffusion in the Software Engineering world. [JONE97] efficiently summarise the *productivity paradox* issue comparing what it means counting a project using Lines of Code (LOC) or a functional unit[4].

The basic concepts for functional measurement can be easily summarised the count of the number of *functionalities* (included in the project *boundary*) from the User's viewpoint, expressed through a certain number of *technical entities*, each of these *weighted* according to its related complexity level, to which add an further portion for the *general complexity* for such specific software solution. Thus, in a generic way, it is possible to summarise what said as follows:

[1] [BUGL03b]

[2] In the following, according to [CONT86], it will be considered PRED(0.25) and with a MRE that should be not greater than 25%, but the Reader can consider each other percentage value - i.e. PRED(0.10) – in line with his/her own exigencies.

[3] For a discussion on FSM evolution and main methodologies (such as Mark-II [ISO02], NESMA [ISO05] and COSMIC-FFP [ISO03]), see [BUGL03a], Chapter 2.

[4] For a complete and exhaustive discussion on the *productivity paradox* and on Backfiring, see [JONE96].

$$size = \left[\sum_{i}^{n} \left(entity_i * complex\_lev \right) \right] * adjustment\_factor$$

2.2 Time for counting project size and information needed

In the FPA CPM (Counting Practice Manual) 4.2 [IFPU04], in Chapter 3 three categories of documents are identified – derived from the Feasibility study – with an ascending detailed level and therefore a greater counting precision:

- Initial User Requirements: this phase represents the User Requirements before the meeting held between the User and the Project Team. The characteristics associated to the usage of the documentation at this stage are to be: incomplete, not presenting some features not derived from the analysis, difficulties in the implementation, some issues extremely generic that cannot allow to derive the correct number of Function Points.
- Initial Technical Requirements: this stage represents the Developers' viewpoint on the Users' requirements created from the Feasibility study. Therefore, some technical issues for implementation are included even they could be not taken into account for the final count. The characteristics associated to the usage of this kind of documentation at this stage are to be: technology dependent, it is possible a not proper identification of Users' functional needs, too high emphasis on technical issues, *boundaries* defined from the technical architecture of other Organization's applications.
- Final Functional Requirements: this stage represent at least the result of the meeting between the User and the Project Team, allowing to make consistent and complete the definition of the *functional requirements*. Such final version is obtained therefore **before** beginning the development phase. As the Counting Practice Manual says, "*The function point count, assuming no additional changes of scope, should be consistent with the count at the completion of development*".

Therefore, the project sizing calculation with a FSM method such as FPA can be done only at the end of the "Analysis" phase in a Project Life Cycle, having to your disposal an "advanced" information detail about the implementation for the software to be released to the Customer.

Business exigencies require more and more to anticipate the moment for estimate the size, in order to define the needed effort and the related cost (and expected revenues) for the project. At the aim to validate such trend, several "early/fast" versions of FPA have been developed [MELI97] [SYMO98] or, more recently, of COSMIC-FFP. They allow – obviously to save time in counting the functional size, but with a lower level of confidence – the final number of functional size units and consequently, if used with a forecasting system, to calculate the estimated effort for developing a software solution, against a "full" usage of a certain sizing technique. A problem could arise when a company during the bid phase could not or had not always the possibility to spend the time needed to properly apply those early techniques. But it must be noted that in any case those techniques would measure only the *product* functional size of a software, not the size of a software *project* [BUGL07c].

Next figures summarises the moments and measures for typically sizing a project during the whole SLC, from the Bid phase on.

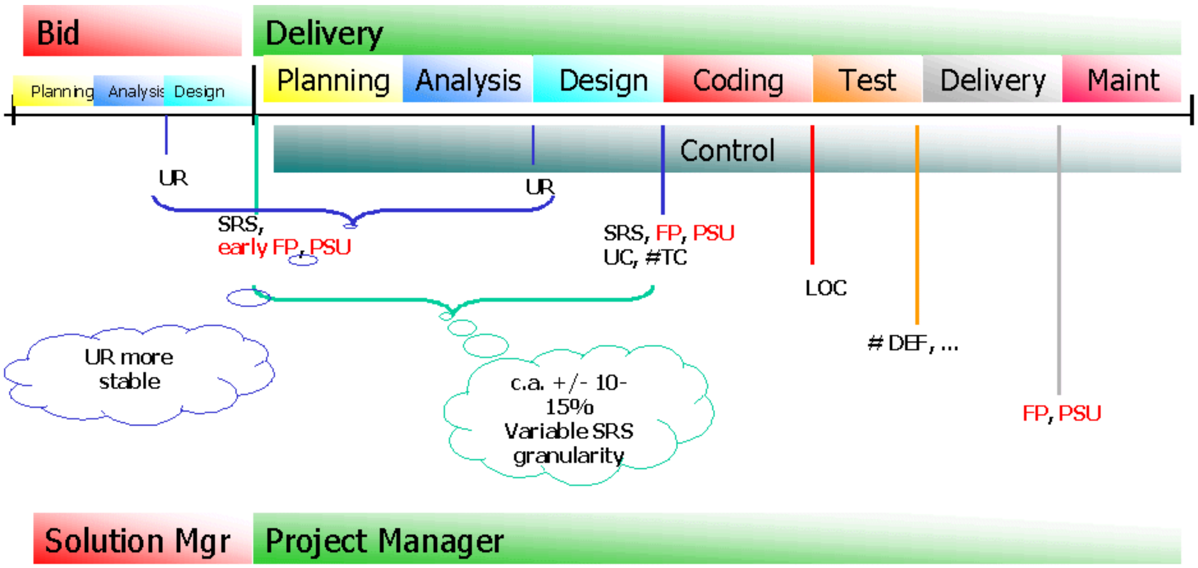


Fig. 1 – *Sizing Measures and possible gathering moments during the Software Life Cycle*

2.3 "Early" and "Standard" methods: friends or foes?

Such "early" versions, even with a lower level of detail, present anyway a counting of logical entities (inputs, outputs, enquiries, micro-functionalities, …). Using an historical project database properly populated with both the "full" technique and the "early & quick" one it is possible to calculate a *conversion factor* to apply on the new projects, using the E/F technique, allowing to obtain with a good approximation the number of "standard" sizing units[5]. In a general form, it is possible to summarise it as follows:

$$size\_full = size\_early * adj\_factor$$

Thus, through the analysis of MREs (Mean Relative Errors), PRED(0.25) and other typical estimation proxies by project and those related to the entire set of projects, it is possible to verify and evaluate the how much the estimations fit with the two systems, the "standard" and the early ones.

Another possible solution is to take into account only the early technique, evaluating the MRE and PRED(0.25) just referring to the estimated and final efforts, it means against itself, in order to derive the adjustment factor to apply for calculating the correct number of man/days.

2.4 Agile Projects and the Estimation Issue

Agile Methods (AM) such as XP, FDD and DSDM represent interesting solutions for projects with unstable requirements, iterative SLC, short-term milestones and small teams. Only in the last five years, the attention was also paid to Project Management practices in agile methodologies. Through this interest, planning and scheduling practices have been tailored to agile methodologies, but with much less attention to the estimation process. AM still being a young approach to software development, much work remains to be done to improve the way AM manage estimates, including tailoring relevant practices from well established and proven "heavyweight" methodologies.

[5] [ISO03] (ISO/IEC version of COSMIC-FFP) on Chapter 6 proposes how to compare results between COSMIC and IFPUG Function Points.

[BUGL07b] presented a detailed analysis of pros & cons in main AM estimation practices, stressing four main common attention points:

- No estimates for the non-functional requirements of projects
- No sizing units adopted
- No practices for gathering and using historical data
- No standards applied

In particular looking at the first bullet, it is difficult to apply a FSMM to an agile project, due to typical requirement instability and for the common usage to stress only the functional side of requirements (i.e. what in XP is called a User Story). Therefore, PSU could represent a sizing unit fitting also with this kind of projects, as illustrated in Section 4.9.

2.5 "Early" methods: which is the right Software Life Cycle phase?

A Software Life Cycle (SLC) standard such as ISO/IEC 12207:1995 [ISO95] shows a list of "*processes, activities and tasks*" to apply for developing or maintaining a software system, but explicitly "*does not specify details for implementation or execution of tasks included in processes*" (chapter 1.5).
Apart from the technical detail on the organization of a "life cycle" (waterfall, spiral, prototype, …) and related selection criteria, it must be taken into account a further, previous SLC phase, the *bid* one, which informative output – if the bid is won – is more consistent than the simple feasibility study. Main Software Process Improvement (SPI) models can help in determining such *hidden* information.

SPICE (ISO/IEC IS 15504-x)

The Project Management process is coded in ISO/IEC 15504 (aka SPICE) as MAN.2 (Project Management) and presents 12 Base Practices (BP), defined in a timely sequence for implementation. In particular, it must be noted that MAN.2.BP4 (*Size and Estimate tasks and resources*) precedes in time BP.10 (*Establish and implement Project Plans*). In the Part 5, Appendix A of the ISO model [ISO02] are furthermore listed inputs and outputs for each one of the defined processes. For the MAN.2 process several outcomes from the bid process (i.e. contract, agreement with the Customer, high-level functional specifications, information of the development environment, …). The Analysis phase for the "final technical requirements" in order to calculate Function Points in SPICE is identifiable within the *primary processes*, in ENG.1.3 (*Software design*), at the end of which it will be available also details of databases, needed for a proper DET and RET counting for the data component in FPA.

Sw-CMM v1.1 (1993) / CMMI v1.1 (2002) / CMMI v1.2 (2006)

The Planning process in the Sw-CMM v1.1 [PAUL93] is included in the Level 2 KPA named SPP (*Software Project Planning*). In particular, Activity 9 (Ac9) in the point#1 says that software sizing estimations must be done for all the main software work products and activities included in the project, referring also to some reference metrics such as LOCs and Function Points, while Activity 2 (Ac2) specifies that the planning for the software project (as a sub-part of the whole project) starts in the early phases and **in parallel** with the planning of the whole project. The Analysis activity is instead managed in the Level 3 KPA named SPE (*Software Product Engineering*). In particular, Software Design is described in Activity 3 (Ac3), with analogous considerations than those done before for SPICE processes.

Same considerations can be done with CMMI v1.1 [SEI02][6] as well as with its newer version 1.2 [SEI06], where the first specific practice when planning a project (PP, SP1.1) refers to the declaration of the scope of the project to be managed and therefore estimated, also (but not only) through its work products.

Therefore, if the needed information for producing the number of Function Points in a certain moment during the project life cycle would not be available, which size number should a Project Manager declare in order to estimate the project effort and consequently planning the activities and create the Gantt chart?

[6] For a mapping between Sw-CMM v1.1 and CMMI v1.1, please refer to [STSC01].

3 Project Size Units (PSU): Rationale

Previous question was purposely provocative, but derives from a real experience in large ICT companies. Some needed premises: not all the past projects, stored in an historical database, declared a *size unit*, basing – as introduced in the first lines of this paper – on experience and estimation by analogy, first two criteria listed in the PMBOK.

But if the application of a quantitative criterion (the third criterion in the PMBOK2004, Chapter 6.4) such as those of Function Points or other similar functional measures is possible just only the closing of the Analysis & Design phase, which is the right "meter" – respecting the same guiding principles – for sizing the software solution under exam and using such number in a forecasting system in order to derive in the Planning phase the number of needed man/days?

The answer, yet presented, has been to think and introduce an "early" estimation technique. This original technique has been named **PSU** (**P**roject **S**ize **U**nit), derived from the functional measurement logic expressed in the Function Point Analysis. If, as said, FPA measures the functional size for a software in a certain time t_x in the project lifecycle, PSU wants – at least as a willing – to keep back the same inspirer criteria, translating them to the time $t_{(x-1)}$ of the lifecycle, referring to the outcomes available in such moment. Retrieving the generic equation for the size calculation using a functional method:

$$size = \left[\sum_{i}^{n} \left(entity_i * complex\_lev \right) \right] * adjustment\_factor$$

<u>the following points have been faced</u>:

- <u>Entities</u>: the starting question was: which information is available when is asked to run the estimation, that is to say at the end of the winning of a Bid? The detail – not irrelevant – is to provide a consistent "answer" for all the projects, no matter to single particularities or *modus operandi* for the single Project Manager. The deliverables surely available are User Requirements, differently formulated by the Customer, as well as all the Technical Annexes produced by the Provider with the accepted technical solution accepted (it is supposed the bid has been won and the project planning is going to start), but there is not yet a sufficient detail in order to count the number of inputs, outputs, files, tables and so on. Therefore the simple User Requirements would represent an over-simplification. Their refinement – before writing the Functional Requirements and Design documents – can, under the agreement within the Project Team, produce a detailed list of requirements. Thus, each one of the refined user requirements to develop in the project will be "translated into practice" through the subsequent activities (**tasks**) written in the WBS, that represent the entity to be measured.
- <u>Complexity level for the considered entities</u>. After determined which is the entity to measure, next step is to give a weight for the complexity level to assign to the different instances for the considered entity. One of the major points to discuss for functional size measurement methods is given just by the weighting of the measured entities and the way to do it. Nowadays, the result is that also in the Function Points and similar methods the adjustment factor is considered as an optional, taking into account only the "*unadjusted*" value, because objectively derived referring to the development of technical activities, the first goal for Albrecht. In our case, since we have chosen to measure not a detailed element as an input or a file, but a higher level entity, the requirements from which such elements will be derived, it is not possible to leave out of consideration their weighting. How can a requirement be weighted, under these premises?

  Because we intend to calculate the project size, the final goal is to estimate the effort needed to produce it. Therefore, if a Project Manager would know to do not have to calculate necessarily the size (as in the first criterion in the PMBOK), next step will be simply to draft a Gantt chart, listing the detailed activities to perform and assigning a maximum duration to each single task. At the aim to create a certain uniformity in "writing" and estimating projects, it will be necessary to determine a time scale (statistically calculated) in order to parameter the tasks to the time needed for their execution, that's the effort:

$$\mathbf{1} \text{ User Req.} \rightarrow \boldsymbol{x} \text{ Detailed User Req's} \rightarrow \boldsymbol{y} \text{ tasks } (\rightarrow \mathbf{w} \text{ sub-tasks}) \rightarrow \mathbf{z} \text{ man/days}$$

Taking into account three levels of complexity (high, medium, low) from the analysis of historical project data and observing the granularity level for tasks inserted in the Gantt charts, we assigned a unique time correspondence for each detailed user requirement with respect to the number of task derived (linked to a standard effort expressed in man/days). Supposing that from a detailed UR two tasks to insert in the Gantt are derived, each one conventionally will not weight more than 5 man/days (total: 10 m/d), we will assign to it a "low" complexity an so on, until the definition of "high" complexity.

The brainstorming within the Project team about the number of tasks for detailing the detailed UR has for sure an influence on the total final number of *unadjusted* PSU calculated. Obviously an UR "translated" into a single "high" complexity task will weight less than an UR with a certain number of "low" complexity tasks. Therefore, the indication statistically derived on the maximum number of man/days for each complexity level has the ultimate scope to uniform the way the project must be expressed. In fact, comparing several projects Gantt charts, no matter to the final number of man/days, beyond the discussion with the Customer, the presence of a single "Analysis" tasks for a total of 40 m/d in place of a detail for each agreed UR, allow also to "read" *a posteriori* if the assignments done have been real, under or over estimated. Since we have not a "visual" comparability among similar projects, the estimation remains an activity totally in the hands and in the experience of the Project Manager, absolutely linked to a non-objective factor.

Surely at this point a question comes out: why do not directly calculate the effort without counting anything else? It would seem that, as the said says, that "*it is the snake that bites its tail*: to suppose an effort (the one for the average effort by task) in order to estimate another one (the total one for the project, by aggregation). The answer, even it would seem trivial, has its foundation in the first formula proposed: the effort is function of the size of a project that can be expressed basically by the number of "things to do". Refining the concept (at a level that could be called L-2), the "things to do" in FPA are data and transactions, classified in the five well-known entities (ILF, EIF, EI, EO, EQ). At a slight higher level (L-1 level), where it is not yet possible to count such detailed entities, it remains the "what to do". The weighting, obviously associated to the needed effort, can therefore happen only as a function of the "number of things to do (tasks)".

The two results from this phase will be therefore:

a) the **number of tasks associated to each detailed UR**, counted on the base of a table that establish a correspondence between the number of m/d to spend in the average for a task considered of high/medium/low complexity (that's therefore a direct application of the average productivity discussed before);

b) The **weight** – statistically derived from the periodical analysis of the historical project database – **associated to each one of the complexity levels** defined.

The multiplication of the number of entities (final technical tasks in the WBS) by the relative weights on the three complexity levels returns therefore the number of ***unadjusted* PSU**, that's the first value determined.

- <u>General adjustment factor.</u> The quantity above determined (unadjusted PSU) refers exclusively to the technical effort to provide for the project. This first number therefore excludes other effort for management and qualitative tasks. Such effort, with respect to the activities that SPICE or ISO/IEC 12207 standard call *primary processes*, will be proportional to the amount of technical activities. Also in this case it is possible to derive the average historical values for these two groups of activities with reference to the amount of effort for the technical activities, that's the number of *unadjusted* PSU. Thus, periodically a table that put in relation he number of unadjusted PSU and the effort for management-qualitative effort to add to the technical one must be updated. As listed in the fourth issue in the PMBOK (*contingency* or *buffer*), the adjustment factor intended, as a risk to consider with respect to the rough technical estimation is inherently included here and derived from the historical effort data for closed projects.

  Therefore, the result of this phase will be the calculation of PSU referring only to the Q/M tasks, but taking care if this component is reasonably aligned to the project historical data from past projects. In fact, a table will be periodically updated, returning the **proportions for the additional effort for management and qualitative tasks** against the total of unadjusted PSU.

Finally, the sum of unadjusted PSU (given by the *primary* processes) with the "adjustment factor" (given by the *organizational* and *support* processes) returns the final number of PSU for the project.

3.1 PSU and FPA: a first-level comparison

The following table compare basic elements for the determining the size in FPA and PSU.

Method \ Elements	Entity	Complexity	Adj. Factor	Complexity
FPA (standard)	Data (ILF, EIF) and Transactions (EI, EO, EQ) related to the functional requirement for a software system	3 levels (H/M/L) for each type of entity.	14 General System Characteristics (GSC)	Weight (0-5) for each one of the 14 GSCs, with a ± 35% variability on the value of unadjusted FPs
PSU (early)	Detailed (functional) UR and derived tasks (rule: 1 task = max *x* m/d) for the tasks of the SLC.	3 levels (H/M/L) for each detailed tasks per UR (and therefore of m/d, statistically derived).	% weighting for evaluating the amount of effort needed in proportion for management and qualitative tasks.	This percentage is derived from the analysis of the historical project database and it is proportional to the number of unadjusted PSU.

Tab. 1 – *FPA and PSU: comparison of basic calculation elements*

The following table proposes other viewpoints for a comparison between PSU (as an early sizing measure) and FPA (as a FSMM):

	Early methods (PSU)	Standard methods (FPA)
SLC phase to be applied	Planning (level L-1)	Design (level L-2)
Accuracy level	Lower than standard methods (avg)	Greater than "early" methods (avg)
The size unit refers to	Project	Functional User Requirements (FUR)

Control parameters for verifying the estimation accuracy	*In both cases, MRE and Pred(0.25) values calculated on the estimated effort must be compared with those calculated at the end of the project and the MMRE and Pred(0.25) on the entire set of projects included in the historical project database used for the forecasting system.*	
Information level needed	Documentation from the Bid phase	Documentation from the Analysis phase
Requested skills for estimation	Project team	Function Points Counter (preferably a CFPS)
Time needed for estimation	0.5 m/d (per PSU counting)	1.5-2 m/d (per FPA counting for medium-size projects [SANT05])
Strengths	• Quick calculation • Not requested FPA knowledge • Project estimation can be done before the Analysis & Design phase	• Greater accuracy in the size calculation to be used for the estimation • External comparability of results
Weaknesses	• Lower accuracy in the size calculation to be used for estimation, verification of correlation with "standard" techniques • Internal Comparability of results	• Greater effort for deriving the number of FPs • Requested the knowledge of FPA • Project Estimation can be done before starting the Developing phase (Coding)
Comments	Experimental & Internal technique.	Consolidated and diffused technique, with counting rules regularly monitored by International bodies.

Tab. 2 – *Early & Standard Sizing Methods: Characteristics, Pros & Cons*

3.2 PSU and FPA: which relationship?

A basic question is: what kind of effort does a Functional Size Measurement Method (FSMM) such as IFPUG FPA, COSMIC-FFP, NESMA, Mark-II measure? The effort related to the whole project or just a part of it? ISO/IEC 14143-1 standard [ISO07] and after also IFPUG [IFPU03] stated that User Requirements can be classified into three possible types, as showed in Fig.2:

- **F**unctional User Requirements (FUR): "*a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures that the software must perform to fulfil the users' needs. They exclude Quality Requirements and any Technical Requirements*"
- **Q**uality Requirements: "*any requirements relating to software quality as defined in ISO 9126:1991*"
- **T**echnical Requirements: "*requirements relating to the technology and environment, for the development, maintenance, support and execution of the software*".

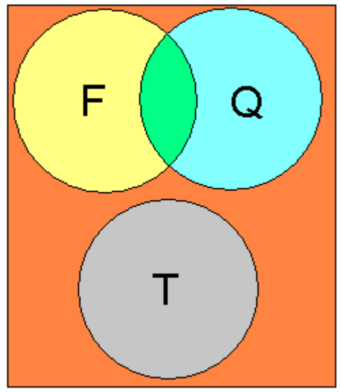


Fig. 2 –*Taxonomy for Requirements: Functional, Quality, and Technical*

The intersection between F and Q is because *functionality* is the first product quality characteristic listed in ISO/IEC 9126-1:2001 standard [ISO01]. Thus, it must be reaffirmed that a FSMM measures the size of solely the functional part of a software work product and not of the whole project whose goal is to produce such software.

The following question is: what does PSU measure? The answer here is: it depends, because it can have several possibilities, from the FUR – as a FSMM, allowing a direct comparability – till the whole projects, considering the whole set of User Requirements, no matter the requirement type (F/Q/T). In the first case, we should refer to **PSU_f** (f=functional), in the second case to **PSU_p** (p=project). The calculation rules are exactly the same and it will be sufficient in the Project Historical Database to consider them as different kind of units, because using a different amount of inputs.

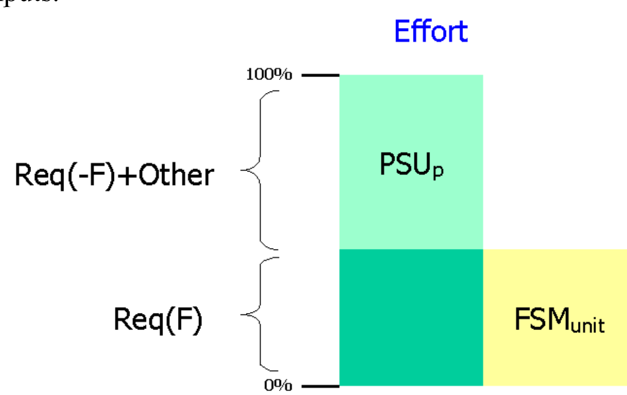


Fig. 3 – *PSU_p and FSMM: relationship*

PSU has been created thinking to an application for software projects, but since with PSU_p the entity to be measured is the "*project*", it is possible to apply it also to other kind of projects, for instance *service* projects. In this case, we could refer to **PSU_s** (s=service) and - as said before – storing them as different sizing unit in the Project Historical Database (**PHD**), clustering projects according to their nature.

Supposing to have measured a set of 5 projects (selected from a cluster of projects with similar characteristics) in the bid phase using PSU_p and after, from the analysis/design phase also with FPA, obtaining a situation such the one presented in Fig.4[7]:

[7] Those values are presented only for discussing the example. Please, do not consider them as conversion ratios.

Prj Set A							Requirements (% Effort)		
Id	FSMM	vers	#su	# PSU_p	# $PSU_{(f)}$	Eff tot	F	Q	T
p101	FPA	4.2	257	155	65	**1200**	42,00%	12,00%	46,00%
p102	FPA	4.2	420	152	85	**1350**	56,00%	9,00%	35,00%
p103	FPA	4.2	221	133	63	**950**	47,00%	15,00%	38,00%
p105	FPA	4.2	380	376	147	**2500**	39,00%	17,00%	44,00%
p107	FPA	4.2	153	112	55	**670**	49,00%	8,00%	43,00%
		Max	420,00	376,00	146,64	**2500,00**	56,00%	17,00%	46,00%
		Avg	286,20	185,60	82,85	**1334,00**	46,60%	12,20%	41,20%
		Med	257,00	152,00	65,10	**1200,00**	**47,00%**	12,00%	43,00%
		Min	153,00	112,00	54,88	**670,00**	39,00%	8,00%	35,00%

Fig. 4 – *PSU_p and FPA: Homogeneity Factor (HF)*

a first information needed is to know the “amount” of functional requirements against the whole set of requirements for such project. This information (HF – Homogeneity Factor) is derived calculating the median of the effort spent for type “F” requirements (functional), that’s the effort directly related to the functional size measured using a FSMM[8] (column “#su – sizing units”). Applying OF to the “PSU_p” column, we can obtain the “PSU_f” column, in order to allow a direct comparability between the two measures[9].

Id	FSMM	vers	#su	# PSU_p	#PSU_f	$Effort_f$
p101	FPA	4.2	257	155	65	**333,9**
p102	FPA	4.2	420	152	85	**599,2**
p103	FPA	4.2	221	133	63	**371,3**
p105	FPA	4.2	380	376	147	**677,0**
p107	FPA	4.2	153	112	55	**238,6**
new prj	FPA	4.2		416	196	**685,5**

Fig. 5 – *PSU_p , PSU_f and FP: an example sub-set of projects*

Using all the 5 projects and considering a linear regression, there would be a R^2=0.508, as shown in Fig.6:

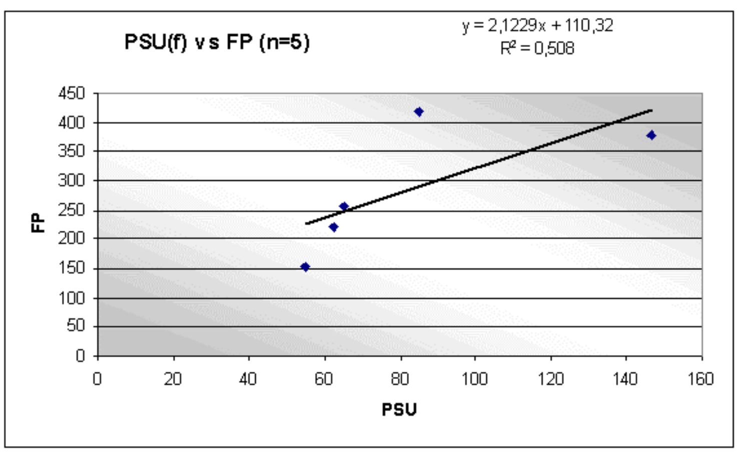


Fig. 6 – *PSU_f vs FP (n=5)*

Looking at the distribution of points and excluding project P105 (a possible outlier), the new linear relationship will be improved as in Fig.7, obtaining a R^2=0.9965.

[8] For instance, looking at the full detail from the PHD, the total effort for project P101 is split by F/Q/T type in this way: F=42%, Q=12%, T=46%. And so on, for all the projects stored in the PHD.

[9] When PSU_f will be taken into account, there is no need to calculate HF.

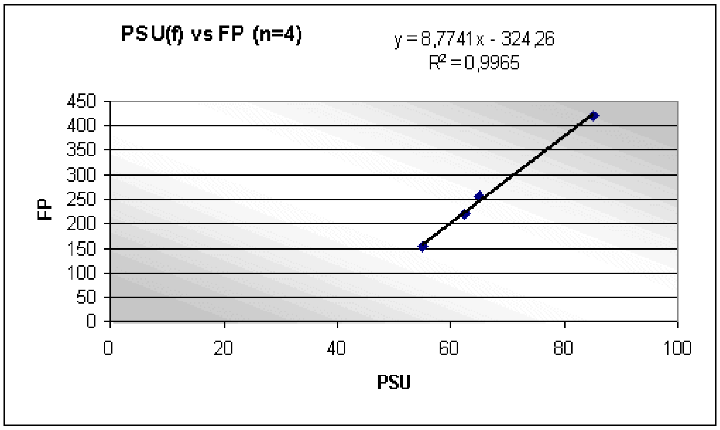


Fig. 7 – *PSU_f vs FP (n=4)*

Applying those coefficients, we would obtain FP*=1395 for the new project, that will be the estimated number of FP to use from the bid phase and to verify from the first real counting moment (design phase). Further adjustments can be introduced/considered by estimators while determining FP*, looking at the historical series of differences between FP* and the FP counted at the design phase.

3.3 PSU and PHD: Backfiring Past Projects

As yet introduced, a central element in the measurement process (i.e. ISO/IEC 15939:2007) is an historical database containing past project data, useful for many purposes, first of all for estimation of next projects. The main issue handling those kind of databases for estimation is the amount of instances contained in such databases: more the records, greater the affordability of forecasting from such database.

With PSU it is possible to calculate from past projects' documentation the number of PSU and to store it in the Project Historical Database (PHD), specifying that project has been measured in a *backfired* manner. The effect will be to have one gathering moment "used" out of the three suggested applying a sizing measure to a new project (bid, design, end of the project).

3.4 Automating PSU

Since PSU works with a WBS, it is possible to integrate the calculation rules discussed in Section 3 directly into a Project Management tool, in order to save time and calculate in a shorter time PSU, implementing also exporting features for enabling project data gathering for building/feeding the PHD.
A list of requirements for automating the technique is available in [BUGL06].

3.5 PSU on the Web

News and updates on PSU are available at: http://www.geocities.com/lbu_measure/psu/psu.htm. From this webpage you can download also templates and other free stuffs related to PSU. Comments and suggestions are welcome for improving the technique: you can send them at luigi.buglione@computer.org.

4 PSU: Calculation Procedure

After providing the rationale for PSU, now it's time to propose its calculation procedure, providing all the practical information.

4.1 Required Inputs

Documents and information to take into account shall be those deriving from Bid results or from internal projects requirements:

- Call for Bids, containing High-Level Requirements (from Customer)
- Assumptions for estimations performed during Bid phase (from Supplier)
- Technical Proposal (from Supplier)
- Initial Project Work Breakdown Structure (WBS, from Supplier)

4.2 Initial Assumptions

- **Consistency in applying counting rules**. As all techniques, also PSU requires – to provide homogeneous values – the consistent application in projects increasing the company Project Historical Database (PHD), following the rules detailed in the following sections. If these rules are not taken into account, this will cause non-homogeneous values invalidating effort estimation results, starting from dimensional data measured through similar but not equivalent rules. For instance, in FPA a maximum variability of 10% is accepted between two counts on the same project, in order to reduce the subjectivity in measuring. The same principle is valid for PSU that is borrowed from FPA approach and rules.
- **Proportionality between size and complexity**. As in any measurement method, the bigger is a given entity, the more complex to manage, and therefore the bigger is the effort to perform this management.
- **Additive Property**. PSU respect the additive property; this implies that different estimations done by several estimators on parts of a project can be put together for obtaining the final PSU_p value. This is a common case, where several Team Leaders have to provide to the Project Managers partial estimations for their own sub-system and for the Project Manager's consolidation into a unique, final effort and sizing values. An advantage in using PSU also for partial counts is that more and more people within the project team will know and share those sizing concepts and it will be easier also for those teams not skilled on FSMM to move towards those methods in the near future.

Some definitions used in the following sections are reported below, in order to clarify the meaning in the present document:

- *User Requirement* means Customer *desiderata*, not the UR documented in the Analysis document, as it is – at the moment of PSU measurement – temporally in the Planning phase yet. In the present document detailed user requirements are defined as **HLR** (High Level Requirements);
- *Detailed User Requirement* means that from a High-Level Requirement expressed by the Customer including several aspects related to the development of a specific function, all possible nuclear requirements shall be derived. In the following sections of this document, Detailed User Requirements are referred to as **RHLR** (Refined HLR). RHLR list represents a pre-analysis, to reuse after for writing Software Requirements Specifications (SRS);
- *Task* means the concrete activity deriving from detailed requirements formulation. For each detailed requirement, a number of activities to be performed shall be inserted in the Project

Gantt, which can be already pointed out in the Planning phase, as an input for the Gantt drafting.

The *complexity* of an activity (going back of a requirement) is parameterised to time required to perform the activity, according to ranges statistically determine from the analysis of Company historical data.

For this reason, to measure PSU the following series of relationships is valid:

1 User Req. → **x** *Detailed User Req.* → **y** *task (→ **w** sub-tasks[10]) → **z** man/days*

And therefore:

1 HLR → **x** *RHLR* → **y** *task (→ **w** sub-tasks) → **z** m/d*

4.3 Sizing calculation rule

In a general way, it is possible to express the project sizing based on functional measurement criteria in the following way:

$$size = \left[\sum_{i}^{n} \left(entity_i * complex\_lev \right) \right] * adjustment\_factor$$

Considering the task classification previously introduced:

$$PSU = PSU_T + PSU_{QM}$$

in detail:

$$PSU = \left[\sum_i \left(t - task_{Hi} * w_i \right) + \sum_j \left(t - task_{Mj} * w_j \right) + \sum_k \left(t - task_{Lk} * w_k \right) \right] + \left[\sum_i \left(qm - task_{Hi} * w_i \right) + \sum_j \left(qm - task_{Mj} * w_j \right) + \sum_k \left(qm - task_{Lk} * w_k \right) \right]$$

The first part of the formula (**PSU_T**) expresses the size of Technical (T) tasks, while the second one (**PSU_{QM}**) shows the weight brought out from Qualitative (Q) and Management (M) tasks, proportional to the first component[11]. Tasks complexity for the is measured according to the following table:

TASK COMPLEXITY	**# DERIVED SUB-TASKS**	**W_I**
High	>5	W_i
Medium	3-5	W_j
Low	1-2	W_k

Table 3 – *Table of weights for PSU measurement (unadjusted)*

The complexity for a task, expressed through the number of sub-tasks derived from a more accurate re-planning allows during time, verifying the tracked MRE (Mean Relative Error) value,

[10] Each task can be refined and split into a series of sub-tasks, detailing more the first-level activity (cfr. Section 4.4)
[11] See Section 5.5 for examples of M/Q/T task classification.

to align Project Managers' ability in the same granularity level for producing a project plan, as reported below. The weights shown in W column, in a descending complexity order, are statistically derived from the analysis should be performed at least twice a year using your own PHD, as well as the thresholds for the Quality Management System (QMS)/Business Process Model (BPM) measures.

About the second component, **PSU_{QM}**, the weight assigned to qualitative and management (p_M) activities, i.e. "Q" and "M" tasks are usually proportional to the T activities to develop and represents therefore in such sense, their *adjustment factor*.
In order to maintain proportionality among effort and size and the effort distributions decided by the project manager for the project, also the "Q" and "M" tasks must be weighted according to Table 3, returning a PSU_{QM} value.
But it will be compared with its median value extracted by the organizational PHD. In the case PSU_{QM} value is lower than its historical median for the desired cluster of projects, it must be decided if to add other "Q/M" tasks, since the "technical" tasks seemed to be overbalanced with respect to organizational and support tasks. On the base of the above information, the project manager will decide which will be the final list of tasks for the project WBS, determining the final PSU value.

The reference median value, derived from historical data as the percentage of the "Q" and "M" tasks on the total amount of the actual project effort and updated on a semi-annual basis, provides information about the percentage to assign according to the PSU_T range:

PSU_T	P_{QM} (%)
Range 1 (i.e.1-20)	$P_{QM\,1}$
Range 2 (i.e. 21-40)	$P_{QM\,2}$
Range 3 (i.e. 41-60)	P_{QM3}
Range 4 (i.e. more than 60)	$P_{QM\,4}$

Table 4 – *Table of weights for PSU_{QM} measurement (weighting factor)*

4.4 Activity counting – level of granularity

The *style* used for detailing project's tasks in a Gantt represents a critical factor for a proper project sizing. For each task inserted in the Gantt, the minimum time unit (*standard*) is **5** man/days, in order to allow a better control on project activities, as suggested by Project Management best practices. Considering the values in Table 5:

Task Complexity	# Derived sub-tasks	Thresholds in M/D	W_I
High	>5	>25 m/d	1.8
Medium	3-5	11-25 m/d	1.4
Low	1-2	5-10 m/d	1.0

Table 5 – *Table of weights for PSU_T measurement (example)*

If a task is considered to be performed in 15 days, distinguishing 3 sub-tasks the number of PSU_T to measure will be 1.4 (weight) for a task of medium complexity (going from 3 to 5 sub-tasks), for a total of 1.4*1=1.4 PSU for that task. And so on for all the others.

The granularity (and the number of PSU) is strictly related to the correct determination of tasks to be performed, i.e. the number of actual "functions" the project shall develop. Analogously, also with FPA if there is a low level of detail in EI, EO, EQ, ILF and EIF, the number of final FPs will be lower.

The increasing detail of tasks against a given first-level activity (e.g. a 15-day task split in 7 sub-tasks 2 m/d long) has a greater qualitative impact on the potential risk related to the activity completion in the estimated time, in function of the performed control. The more are control activities and their frequency on a given activity, the lower is the risk of late or problematic releasing the outcomes derived from activities, the lower the probability of higher MRE% or frequent re-planning. This detail can also represent a way to communicate to Customers, besides the Project Team, about the level of attention in planning "its" project.

A first criticism could be that different ways of creating a WBS on the same projects would drive to different results, with subsequent problems in estimating from those historical data. It could be right, but in the short-term. In fact, PSU it is **not only** a technique for associating a size to the estimated project effort, **but also** a way – through consistent application in a company – to harmonize and standardize the granularity level adopted from Project Managers in managing their projects. Next figure shows the trend for this phenomenon that in the mid-term will be "absorbed" if properly managed and followed from PSU adoption on.

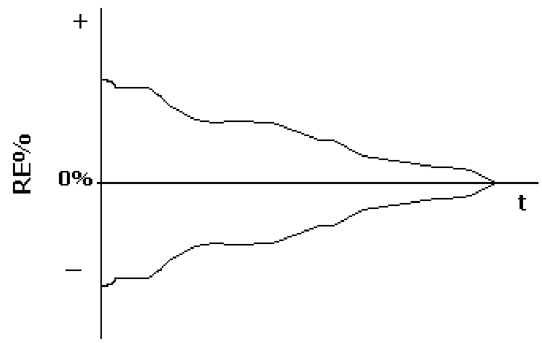


Fig. 8 – *RE% expected trend*

A suggested strategy for reaching this goal (minimizing the RE% - Relative Error) and make more homogeneous among Project Managers within a company the way to create WBS at the same level of granularity is the following:

- Write a first version of the project WBS
- Classify tasks by type (M/Q/T)
- Determine the nature of such task (Functional / Non-Functional)
- Associate each task with the related SLC phase
- Estimating effort to each tasks
- Sort tasks by descending work-effort [and - therefore - of task complexity (H/M/L)]:

At this moment, you'll have three groups of tasks and your objective will be to minimize (possibly to delete) the number of High complexity tasks (those with a related effort greater than 25m/d) and remain with a final list containing only Low and Medium complexity tasks (it would mean that your style in creating WBS is quite granular, reducing the possibility to have high variability in your projects MRE%).

Next question is: how can I do it? Suppose to have a single task for "Project Management" for 40m/d along a 320m/d project. Probably there will be several sub-activities really performed or milestones considered within that 40m/d bar drawn into the Gantt chart. And those *real* micro-activities should be considered. Again, for a generic "Test Management" activity planned for 35 m/d, possible criteria for splitting it into more sub-tasks (and therefore reducing tasks complexity) could be a split by testing levels, by sub-systems tested, etc.

Here in the following there are some possible suggestions, according the kind of task considered.

Initial task	Criterion for splitting initial tasks
Project Management	• By "Planning" and "Monitoring & Control"
Test Management	• By testing level (unit, integration, system, …) • By sub-system • By testing groups • Pre-Post delivery • …
Coding	• By module • By programming languages (if more than one) • By working groups (if more than one) • …
…	…

4.5 Weighting system

As shown in the above formulas, task complexity and the effort contribution provided by Q/M-type tasks are expressed through a system of weights. This allows to obtain a final result in terms of number of sizing units closer and closer to the real project complexity, assuming a greater relevance when referred to data comparability across time.

Every time an historical analysis of data is referenced for statistical determinations, this is to be intended as starting from the six months following the data collection start-up. As a matter of fact, in the first period, the values for these parameters are necessarily estimated on experience basis, as previous historical series are not available, as indicated in PMBOK2004 (process 6.4 - Activity Duration Estimation) [PMI04]. Weights revision must be periodically performed, as well as a possible modification of the number of complexity levels currently determined (both for PSU_T and PSU_{QM}).

This constant updating of the weighting system for a PSU implementation implies that PSU can be applied only for internal benchmarking using that unique system of weights that a certain company will derive from its historical project size-effort data. It must be stressed that it is a characteristic (and not necessarily a limit) of the technique.

Note: the weights and effort ranges presented in the following Sections of Chapter 4 are just example figures allowing to show some calculation examples. See Chapter 5 about how to set up and adjust such values for your Organization, starting from your own project data.

4.6 Sizing Procedure for PSU Calculation

After providing all assumptions, motivations and elements to measure Project PSU, this section summarizes all the steps to perform to determine the number of PSU. All steps, apart from step 2 (brainstorming meeting) will be performed by the Estimator (typically the Project Manager).

S	SS	Description	Input	Output
1		**Collection of available project documents/information**	**Initial Requests from Customer**	**All needed documents collected**
2		**Brainstorming session with proper project team resources, including *affected groups* possibly interested;** in detail:	**All needed documents collected**	**Quantitative assumptions formulated**
	a	Issuing high level requirement list provided by the Customer or derived from documents mentioned in step 1	*Documents from the Bid Phase*	*List of HLRs*
	b	Issuing detailed requirement list for each high level requirement mentioned in step 2a	*List of HLRs*	*List of RHLRs*
	c	Determination of the number of required activities for each detailed requirement mentioned in step 2b.	*List of RHLRs*	*# tasks per each RHLR*
	d	Assigning estimated effort for each task identified in step 2c in man/days and classify by nature (M/Q/T), type (F/NF) and SLC phase	*# activities for each RHLR*	*Effort (m/d) per each detected activity, classified by nature/,type and SLC phase*
	e	Formalization of assumption in the "*Assumptions*" sheet	• *List of HLRs* • *List of RHLRs* • *No. of tasks* • *Effort per each task*	*"Assumptions" sheet filled*
	f	Closing up meeting	*All previous outputs*	Quantitative assumptions formulated
3		**PSU calculation**	**Quantitative assumptions formulated**	**# PSU**
	a	Deriving tasks complexity	• *List of RHLRs* • *No. of tasks* • *Effort per each task*	*Complexity level per each task*
	b	Minimization (when possible) of the H-level complexity tasks (refinement; §4.4)	*Complexity level per each task*	*Complexity level per each task (refined)*
	c	Calculation of PSU_T for each complexity level	*Complexity level per each task (refined)*	*# PSU_T per complexity level*
	d	Calculation of PSU_{QM} for each complexity level	*Complexity level per each task*	*# **PSU_{QM}** per complexity level*
	e	Check PSU_{QM} and % Effort (QM) against historical median percentage to apply from PHD for the cluster of projects closer to the estimated one. If a WBS revision is needed, come back to step 2c.	*# PSU_{QM} ; % P_{QM} to apply; % Effort(QM)*	*final # PSU_{QM} ; Final effort(Q/M) tasks*
	f	Summing up PSU_T and PSU_{QM}	# PSU_T ; # PSU_{QM} ;	Tot.# PSU

4.7 A Sizing example

A sizing example is now presented, with a detailed comment step by step, according to the sizing procedure.

1		**Collection of available project documents/information**

Documentation and information from the bid phase, after winning it, are gathered and used.

2		**Brainstorming session with proper project team resources, including *affected groups* possibly interested;** in detail:
	a	Issuing high level requirement list provided by the Customer or derived from documents mentioned in step 1

From available documents 4 HLR are derived.

	b	Issuing detailed requirement list for each high level requirement mentioned in step 2a

During brainstorming, the 4 HRL are analysed (a fifth one is derived from the analysis, for Planning & Control) and refined into 12 RHLR as follows:

HLR	RHLR
HLR#01	RHRL#01
	RHRL#02
HLR#02	RHRL#03
	RHRL#04
	RHRL#05
HLR#03	RHRL#06
	RHRL#07
	RHRL#08
HLR#04	RHRL#09
	RHRL#10
HLR#05	RHRL#11
	RHRL#12

	c	Determination of the number of required activities for each detailed requirement mentioned in step 2b.

The detail of the number of detailed activities required/proper for each RHLR is added to the previous table:

HLR	RHLR	Tasks
HLR#01	RHRL#01	A#01
		A#02
	RHRL#02	A#03
		A#04
		A#05
HLR#02	RHLR#03	A#06
	RHLR#04	A#07
		A#08
	RHLR#05	A#09
		A#10

HLR#03	RHLR#06	A#11
	RHLR#07	A#12
		A#13
		A#14
	RHLR#08	A#15
		A#16
		A#17
		A#18
HLR#04	RHLR#09	A#19
	RHLR#10	A#20
		A#21
		A#22
		A#23
		A#24
		A#25
HLR#05	RHLR#11	A#26
		A#27
		A#28
		A#29
		A#30
		A#31
	RHLR#12	A#32
		A#33
		A#34
		A#35
		A#36
		A#37

	d	Assigning estimated effort for each task identified in step 2c in man/days and classify by nature (M/Q/T), type (F/NF) and SLC phase

To each activity (A) to insert in detailed Gantt, a value in m/d is estimated, according to the M/Q/T tasks classification, its type (functional/non-functional) and the related SLC phase (according the SLC phases used in the QMS processes).

#HLR	RHLR	Tasks	M/d	Type	F/NF	SLC
HLR#01	RHRL#01	A#01	5	M	NF	Planning
		A#02	2	M	NF	Planning
	RHRL#02	A#03	7	M	NF	Control
		A#04	5	Q	NF	Control
		A#05	5	Q	NF	Control
HLR#02	RHLR#03	A#06	11.5	T	F	An/Design
	RHLR#04	A#07	30	T	F	Construction
		A#08	15	T	F	Construction
	RHLR#05	A#09	21.5	T	F	Test
		A#10	15	T	NF	Delivery
HLR#03	RHLR#06	A#11	2	T	NF	An/Design
	RHLR#07	A#12	13	T	F	Construction
		A#13	12	T	F	Construction
		A#14	10	T	F	Construction
	RHLR#08	A#15	7	T	F	Test
		A#16	24	T	NF	Delivery
		A#17	17.5	T	NF	Delivery
		A#18	7.5	T	NF	Delivery

HLR#04	RHLR#09	A#19	20	T	F	Change Req
	RHLR#10	A#20	8	T	F	Construction
		A#21	2	T	F	Construction
		A#22	2	T	F	Construction
		A#23	5	T	NF	Test
		A#24	4	T	NF	Delivery
		A#25	4	T	NF	Delivery
HLR#05	RHLR#11	A#26	15	T	NF	Problem Rep
		A#27	6	T	NF	Problem Rep
		A#28	2	T	NF	Problem Rep
		A#29	2	T	F	Problem Rep
		A#30	2	T	NF	Problem Rep
		A#31	1.5	T	NF	Problem Rep
	RHLR#12	A#32	5.5	T	NF	Construction
		A#33	3	T	F	Construction
		A#34	4	T	NF	Test
		A#35	11	T	NF	Delivery
		A#36	2.5	T	NF	Delivery
		A#37	2	T	NF	Delivery
			311.50			

A total effort of **311.50**m/d have been estimated for **32** technical (t-tasks) activities and **5** quality & management (qm-tasks) out of 37 total ones, according to the M/Q/T tasks classification.

	e	Formalization of assumption in an "*Assumptions*" sheet

All information provided as example, shall be inserted in a sheet among estimation assumptions considered for the project.

	f	Closing up meeting

The meeting will eventually be closed up, formalizing all decisions and all contributions from affected groups, relating to requirements that involve them for the subject.

3		**PSU calculation**
	a	Deriving tasks complexity

Here the Project Manager, having a pre-analysis for the current project, shall associate complexity to each task according to the time thresholds and weights specified in Section 4.4. From the previous table, derives the following:

#HLR	RHLR	Tasks	M/d	Type	F/NF	SLC	Compl
HLR#01	RHRL#01	A#01	5	M	NF	Planning	L
		A#02	2	M	NF	Planning	L
	RHRL#02	A#03	7	M	NF	Control	L
		A#04	5	Q	NF	Control	L
		A#05	5	Q	NF	Control	L
HLR#02	RHLR#03	A#06	11.5	T	F	An/Design	M
	RHLR#04	A#07	30	T	F	Construction	H
		A#08	15	T	F	Construction	M
	RHLR#05	A#09	21.5	T	F	Test	M
		A#10	15	T	NF	Delivery	M

HLR#03	RHLR#06	A#11	2	T	NF	An/Design	L
	RHLR#07	A#12	13	T	F	Construction	M
		A#13	12	T	F	Construction	M
		A#14	10	T	F	Construction	L
	RHLR#08	A#15	7	T	F	Test	L
		A#16	24	T	NF	Delivery	M
		A#17	17.5	T	NF	Delivery	M
		A#18	7.5	T	NF	Delivery	L
HLR#04	RHLR#09	A#19	20	T	F	Change Req	M
	RHLR#10	A#20	8	T	F	Construction	L
		A#21	2	T	F	Construction	L
		A#22	2	T	F	Construction	L
		A#23	5	T	NF	Test	L
		A#24	4	T	NF	Delivery	L
		A#25	4	T	NF	Delivery	L
HLR#05	RHLR#11	A#26	15	T	NF	Problem Rep	M
		A#27	6	T	NF	Problem Rep	L
		A#28	2	T	NF	Problem Rep	L
		A#29	2	T	F	Problem Rep	L
		A#30	2	T	NF	Problem Rep	L
		A#31	1.5	T	NF	Problem Rep	L
	RHLR#12	A#32	5.5	T	NF	Construction	L
		A#33	3	T	F	Construction	L
		A#34	4	T	NF	Test	L
		A#35	11	T	NF	Delivery	M
		A#36	2.5	T	NF	Delivery	L
		A#37	2	T	NF	Delivery	L
			311.50				

We will obtain therefore the following complexity distribution:

Task Complexity	**# t-tasks**	**# q-tasks**	**# m-tasks**
High	1	0	0
Medium	12	0	0
Low	19	2	3
	32	**2**	**3**

	b	Minimization of high-level complexity tasks

Supposing the previous WBS classification, there is only a high-level complexity task (A#07). In this case the Project Manager decides to maintain the whole task without decomposing it.

	c	Calculation of PSU_T for each complexity level

Supposing the following weights:

Task Complexity	**W_I**
High	1.8
Medium	1.4
Low	1.0

The following results are obtained for t-tasks:

Task Complexity	**# Task**	**W_I**	**PSU_T**

High	1	1.8	1.8
Medium	11	1.4	15.4
Low	20	1.0	20.0
			37.2

	d	Calculation of PSU_{QM} for each complexity level

Supposing the same weights, the following results are obtained for qm-tasks:

Task Complexity	# qm-Tasks	W_I	PSU_{QM}
High	0	1.8	0.0
Medium	0	1.4	0.0
Low	5	1.0	5.0

	e	Check PSU_{QM} and % Effort (QM) against historical median percentage to apply from PHD for the cluster of projects closer to the estimated one. If a WBS revision is needed, come back to step 2c.

Supposing to derive the following PQM median values of proportionality between the T/QM size from the internal PHD:

PSU_T	$P_{QM\ (\%)}$
1-20	10%
21-40	12%
41-60	15%
More than 60	18%

Disposing of **37.2** PSU_T, the percentage to apply should be equal to **12%**; looking at the two PSU values obtained:

	PSU - Abs	PSU - %
PSU_T	37.2	88.15
PSU_{QM}	5.0	11.85
	42.2	**100.00**

the size for the Q/M tasks is seems to be in line with the historical median values retrieved from the internal PHD (in this case, 5 PSUqm / 37.2 PSUt = 13.44%)

	f	Summing up PSU_T and PSU_{QM}

The last operation is to determine the final PSU value for the Project, summing the two PSU quantities after evaluating if Q/M activities are adequate (in effort and size) for a proper management of the project.
The following results are obtained:

$$PSU = PSU_T + PSU_{QM} = 37.2 + 5.0 = 42.2$$

and rounding the value to the unit, **42 PSU**.

A series of **additional information** will be available for being stored into the PHD and used for estimation purposes. In particular:

- **Task classification by # of tasks / effort**

	No.Tasks		Effort (m/d)	
Task Classification	**Abs**	**%**	**Abs**	**%**
M- Management	3	8.11	14.00	4.49
Q – Quality	2	5.41	10.00	3.21
T- Technical	32	86.49	287.50	92.30
	37	**100.00**	**311.50**	**100.00**

- **Functional / Non-functional Requirement type**

	No.Tasks		Effort (m/d)	
Req.Type	**Abs**	**%**	**Abs**	**%**
F – Functional	14	37.84	157.0	50.40
NF- Non-functional	23	62.16	154.5	49.60
	37	**100.00**	**311.50**	**100.00**

- **Task classification by complexity**

	No.Tasks (all)		No. T-tasks	No. QM-tasks
Complexity	**Abs**	**%**	**Abs**	**Abs**
H- High	1	2.70	1	0
M – Medium	11	29.73	11	0
L - Low	25	67.57	20	5
	37	**100.00**	**32**	**5**

- **Task classification by SLC phase**

	No.Tasks		Effort	
SLC Phase	**Abs**	**%**	**Abs**	%
Planning	2	5.41	7.00	2.25
Control	3	8.11	17.00	5.46
Analysis & Design	2	5.41	13.50	4.33
Construction	10	27.03	100.50	32.26
Test	4	10.81	37.50	12.04
Delivery	9	24.32	87.50	28.09
Problem Reports	6	16.22	28.50	9.15
Change Requests	1	2.70	20.00	6.42
	37	**100.00**	**311.50**	**100.00**

- **Effort classification by task nature (M/Q/T) / SLC Phase**

	Effort			
	M	**Q**	**T**	
Abs	14.00	10.00	287.50	**311.50**
%	**4.49**	**3.21**	**92.30**	**100.00**

4.8 Tracking and re-calculation of PSU

As indicated in Fig.1, a sizing measure should be calculated in several moments during the whole Project Life Cycle:

- in the **Bid** Phase,
- at the end of the **Design** phase and finally,
- when the project has been **closed**.

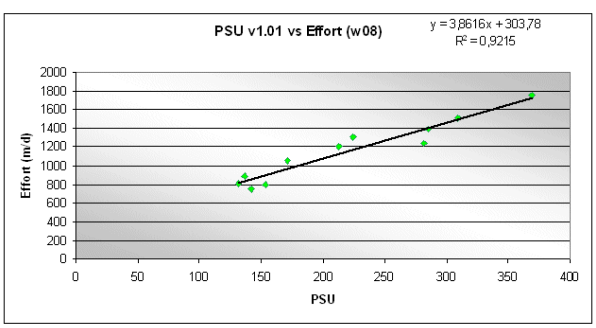
PSU can be used as other sizing measures for tracking the project and be part of derived measured (metrics) as – for instance – defect density, productivity [BUGL07c], etc.

4.9 PSU v1.01 vs PSU v1.2: comparing results

After presenting the modifications in the calculation rules, a question could arise: which is the impact on project size using PSU v1.01 and v1.2? Let's consider a sample of 11 projects, as in the following table where the tasks complexity has been derived applying the example ranges and weights:

project	PSU v1.01	PSU v1.2	Diff	Effort*	Effort	MRE	H	M	L
P001	282	342	60	1280	1236	-3.56%	3	7	327
P002	154	172	18	750	797	5.90%	5	7	153
P003	369	440	71	1680	1752	4.11%	4	8	422
P004	309	359	50	1648	1504	-9.57%	4	5	345
P005	142	163	21	900	743	-21.13%	4	5	149
P006	285	339	54	1459	1388	-5.12%	6	6	320
P007	171	212	41	980	1055	7.11%	4	8	194
P008	137	177	40	947	886	-6.88%	4	5	163
P009	224	276	52	1525	1308	-16.59%	6	7	255
P014	132	161	29	945	810	-16.67%	4	5	147
P015	213	266	53	1345	1200	-12.08%	5	6	249

As in the fourth column, the new rule about the QM tasks will bring an increase in the final PSU value. Plotting such data on a scattered diagram and comparing PSU vs Effort using both series, it is possible to note a light increase (+2%) using the new counting rules, since there is a better proportionality treating all tasks in the same manner, even if the QM ones should maintain a proportionality with T ones, on the base of a periodical re-assessment from PHD data.



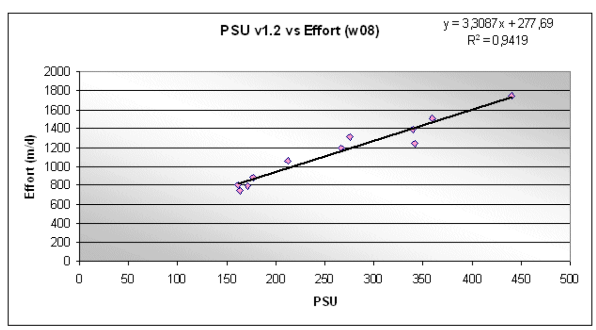


Fig. 9 – *PSU vs Effort (n=11): version 1.01 and version 1.2*

4.10Using PSU with Agile Projects

As previously introduced in Section 2.4, Agile projects usually adopt an iterative estimation process, but typically based on experience and analogy, without any standard quantitative size unit (refer to [BUGL07b] for a detailed analysis of the main adopted methods). This implies that every new project must be estimated again by experience-analogy and not by parametric models, since an historical data series expressing (at least) project size and effort would be never gathered, missing the size values.

PSU can represent a possible size unit to adopt, both at the *project* level as well as at the *requirement* level [BUGL07a]. In fact, the unit to estimate is the single requirement, typically observed from the functional viewpoint by the end user (the Customer). In XP each requirement provided by the Customer is called User Story (US). Figure 10 shows a well-known example about the automation of a coffee maker machine[12].

Title: *Waiting State*		
AccTest: *checkOptions0*	**Priority**: 1	**Story Points**: 2
When the Coffee Maker is not in use it waits for user input. There are six different options of user input: 1) add recipe, 2) delete a recipe, 3) edit a recipe, 4) add inventory, 5) check inventory, and 6) purchase beverage		

Fig. 10 – *User Story: an example (Coffee Maker)*

As shown, the solely functional side of this US represents the base for estimate the whole work effort (and costs) needed, approximated here in Story Points.

Considering the ISO/IEC 14143-1 requirement taxonomy previously introduced, it is possible to "power" the original US template structure into a new one, called US^2 (2^{nd}-generation User Stories), where each initial functional user requirement can be completed with its "non-functional" side, expressing its Q/T derived parts (whether they exist or have a technical sense to be managed in the project), typically introduced by the Provider during the requirement elicitation phase. The Provider has also to detail how each F/Q/T requirement piece will be translated into tasks, creating a first draft of a WBS, summing the outcome of this activity requirement by requirement. There will be an iterative feedback between Customer and Provider till an agreement point will be reached.

Another addition is to explicitly add to the US template the measurement unit and the US estimated effort, helping also for an external evaluation and comparability outside the original team working on the project.

Title: *Waiting State*			
AccTest: *checkOptions0*	**Priority**: 1	**(MeasUnit)**:	**Effort (m/d)**:
F	When the Coffee Maker is not in use it waits for user input. There are six different options of user input: 1) add recipe, 2) delete a recipe, 3) edit a recipe, 4) add inventory, 5) check inventory, and 6) purchase beverage		
Q			
T			

Fig. 11 – *US^2: main changes from traditional US (in blue)*

[12] URL: http://open.ncsu.edu/se/tutorials/coffee_maker/

It is not an apparently huge change, but the written formalization of requirements, one by one, including also what the Customer could think was *hidden* (that's often the Q/T part) and drafting also the tasks is a very good advancement for reducing possible future discussions about the congruity of estimates.

Therefore the steps to run would be:

1. The Customer proposes a US^2 (F-side) and pass it to the Provider
2. The Provided complements the US^2 deriving the Q/T side (when possible and have a sense in the context of the project) and send it back to the Customer
3. the Customer evaluate the proposal; if ok[13], go on with step 4, otherwise go back to step 2 with comments/suggestions
4. The Provider translates the F/Q/T parts into tasks with an effort estimation in m/d, discussing them with the Customer, till they arrive to an agreement point. At such moment, the draft WBS – given by the sum of the partial WBS, US^2 by US^2 – has been produced, where there must be added some additional tasks per each iteration (i.e. **Sprint**, in the Scrum language) referred to the organizational & support tasks (what in PSU are the Q/M tasks type), whether not yet considered in the single US^2.

So, what about PSU in this discussion? It would be possible simply to adopt this refinement without introducing a size measure, but it would not be possible again to know which is the size for a certain requirement and – for addition – of the agile project itself.

Otherwise, the additional steps to run would be:

1. Calculate PSU per each US^2 with the above illustrated formula and insert size and effort values in the US^2 template
2. Assign all the US^2 by project iteration
3. Add the Q/M tasks needed per each iteration and calculate the additional PSU_{QM}
4. Sum the size of all the US^2 assigned to an iteration and the additional PSU_{QM} in order to obtain the *iteration* size (and effort)
5. Sum the overall size of each iteration in order to obtain the *project* size (and effort).

[13] Possible evaluation criteria could be the ones proposed by Mike Cohn, summarised in the **INVEST** acronym [COHN05].

5 Setting up PSU in your Organization

After presenting the assumptions and the *mechanics* of PSU calculation in Section 4, now it's time to discuss how to properly setting up PSU in your Organization, moving from your own project data, since PSU is an *open* technique which goal is to allow firstly an internal improvement for the sizing & estimation process before being used for external benchmarks.

There are three elements that need to be set-up:

1. The effort ranges from the WBS tasks analysis;
2. The weights expressing the complexity of tasks within such ranges;
3. The weights expressing the median proportionality of QM tasks against T tasks.

5.1 Tasks effort ranges

First element to set up is the number of effort ranges to take into account. Two possible criteria:

1. a pre-established, fixed number of complexity levels (i.e. three: High, Medium, Low)
2. statistically derived from a Pareto analysis

In both cases, a representative samples of WBS from past projects must be analyzed, taking care to the 'diagramming style' used by project managers in creating such WBS and the typical length by groups of tasks (by nature, by SLC phases, ….). The example ranges (H/M/L) used for the calculation example in Section 4 (Low complexity: 1-10m/d; Medium complexity: 11-24m/d; High complexity: from 25+m/d) falls into this first criterion.

Looking at the second criterion, it would be sufficient to statistically analyze those data with Pareto analysis for deriving here the ranges from the distribution of the m/d for such projects.

In both cases a meta-rule should be to obtain a picture the more representative as possible from projects. It would be not useful to choose to use three ranges only because it is a granularity level widely applied in statistical analysis. As said before, PSU is an *open* technique, mainly devoted to provide help for internal improvements.

About the frequencies for updating such ranges, this element, differently from the complexity weights, must be more stable during time, following a *'to-be' strategy* from the actual style the project managers of the Organization create their own WBS. In fact, a prerequisite to profitably use PSU is to have homogeneous input data (WBS) as much as possible towards a trend as the one shown in Figure 8 about expected ranges for estimation errors. This frequency could be – for instance – one time per year, but it must be calibrated according to the number of projects averagely managed per project manager during a certain period: the higher the value, the higher the frequency for re-evaluating the effort ranges from past WBS.

Next table shows an example with 5 possible revision in an Organization from the start-up (T_1) till a 'maturity' point (T_5).

Effort ranges (m/d)	T_1	T_2	T_3	T_4	T_5
High	51+	…	25+	…	11+
Medium	19-50	…	11-24	…	4-10
Low	1-18	…	1-10	…	1-3

A concurrent element that must be remembered is the 'minimization' strategy for reducing as much as possible the high-level complexity tasks (see the PSU Calculation Procedure, step 3b); it will contribute between two periods in refining directly in WBS the ranges values, but only if a project manager have been considered pragmatically useful such refinement during the project lifetime, otherwise the tasks complexity will remain the same.

The steps to run in order to obtain the effort ranges (*Pareto Analysis*[14]) are:

1. Collect the projects' effort data per task, putting them into a spreadsheet, one project per column;
2. Sort them in ascending order, from the lower one up to the higher one;
3. Find the cumulative counts. Each category's cumulative count is the count for that category added to the counts for all larger categories;
4. Create an histogram chart with two data series (data points, % for the cumulative value from the sorted frequencies);
5. Determine the main groups (number and effort range) from this sorted series.

Other '*qualitative'* information to analyse from the projects' data in order to establish the proper ranges are the maximum, median, average and minimum values for:

- Effort (m/d)
- Effort/task

An example will follow.

#Prj (n=11)	***Effort (m/d)***	***Tasks***	***Effort/task***	***Max***	***Median***	***Avg***	***Min***
P002	797	165	4.83	100	2.00	4.83	1.00
P005	743	158	4.70	100	2.00	4.70	1.00
P008	886	172	5.15	100	3.00	5.15	1.00
P014	810	156	5.19	100	3.00	5.19	1.00
P020	876	168	5.21	100	3.00	5.21	1.00
P021	723	144	5.02	100	2.00	5.02	1.00
P038	493	191	2.58	50	2.00	2.58	1.00
P039	950	359	2.65	150	2.00	2.65	1.00
P042	931	208	4.48	100	2.00	4.48	1.00
P043	898	206	4.36	100	2.00	4.36	1.00
P044	502	209	2.40	25	2.00	2.40	1.00

With these overall values for the 11 projects:

	Effort	**Effort/task**
Max	150	5.21
Median	2.00	4.70
Average	4.03	4.23
Min	1	2.40

That put in evidence a very detailed WBS but with some high-level complexity tasks making higher the average values (both for absolute effort and effort/task).

Counting the frequencies for the ordered effort values, we obtain this distribution (listed only the non-null values):

[14] It is possible to use the *Histogram* feature from the MS-Excel 'Analysis Tool Pak' add-in.

m/d	Freq	Cum.Eff	%		m/d	Freq	Cum.Eff	%
1	428	428	20.04%		19	1	2062	96.54%
2	958	1386	64.89%		20	20	2082	97.47%
3	186	1572	73.60%		22	8	2090	97.85%
4	244	1816	85.02%		23	7	2097	98.17%
5	82	1898	88.86%		25	1	2098	98.22%
6	5	1903	89.09%		30	9	2107	98.64%
8	80	1983	92.84%		36	8	2115	99.02%
9	12	1995	93.40%		37	7	2122	99.34%
10	52	2047	95.83%		40	2	2124	99.44%
14	8	2055	96.21%		50	2	2126	99.53%
15	3	2058	96.35%		88	1	2127	99.58%
16	2	2060	96.44%		100	8	2135	99.95%
18	1	2061	96.49%		150	1	2136	100.00%

And the following histogram chart:

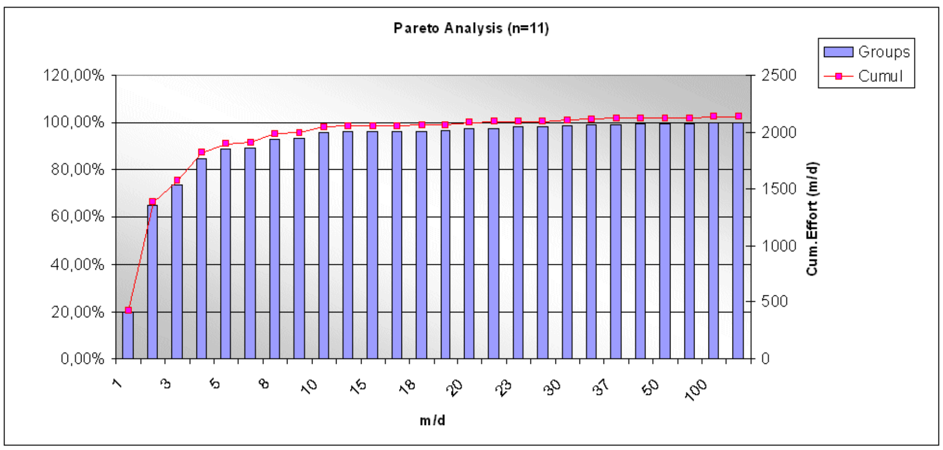


Fig. 12 – *Pareto Analysis (Histogram chart) with 11 projects*

From which it is possible to derive four groups (and relative ranges):

Range	Eff(Min)	Eff(Max)
High	20	+
Med-High	8	19
Med-Low	2	7
Low	0	1

While if a requirement is to maintain the three typical ranges (H/M/L), we would obtain:

Range	Eff(Min)	Eff(Max)
High	8	+
Medium	3	7
Low	0	2

5.2 Complexity weights

After determining the proper number of ranges for classifying task complexity, next step is to determine the proper weights for such ranges. This operation has the goal to determine periodically the best set of tasks weights allowing – on the current PHD data – to properly estimate next projects. Suppose to have 11 projects with the following detail:

project	PSU v1.2	Effort*	Effort	MRE	H	M	L
P001	342	1280	1236	-3.56%	3	7	327
P002	172	750	797	5.90%	5	7	153
P003	440	1680	1752	4.11%	4	8	422
P004	359	1648	1504	-9.57%	4	5	345
P005	163	900	743	-21.13%	4	5	149
P006	339	1459	1388	-5.12%	6	6	320
P007	212	980	1055	7.11%	4	8	194
P008	177	947	886	-6.88%	4	5	163
P009	276	1525	1308	-16.59%	6	7	255
P014	161	945	810	-16.67%	4	5	147
P015	266	1345	1200	-12.08%	5	6	249

With the tasks classified according to the initial set of example values:

Range	Eff (Min)	Eff (Max)	Weight
High	25	+	1.8
Medium	11	24	1.4
Low	0	10	1.0

The criterion to follow will be the maximization of R^2 recalculating it on the PHD projects; weights will be modified constant increases. This set of values can be also derived using regression analysis or manually creating a data series with such increases. Let's suppose 30 triple of points to verify, as in the following table, where w(8) is the current triple adopted:

w	L	M	H	w	L	M	H
1	1,00	1,05	1,10	16	1,00	1,80	2,60
2	1,00	1,10	1,20	17	1,00	1,85	2,70
3	1,00	1,15	1,30	18	1,00	1,90	2,80
4	1,00	1,20	1,40	19	1,00	1,95	2,90
5	1,00	1,25	1,50	20	1,00	2,00	3,00
6	1,00	1,30	1,60	21	1,00	2,05	3,10
7	1,00	1,35	1,70	22	1,00	2,10	3,20
8	1,00	1,40	1,80	23	1,00	2,15	3,30
9	1,00	1,45	1,90	24	1,00	2,20	3,40
10	1,00	1,50	2,00	25	1,00	2,25	3,50
11	1,00	1,55	2,10	26	1,00	2,30	3,60
12	1,00	1,60	2,20	27	1,00	2,35	3,70
13	1,00	1,65	2,30	28	1,00	2,40	3,80
14	1,00	1,70	2,40	29	1,00	2,45	3,90
15	1,00	1,75	2,50	30	1,00	2,50	4,00

Next steps will be to recalculate PSU for each project in the list according to each of the possible triples and then for each combination calculate R^2, obtaining which triple will determine the maximum forecasting probability for next projects.

In detail:

	P001	P002	P003	P004	P005	P006	P007	P008	P009	P014	P015	R2
w1	338	166	435	355	159	333	207	173	269	157	261	0,9404
w2	338	167	436	355	160	334	208	174	270	158	262	0,9406

w3	339	168	436	356	160	335	209	174	271	158	263	**0,9409**
w4	340	168	437	357	161	336	210	175	272	159	264	**0,9412**
w5	340	169	438	357	162	337	210	176	273	160	264	**0,9415**
w6	341	170	439	358	162	338	211	176	274	160	265	**0,9417**
w7	342	171	440	359	163	339	212	177	275	161	266	**0,9420**
w8	**342**	**172**	**440**	**359**	**163**	**340**	**213**	**177**	**276**	**161**	**267**	**0,9423**
w9	343	173	441	360	164	340	214	178	277	162	268	**0,9425**
w10	344	174	442	361	165	341	214	179	278	163	268	**0,9428**
w11	344	174	443	361	165	342	215	179	279	163	269	**0,9430**
w12	345	175	444	362	166	343	216	180	280	164	270	**0,9433**
w13	345	176	444	362	167	344	217	181	281	165	271	**0,9436**
w14	346	177	445	363	167	345	218	181	282	165	272	**0,9438**
w15	347	178	446	364	168	346	218	182	283	166	272	**0,9441**
w16	347	179	447	364	169	347	219	183	284	167	273	**0,9443**
w17	348	179	448	365	169	348	220	183	285	167	274	**0,9445**
w18	349	180	448	366	170	349	221	184	285	168	275	**0,9448**
w19	349	181	449	366	171	349	222	185	286	169	276	**0,9450**
w20	350	182	450	367	171	350	222	185	287	169	276	**0,9452**
w21	351	183	451	368	172	351	223	186	288	170	277	**0,9455**
w22	351	184	452	368	173	352	224	187	289	171	278	**0,9457**
w23	352	185	452	369	173	353	225	187	290	171	279	**0,9459**
w24	353	185	453	370	174	354	226	188	291	172	280	**0,9462**
w25	353	186	454	370	175	355	226	189	292	173	280	**0,9464**
w26	354	187	455	371	175	356	227	189	293	173	281	**0,9466**
w27	355	188	456	372	176	357	228	190	294	174	282	**0,9468**
w28	355	189	456	372	176	358	229	190	295	174	283	**0,9470**
w29	356	190	457	373	177	358	230	191	296	175	284	**0,9472**
w30	357	191	458	374	165	344	210	179	279	163	269	**0,9239**

where **w(29)** is the triple allowing to obtain the higher R^2, as also visible in next figure:

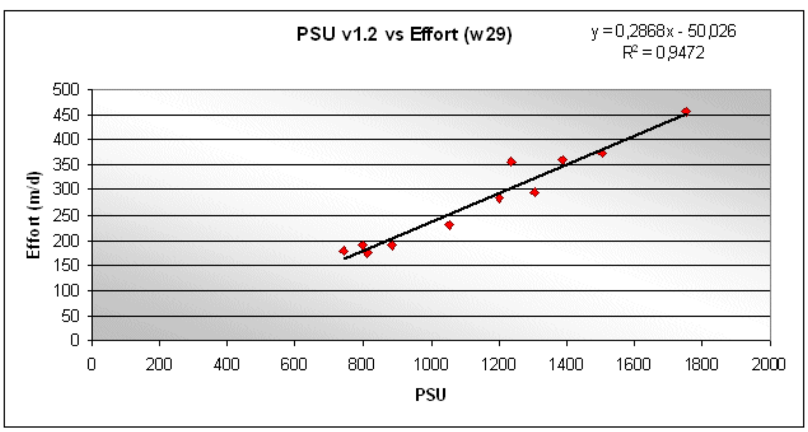


Fig. 13 – *Evaluation with the new task weights (w29)*

5.3 QM tasks

Last issue is about the determination and periodical re-evaluation of the proportionality between T and QM tasks in terms of size and effort spent in projects. Also in this case, as in the previous one, an analysis from PHD must be run, taking into account the overall effort and its distribution by QM vs T types per PSU ranges. Considering the same 11 projects of the previous example, and sorting them in an ascending order, it is possible it is possible to observe that for such range of PSU(t), moving from 112 up to 312, the median and average values for %PSU(qm) is quite stable around 30%

#	project	PSU v1.2	PSU(t)	PSU(qm)	%PSU(t)	%PSU(qm)	Effort
1	P014	161	112	49	69.48%	30.52%	810
2	P005	163	120	43	73.53%	26.47%	743
3	P002	172	130	42	75.79%	24.21%	797
4	P008	177	116	61	65.46%	34.54%	886
5	P007	212	145	68	68.17%	31.83%	1055
6	P015	266	180	86	67.72%	32.28%	1200
7	P009	276	190	86	68.80%	31.20%	1308
8	P006	339	242	98	71.23%	28.77%	1388
9	P001	342	239	103	69.96%	30.04%	1236
10	P004	359	262	97	72.94%	27.06%	1504
11	P003	440	312	128	70.94%	29.06%	1752
	Max	440	312	128	75.79%	34.54%	1752
	Median	266.40	180.40	86.00	69.96%	30.04%	1200.00
	Avg	264.44	186.22	78.22	70.36%	29.64%	1152.64
	Min	161	112	42	65,46%	24,21%	743

A first-level observation from this reduced dataset could be that the original proportionality table (on the left), created observing project with a lower size during past periods, could be refined adding more levels increasing size.

PSU_T	$P_{QM\,(\%)}$		PSU_T	$P_{QM\,(\%)}$
1-20	10%		1-20	10%
21-40	12%		21-40	12%
41-60	15%	➲	41-60	15%
61+	18%		61-90	18%
			91-120	22%
			121-160	25%
			161+	30%

6 PSU and Effort Estimation

The estimation of the time needed for producing the output for a certain activity/project is thus *function* of the chosen *sizing unit*. So:

$$Effort = f(size)$$

Function *f* to transform a time figure (man/days) for the project is derivable through algebra (*regression analysis*). There is no a predefined preferred kind of regression equation, but it is preferable to verify some hypothesis with more than one model (e.g. linear, exponential, logarithmic, …) and both with one and more independent variables (in the first case, typically the size, in the other case it could be size more the number of defects, the number of requirements, etc.)

Estimating with PSU, due to the possible distributions in different implementations of the number of M/Q/T tasks through projects, a suggested approach is to use – in case of regression analysis with more independent variables – to consider the number of PSU and the number of M/Q/T task or only those last ones.

6.1 Project Historical Database (PHD): essential data

Which are the essential data needed for calculating an effort estimation through regression technique? Every company should choice the proxies more significant to them for classifying projects and filter them in order to obtain clusters of projects. A well-known software project repository for FSMM is the ISBSG (International Software Benchmarking Standards Group - http://www.isbsg.org/): the list of fields used in the Release 10 (r10) is available at [ISBS07].

Anyway, it is suggested to consider two families of fields: Organizational and Technical ones, useful to create – through the desired queries – homogeneous clusters of projects to use as inputs for effort estimation:

Organizational Data

- Project Id.
- Project Manager
- Reference Market Segment (i.e.: Telecom, Public Sector, ...)
- Project Area in the Market Segment (i.e.: Telecom – CRM)
- Product-Technology for a certain typology in the Market Segment (i.e.: SAP/R3 – BSCS – Vantive)
- Project Type (i.e.: New Development, Corrective Maintenance, Enhancement Maintenance, ...)
- Software Life Cycle selected (e.g. Waterfall, Spiral, Prototype, etc.)
- Approach to the SLC selected (e.g. Sashimi, V-Shape, Pure for the Waterfall, etc.)
- …

Technical Data

- Sizing Unit (PSU, FP, #Entities)
- No. of sizing units (# SU) using at least two sizing units (e.g. PSU, FPA, UCP, etc.)
- Effort *predicted* in man/days by SLC phase (Planning/Control, Analysis/Design, Production, Test, Delivery, PR/CR) and by task typology (M/Q/T)

- *Actual* Effort in man/days by SLC phase (Planning/Control, Analysis/Design, Production, Test, Delivery, PR/CR) and by task typology (M/Q/T)
- Number of User Requirements
- Number of tasks in the project Gantt, classified by complexity (H/M/L)

In order to perform estimation, the use of "**Lifecycle**" field as a further filter to be chosen by PM only represents another qualitative parameter for project likeness, with the purpose to choice the group of projects closer – in terms of organisational/technical characteristics – to what will be the subject of estimation.

Technically, the selection of a "A" life cycle instead of a "B" one, does not *directly* affect estimation. Indeed, the effort estimated on the new project, with any selected life cycle, only has as an input the size and the effort of similar projects concluded and selected by the PM for the estimation. Furthermore, the verification of the reliability of the forecasting system, as indicated below, is performed through control indicators such as MRE, MMRE and PRED (0.25).

Further verification on this issue can be found in "closed" applications of regression analysis, as **COCOMO** (*Cost Construction Model*) by Barry Boehm, where both in the first version (1981) and in the second (1997) none of used parameters is related to the type of life cycle adopted in the project.

6.2 PHD population

The greater the number of tracked projects, the greater the probability to have to your disposal a set of relevant projects for properly estimating the effort needed for the new project. It is suggested to have at least 8-10 projects for each desired[15] typology, in order to profitably use those historical data to regression curves.

In the case the *Project Historical Database* (PHD) does not contain in a certain moment the same amount of project data types useful to a new estimation, it is suggested that the Project Manager – according to the "Activity Duration Estimation" (process #6.4 of PMBOK2004 [PMI04], will apply the first two estimation criteria, listed in the same order provided in section "*Tools & Techniques*", in detail:

- **Expert Judgement:** "*the judgement of an expert, guided by historical information, should be used where possible. If this expertise is not available, all estimation will be implicitly uncertain and risky*". In our case, with reference to consultation of historical information, the Project Manager can view data contained in PHD and the related MRE/MMRE and Pred (0.25), collecting all useful information to evaluate on the basis of his experience and competence.
- **Analogous Estimating:** "*the estimation according to analogy, also called top-down estimating, indicates the use of values of duration derived from similar activities as a basis for the estimation of durations of future ones. It is often used to estimate project duration, when there is little information details about the project (e.g. in the preliminary project phases). The estimation by analogy is a kind of "**expert judgement**"*.

At any rate, the number of PSU shall be calculated on the basis of available documents, in order to populate the PHD for future estimations.

[15] Typology means all the characteristics selected by PM in a specific case from the PHD according his needs (life cycle, type of development, sizing unit, etc.) from 1 to N possible filters on the database.

6.3 Estimation tools

There are a plenty of statistical tools to use for those estimation purposes. Without using sophisticated tools, also MS-Excel has basic capabilities using the *scattered diagram* type. A useful add-in is the "*Analysis ToolPak*", that allows several statistical analysis (ANOVA, Correlation, Covariance, etc.), also the Multiple Linear Regression analysis.

6.4 Estimating with PSU

The usual estimation process considers as main input, as said, the size of tasks to be performed in order to derive how much time should be needed to perform those tasks. Differently from a FSMM such as FPA, when using PSU as size unit for projects, it could seem that *it is the snake that bites its tail*: the Estimator would estimate the effort project value an in order to derive the same value using regression analysis?

In such case, the usual effort estimation procedure will assume a different "flavour": in fact, it will represent an iterative check for the Estimator to verify and adjust the initial number of man/days derived calculating the number of PSU, as from the calculation procedure (section 4).

Suppose to have selected ten projects from your PHD (Fig.14), sorted by descending MRE%, plotted on a scattered diagram, and after calculated R^2 using the linear regression:

Prj	PSU	Effort*	Effort	MRE%
P001	**42,66**	420,35	318,46	31,99%
P008	**26,21**	146,00	112,00	30,36%
P006	**42,10**	184,00	149,50	23,08%
P002	**119,85**	316,00	410,00	22,93%
P005	**18,34**	147,00	135,20	8,73%
P007	**16,75**	46,00	49,50	7,07%
P010	**89,16**	910,00	882,00	3,17%
P009	**66,85**	244,00	237,00	2,95%
P003	**34,60**	150,05	149,25	0,54%
P004	**22,00**	144,00	144,60	0,41%
Max	119,85	910,00	882,00	31,99%
Avg	47,85	270,74	258,75	**13,12%**
Med	38,35	167,03	149,38	7,90%
Min	16,75	46,00	49,50	0,41%
			Pred(0.10)	60,00%
			Pred(0.15)	60,00%
			Pred(0.20)	60,00%
			Pred(0.25)	80,00%

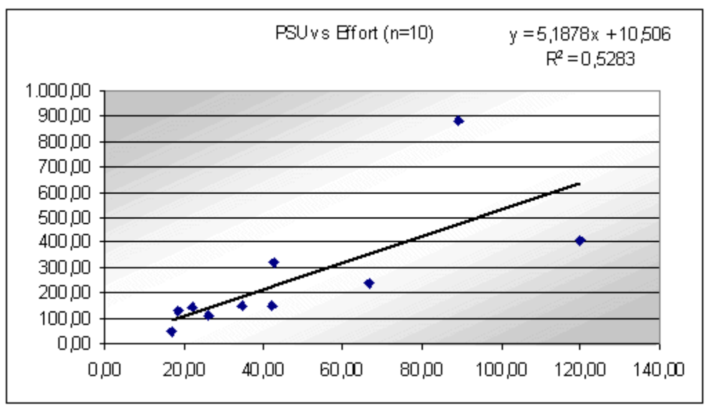


Fig. 14 – *Dataset with N=10 Projects and Linear Regression equation* ***(R^2=52.83%)***

The projects with a MRE%>25% were excluded and after it was recalculated the regression curve for n=6 using both linear (Fig.15) and logarithmic (Fig.16) equations:

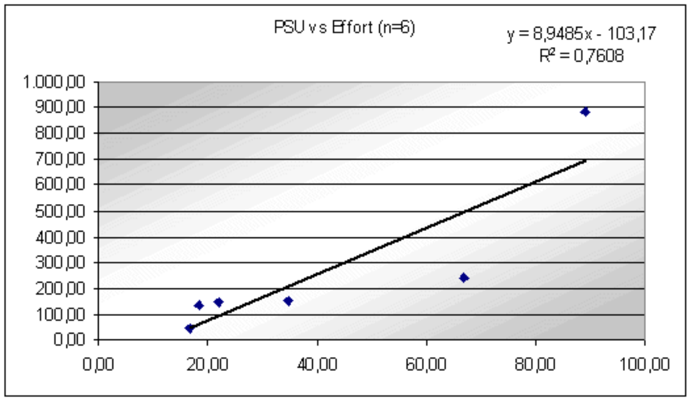


Fig. 15 – *Dataset with N=6 Projects and Linear Regression equation (R^2=76.08%)*

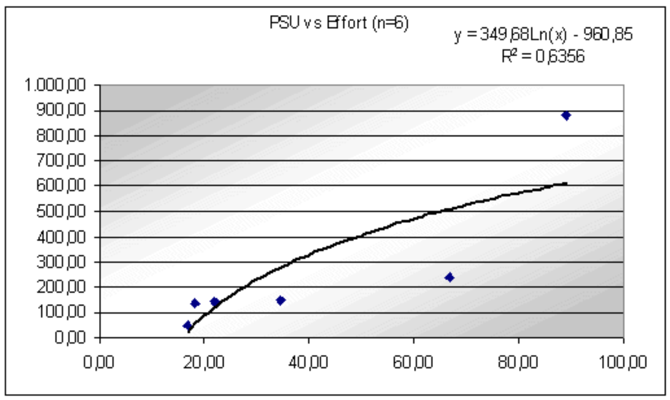


Fig. 16 – *Dataset with N=6 Projects and Logarithmic Regression equation (R^2=63.56%)*

Suppose that the Estimator, for a new project coded P011, will calculate a size of 61.34 PSU with an effort of 415 m/d. Applying the two equations, the result will be:

Prj	PSU	Effort*	Effort	MRE%			
P011	61,34	415	445,73099	Linear			R^2
					a	-103,17	0,7608
					b	8,95	
			478,584	Exponential			R^2
					a	-960,85	0,6356
					b	349,68	

Fig. 17 – *Summary for project P011*

The question is: which will be the proper number of m/d to estimate for P011? 415, derived directly from the PSU calculation or it could be more adequate to increase that number according to the regression analysis, even R^2 values no so high?

The Estimator – having all those values at his/her disposal – must decide if maintaining the initial estimation (415 m/d) or modifying it. In this second hypothesis, he/she has to redistribute in the PSU calculation sheet the m/d to add/subtract to the initial value. This change can influence the number of PSU. So, the revised PSU value will be applied another time to the two regression equations, deriving 2 new estimated values and R^2. And so on, the Estimator will iterate this calculation until the number of m/d will seem to him/her adequate to the new project.

Only when the project will be closed, it will be possible to evaluate with MRE% how much the estimation has been correct.

6.5 M/Q/T Tasks Classification: some examples

In Section 4.3, it was introduced a classification of project tasks into three possible categories:

- **M – Management**: tasks referred to Project Management, typically from ISO/IEC 12207 *organization* processes
- **Q – Quality**: tasks referred to Quality Management, Document and Configuration Management, typically from ISO/IEC 12207 *support* and *organization* processes
- **T – Technical**: tasks referred to the main activities for software development, those ones ISO/IEC 12207 calls *primary* processes

Here in the following, a list of possible tasks at lower levels to be included under the M/Q/T main leafs in the WBS:

M/Q/T	Main Task	Sub-Task$_1$	Sub-Task$_2$
M-Management			
	Scope		
		Determine Project Scope & Organization	
		Determine Preliminary Resources	
		Assign Development Staff	
		…	
		Scope Complete (milestone)	
	Control		
		Reviews	
		Working Progress Check	
		Invoicing Milestones (milestones - multiple)	
		…	
	Configuration Mgmt		
		Sw Configuration Mgmt Plan	
		Environment Implementation	
		…	
Q - Quality			
	Project Plan		
	Quality Plan		
	…		

T - Technical

	Analysis		
		Draft Preliminary UR	
		…	
		Review UR	
		Obtain Approval(s)	
		Final Analysis Documentation	
	Design	Draft SRS	
		Review SRS	
		Obtain Approval(s)	
		…	
		Final Design Documentation	
	Construction		
		Develop Code	
		Verify Code	
		Development Complete	
		Baseline Development Environment	
	Testing		
		Develop Test Plan	
		Review Test Plan	
		Baseline Testing documentation	
		Unit Test (UT)	UT execution
			UT reporting
			PR Management
			UT complete (milestone)
		Integration Test (IT)	
			IT execution
			IT reporting
			PR Management
			IT complete (milestone)
		System Test (ST)	
			ST execution
			ST reporting
			PR Management
			ST complete (milestone)
	Training		
		Develop Training Specifications	
		Identify training delivery methodology	
		Develop training materials	
		Training materials complete (milestone)	
		Training Sessions (Service)	
	Documentation		
		Develop User Documentation	
		Review User Documentation	
		Develop Installation Manual	
		Review Installation Manual	
		…	
		Documentation complete (milestone)	
	Delivery		
		…	
	Post-Delivery		
		…	

7 Conclusions & Prospects

It does not exist a single truth, but several concurrent viewpoints. As for all the possible choices to do, it will exist a trade-off point, above or below which it will be more convenient to adopt an early or a standard estimation technique. The question about the opportunity in adopting an early estimation technique derives exclusively from the moment such information must be available, not always coincident with the end of the Design phase.

The functional measurement logic, expressed in FSM (Functional Size Measurement) methods such as FPA represent absolutely the right direction towards which continuing the path. Retrieving such logic, an early estimation system such PSU brings lower costs but a reduced affordability of the related forecasting system for sizing a project. The correlation between the estimation results produced both with a standard and an early technique produces for sure proper values in order to evaluate such trade-off.

PSU technique, created in 2003, can be applied for estimating future development projects, trying to optimise the effort for sizing a project and minimizing the estimation error.

A list of **FAQ** will be also maintained on the PSU webpage (*http://www.geocities.com/lbu_measure/psu/psu.htm*).

--- ***End of the Document*** ---