Multidimensional Software Performance Measurement Models: A Tetrahedron-based Design

Luigi Buglione
Facoltà di Economia
Università di Roma "La Sapienza"
Via del Castro Laurenziano, 9
I-00161 Rome, Italy
E-mail: buglione@tin.it, buglione@scec.eco.uniroma1.it
Tel: (39) 338 95.46.917
Fax: (39) 06 56.21.768

Alain Abran
Software Engineering Management Research Laboratory
Université du Québec à Montréal
C.P. 8888, Succ. Centre-Ville
Montréal, Québec, Canada
E-mail: abran.alain@uqam.ca
Tel: (514) 987-3000 (8900)
Fax: (514) 987-8477

Abstract – This work presents an improved version of an *open multi-dimensional model* of performance, called **QEST** (**Q**uality factor + **E**conomic, **S**ocial and **T**echnical dimensions) [8]. Performance is defined here as productivity adjusted by quality, both of which can be represented from multiple viewpoints. The QEST model integrates into a single representation three dimensions, each one represented by a productivity measurement value derived from an instrument-based measurement process, which value is then adjusted by a perception-based measurement of quality achieved. Both components of performance, that is productivity and quality, take into account the same three distinct viewpoints of performance:

- **economic dimension**, the perspective is the *managers'* viewpoint, with particular attention paid to cost and schedule drivers;
- **social dimension**, the perspective is the *users'* viewpoint, with particular attention paid to the quality in use drivers;
- **technical dimension**, the perspective is the *developers'* viewpoint, with particular attention paid to technical quality.

Keywords – Performance Measurement, Software Product Quality, Metrics, Function Point Analysis, ISO/IEC 9126.

1 Introduction

Measures are increasingly being recognized as fundamental to adequately assessing current software practices and software products and to setting realistic targets when designing improvement programs.

The focus of this work is software products, which are to be measured and assessed through a three-dimensional measurement model with the ability to handle independent sets of dimensions without predefined weights.

Such a generic three-dimensional structure will allow organizations to choose the components of each dimension according to their own needs and will give them the ability to select relevant measures and to implement them. This type of model will be referred to here as an *open multi-dimensional model of performance.*

The paper is subdivided into three parts:

- a descriptive part, presenting the conceptual aspects of this open model, including its structure and components;
- a mathematical part, presenting the mathematical expressions for tetrahedron-based geometrical figures;
- a procedural part [2], with the description of the steps required to implement the model in a specific environment.

2 Measurement of performance

In a competitive market period such as the current one, company competitiveness strongly depends on myriad types of factors such as the capability to react on time to customers' requests and the minimization of costs of goods and services offered. Monitoring these factors and their impact on the development process is increasingly critical. Therefore, measuring performance levels becomes a key component for improving the planning, monitoring and delivery of goods and services, as well as for the design of improvement programs.

Performance is not a single one-dimensional concept: it is not enough to meet a specific target in an unconstrained environment. It is a multidimensional concept that must integrate multiple viewpoints, most of which are present simultaneously in the software development process, such as:

- the **E**conomic one, represented by the managers' viewpoint;
- the **S**ocial one, represented by the users' viewpoint;
- the **T**echnical one, represented by the developers' viewpoint.

Performance models in the software engineering literature mostly take into consideration the first and third of the viewpoints listed above, and handle them separately. Because of a growing involvement of users with computer technologies, the second viewpoint should also be taken into account in software assessment, thereby adding complexity to performance measurement when three dimensions have to be taken into account simultaneously. If the three dimensions can be handled into account concurrently in an integrated mode, then such types of models can more adequately represent performance measurement.

In the literature survey, three studies were identified as dealing with multidimensionality in software performance measurement, although sometimes from distinct perspective of performance, as well as three distinct approaches:

- in Gonzales [11] a **vectorial approach** is proposed to measure software complexity, always considering a 3D space; the three *dimensions* are given by Length, Time and Level for each of the three complexity domains (Syntactical, Functional, Computational) with a list of predefined and non-normalized complexity factors and metrics;
- in Hatfield [12], the measurement of product performance is defined as the single viewpoint related to product assessment (asset / customer-project / strategic management) as a *dimension*, but it represents the 3D concept through a **cube** and uses only a single non-normalized metric per dimension;

- in Donaldson & Siegel [10], *n* different normalized metrics are used to define the "product integrity value" (and not the single interest group) as a *dimension* using a vectorial approach, representing the concept in a 2D space through the use of **Kiviat graphs**.

The model proposed in this paper to combine these assessments from the three dimensions within a single value to determine performance is referred to as the **QEST model** (**Q**uality, and **E**conomic, **S**ocial and **T**echnical productivity).

The QEST model proposes the use of a certain number of measures not predetermined by the model itself (not as in Gonzales and Hatfield) and expresses the performance measuring concept with a 3D construction (rather than a 2D one, as in Donaldson & Siegel), and with a pyramidal representation (3 sides - 3 viewpoints) rather than a cubic one (4 sides - 3 viewpoints, as in Hatfield); QEST proposes a geometrical representation or performance with the same number of sides as the number of viewpoints considered.

Another unusual feature of the QEST model is the following: the measurement of performance (**p**) is given by the *integration* of an instrument-based measurement process (expressed in the model by the component **RP** - *Rough Productivity*) with a perception-based measurement process based on the subjective perception of quality - and expressed in the model by the component **QF** - *Quality Factor*)[1].

In summary:

Performance = Productivity and Quality

Performance = PR and Q

Performance measurement = (Instrument-based measurement of Productivity) and (Perception-based measurement of Quality)

3 The structure of the QEST model

The QEST model provides a multi-dimensional *structured shell* which can then be filled according to management objectives in relation to a specific project, and can therefore be referred to as an *open model*. This section presents the design of this *open model* for the measurement of software project performance, making it possible to handle the multiple and distinct viewpoints already discussed, all of which exist concurrently in any software organization.

The basic purpose of the structured shell of the open model is, as stated above, to express performance as the combination of the specific measures (or sets of measures) selected for each of the three dimensions, these values being derived from both an instrument-based measurement of productivity and a perception-based measurement of quality.

[1] In everyday life, when a person needs to buy a technology object (such as software), he first of all looks at the technical requirements (quantitative assessment) and only then does he consider qualitative aspects, like usability, aesthetics, etc. But the final assessment is unquestionably given by the combination of the two sets of criteria.

A three-dimensional geometrical representation of a tetrahedron was selected as the basis of the model and is illustrated in Figure 1.

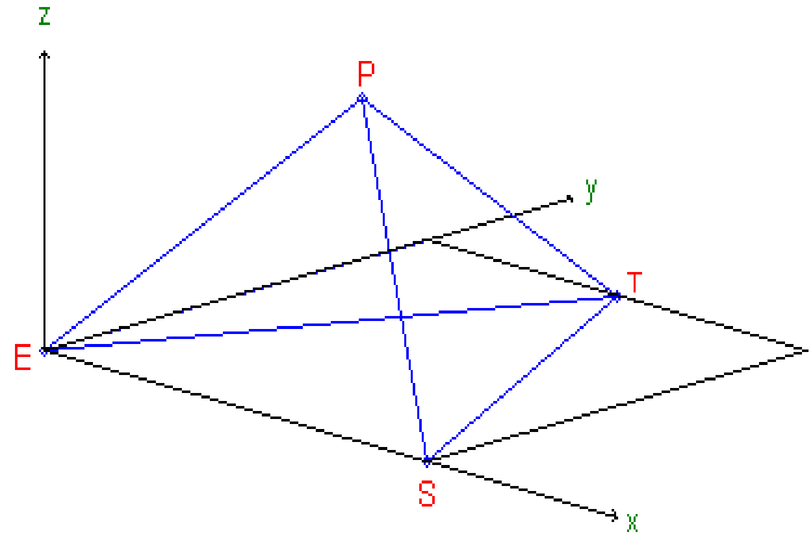


Figure 1: *Regular tetrahedron with E, S, T dimensions as base axes and performance P as vertex*

This open model can be represented as a *regular tetrahedron* in a three-dimensional space where[2]:

- the three dimensions (E, S, T) in the space correspond to the pyramid base corners and the convergence of the sides, the P vertex, which describes the top performance level;
- when the three sides are of equal length, the solid shape that represents this three-dimensional concept is therefore a pyramid with its triangular base and sides of equal length (*tetrahedron*).

This pyramid-type representation imposes the following constraint: the sides must be equal, and this is achieved through giving equal weights to each of the three different dimensions chosen – and with sides of length exactly equal to 1 (*regular tetrahedron*); in this way, the dimensions are represented through a normalized value between 0 and 1 for each of them on a ratio scale, for ease of understanding.

For any specific project, its value on each dimension is given by the weighted sum of a list of *n* normalized measures having been selected as representative of each of the three viewpoints. The selection of the specific measures within each dimension is an implementation issue.

[2] To obtain a less complex geometrical formula, the ES line on all Figures is put on the X axis, which is just one of the possible positions it can assume in a 3D space. Theother sides of a tetrahedrom being at 60 degrees, the other sides will not be on the y or z axes. Knowing the geometrical relations between measures in a regular triangle with sides of unit length, it is possible to obtain the other point coordinates easily. Note that H point represents the center of the base of the regular tetrahedron, expressed in Figure 3.

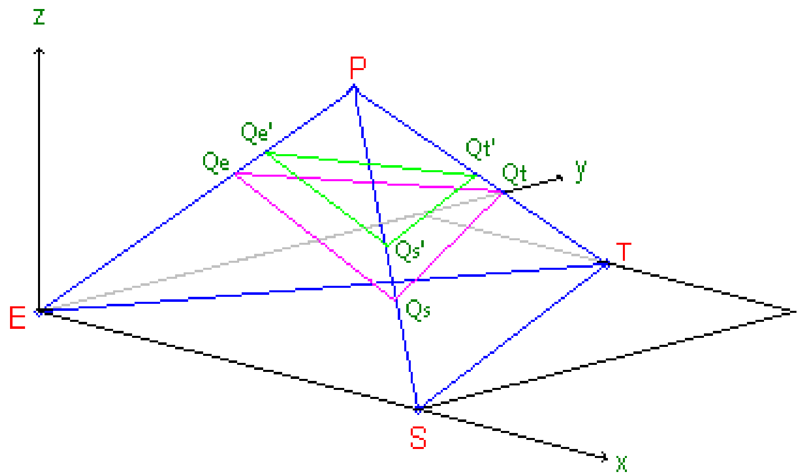


Figure 2: *(Qe,Q s,Q t) and (Qe',Q s',Q t') plane sections*

The values of the three dimensions, shown in Figure 2 as (Qe, Qs, Qt), each placed on its tetrahedron side, describe a sloped plane section in the space and return the three-dimensional productivity measurement.

The three-dimensional measure of quality (QF) represents a three-dimensional weighting factor of the productivity measurement representation. From a geometric viewpoint, it is representable through an upward or downward translation of the (Qe, Qs, Qt) section describing the new (Qe', Qs', Qt'), if the project quality characteristic value is greater or smaller than a predetermined quality target for each dimension. The determination of specific target values is again an implementation issue.

Refer to the new sloped section for every calculation.

According to some notions in analytic geometry [6, 19], it is possible to determine the equation of the (*Qe', Qs', Qt'*) sloped section, the one referred to as the arithmetic sum of the productivity and quality measurement.

Consider the economic dimension, for example. The formula for determining the translated point will be $\boxed{e' = e + QF}$, where *e* is the Economic Productivity value and *QF* the Quality Factor.

After (*e, s, t*) index calculation and knowing the E, S, T, P coordinates:

$$E = (0,0,0); S = (1,0,0); T = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right); P = \left(\frac{1}{2}, \frac{1}{2\sqrt{3}}, \frac{\sqrt{6}}{3}\right); H = \left(\frac{1}{2}, \frac{1}{2\sqrt{3}}, 0\right)$$

Expressing (*e, s, t*) coordinates in the 3D space with *Qe, Qs* and *Qt* as follows:	and (*e', s', t'*) coordinates with *Qe', Qs'* and *Qt'* in the same way:

$\begin{cases} Qe = E + e \bullet \vec{EP} = \left(\frac{1}{2}e, \frac{1}{2\sqrt{3}}e, \frac{\sqrt{6}}{3}e\right) \\ Qs = S + s \bullet \vec{SP} = \left(1-\frac{1}{2}s, \frac{1}{2\sqrt{3}}s, \frac{\sqrt{6}}{3}s\right) \\ Qt = T + t \bullet \vec{TP} = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}-\frac{t}{\sqrt{3}}, \frac{\sqrt{6}}{3}t\right) \end{cases}$	$\begin{cases} Qe' = E + e' \bullet \vec{EP} = \left(\frac{1}{2}e', \frac{1}{2\sqrt{3}}e', \frac{\sqrt{6}}{3}e'\right) \\ Qs' = S + s' \bullet \vec{SP} = \left(1-\frac{1}{2}s', \frac{1}{2\sqrt{3}}s', \frac{\sqrt{6}}{3}s'\right) \\ Qt' = T + t' \bullet \vec{TP} = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}-\frac{t'}{\sqrt{3}}, \frac{\sqrt{6}}{3}t'\right) \end{cases}$

Then, starting from the generic equation of a plane in a 3D space:

$$\Pi: \begin{vmatrix} X-x_1 & Y-y_1 & Z-z_1 \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix} = 0$$

it is possible to obtain the sloped section equation:

$$X\,\frac{(s't'-s'-e't'+e')}{\sqrt{2}} + Y\left(\frac{(s'-2e's'+e'-2t'+s't'+e't')}{\sqrt{6}}\right) + Z\,\frac{(3-2e'-2s'-2t'+e's'+e't'+s't')}{2\sqrt{3}} + \frac{(e's'+e't'-e'-e's't')}{\sqrt{2}} = 0$$

The sloped section coefficients are therefore:

$$\begin{cases} a = \frac{(s't'-s'-e't'+e')}{\sqrt{2}} \\ b = \frac{(s'-2e's'+e'-2t'+s't'+e't')}{\sqrt{6}} \\ c = \frac{(3-2e'-2s'-2t'+e's'+e't'+s't')}{2\sqrt{3}} \\ d = \frac{(e's'+e't'-e'-e's't')}{\sqrt{2}} \end{cases}$$

With this 3D representation (the sloped plane), it is possible to determine and represent performance considering at least three distinct geometrical concepts:

- the **distance** between the center of gravity of the tetrahedron base and the center of the plane section along the tetrahedron height – the greater the distance from zero, the higher the performance level. The inclination angle of the section also represents here additional information about dimensions.

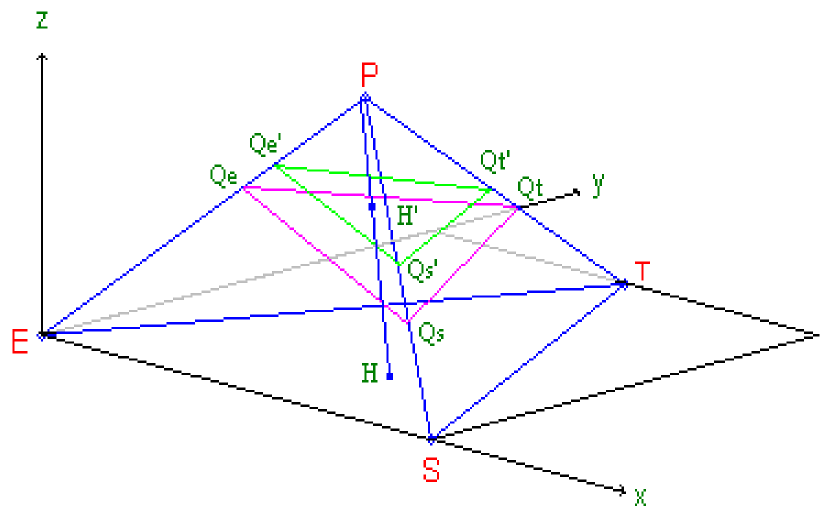


Figure 3: *HH' distance*

Z is the height of the intersection point between Π plane and the perpendicular straight line to the P vertex. Substituting X and Y values with those of the P point, it is possible to obtain Z as follows:

$$Z = \frac{1}{c}\left(-\frac{a}{2} - \frac{b}{2\sqrt{3}} - d\right)$$

where *a*, *b*, *c*, *d* are the generic coefficients of the sloped section equation.
In the end, this Z value must be translated into the corresponding percentage term. In fact, the height of a regular tetrahedron is equal to $\sqrt{6}/3$, and so the final formula for expressing the *p* index through the distance *g* is the following:

$$\boxed{p = \frac{g}{\sqrt{6}/3}}$$

- the **area** of the sloped plane section – the smaller the area, the higher the performance level. Additional information is also given by the inclination angle of the plane, indicating the best and worst dimensions.

It is possible to make the sloped section area calculation by means of Erone's formula. It is sufficient to know the length of the three sides of the *(Qe', Qs'. Qt')* triangle, called *a*, *b*, *c*, derived as the distance between two points in a 3D space:

$$\begin{cases} a = \left|Qe'\,Qs'\right| = \sqrt{(x_e - x_s)^2 + (y_e - y_s)^2 + (z_e - z_s)^2} = \sqrt{1 + e'^2 + s'^2 - e' - s' - e'\,s'} \\ b = \left|\vec{Q}e'\,Qt'\right| = \sqrt{(x_e - x_t)^2 + (y_e - y_t)^2 + (z_e - z_t)^2} = \sqrt{1 + e'^2 + t'^2 - e' - t' - e'\,t'} \\ c = \left|\vec{Q}s'\,Qt'\right| = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2 + (z_s - z_t)^2} = \sqrt{1 + s'^2 + t'^2 - s' - t' - s'\,t'} \end{cases}$$

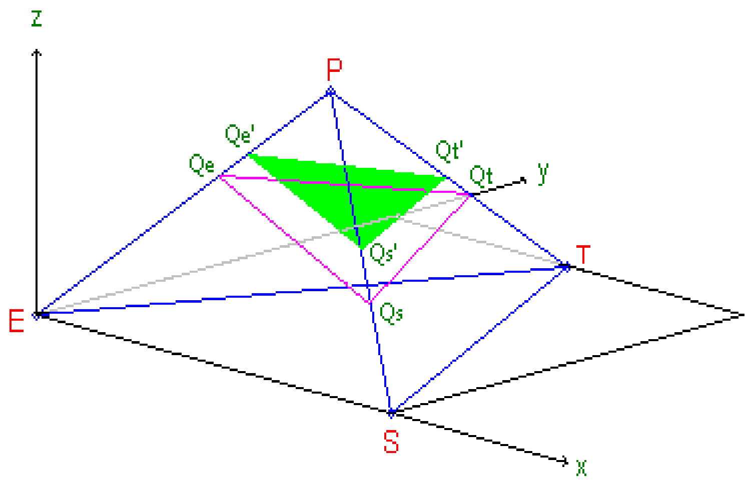


Figure 4: *(Qe', Qs'. Qt') Area*

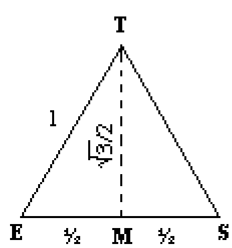
The area can be obtained with the following formula:

$$A = \sqrt{sp(sp-a)(sp-b)(sp-c)}$$

where *sp* means the semiperimeter of the triangle. The *p* value is equal to:

$$\boxed{p = 1 - \frac{A}{A_{\max}}}$$

where *Amax* is the maximum area value a triangle can have inside a regular tetrahedron, and corresponds to the area of the tetrahedron base, which is equal to $\sqrt{3}/4$ (see Figure 5). The smaller the area, the greater the *p* value. So, it is necessary to consider the ratio between the difference (*Amax* - *A*) on the *Amax* to arrive at an adequate value.



$$TM = \sqrt{ET^2 - EM^2} = \sqrt{1^2 - \left(\frac{1}{2}\right)^2} = \frac{\sqrt{3}}{2}$$

$$A_{est} = \frac{B \bullet h}{2} = \frac{1 \bullet \sqrt{3}/2}{2} = \frac{\sqrt{3}}{4}$$

Figure 5: *Tetrahedron Base Area*

- the **volume** of the lower part of the truncated tetrahedron – the greater the volume, the higher the performance level.

The total volume of a regular tetrahedron is equal to: $V_{TOT} = \frac{l^3\sqrt{2}}{12}$ and, since in this case l=1, the total volume is: $V_{TOT} = \frac{\sqrt{2}}{12}$

It is possible to calculate the volume of the truncated tetrahedron as the difference between the total volume and the volume of the upper solid shape delimited by the sloped section.

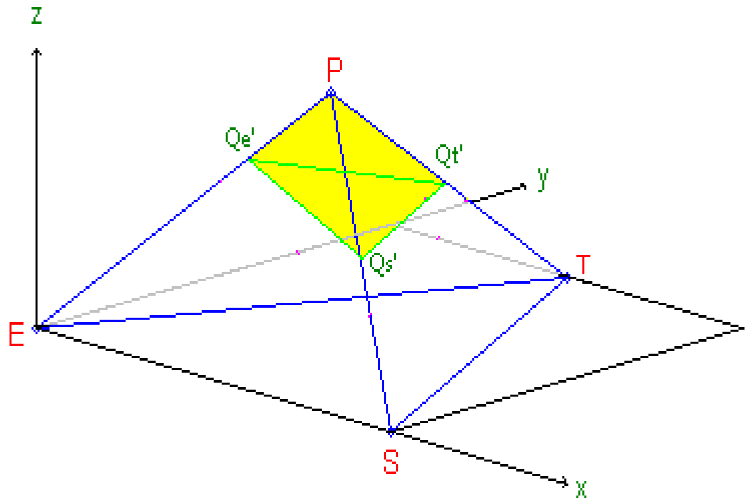


Figure 6: *V Volume*

It is possible to calculate this volume by determining the distance between the sloped section and the tetrahedron vertex. This distance can be considered to be the height of this oblique pyramid. Substituting the (a, b, c, d) coefficients of the sloped section equation in the following formula

$$h = \frac{|ax + by + cz + d|}{\sqrt{a^2 + b^2 + c^2}}$$

it is possible to obtain the volume from the well-known pyramid volume formula $\boxed{V = \frac{B \bullet h}{3}}$, where B means the sloped section area, calculated above.

So, the *p* value in this last case is equal to:

$$\boxed{p = 1 - \frac{V}{V_{TOT}}}$$

This third type of geometrical information of course carries more information than the previous two types, and it is the concept chosen for our model.

Exceptions. The above-cited formulas are not valid in a few particular cases. In fact, if a couple of the triples of the (e', s', t') values are equal to 1, the only formula for determining *p* is:

$$\boxed{p = M_x(e', s', t')}$$

because the sloped section becomes a straight line (couple) of a single point (triple), and it is no longer possible to determine the distance, the area or the volume as presented earlier.
Therefore, the formula for determining p is:

$$\left\langle \begin{array}{ll} p = M_x(e',s',t') & \textit{if } e'=s'=1;\ e'=t'=1;\ s'=t'=1;\ e'=s'=t'=1 \\ \begin{cases} p = \dfrac{g}{\sqrt{6}/3} \\ p = 1 - \dfrac{A}{A_{max}} \\ p = 1 - \dfrac{V}{V_{tot}} \end{cases} & \textit{other values} \end{array} \right.$$

4 Implementation of requirements

A procedure is presented now for the implementation of the QEST model. This procedure follows the **P**lan-**M**easure-**A**ssess-**I**mprove (**PMAI**) cycle, which conforms to the Shewhart and Deming PDCA cycle [9], as shown in Figure 7[3]:

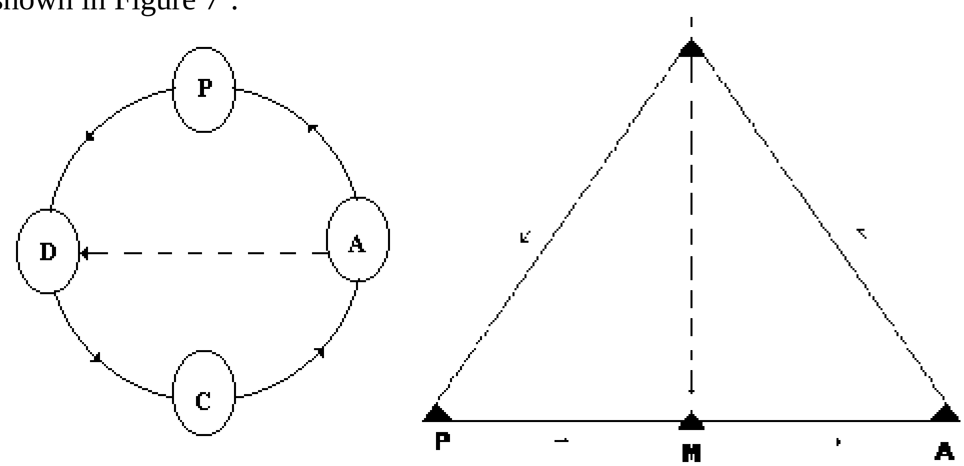


Figure 7: *PDCA cycle (on the left) and PMAI cycle (on the right)*

PLAN
1. Determination of measurement guidelines
2. Selection of representative measures for each dimension
3. Determination of relative importance between productivity and quality in the assessment of performance for a specific project, or set of projects
4. Determination of ratio weights
5. Establishment of acceptability threshold values

[3] The new figure is derived from the well-known PDCA representation, and is obtained through a 90° left rotation and by positioning the fourth phase on the vertex of the triangle. The reason for this is that it results in a better fit with the model concept.

MEASURE

6. Data gathering
7. Application of numerical assignment rules
8. Normalization of the ratios
9. Calculation of QF, V and p

ASSESS

10. Presentation of measurement results
11. Analysis on the observed values

IMPROVE

12. Process Improvement

Various authors have proposed many distinct definitions of **quality**, but, for the purposes of this work, the definition of quality in [15] was selected: "the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs." Referring to product quality, specifically that of software, it must be interpreted in light of the concept of **purpose of use**, considering both internal attributes (product characteristics) and external ones (aim of use).
Therefore, software quality must also be viewed as the concurrent integration of the three different viewpoints previously mentioned:

- viewpoint of the **user**, for whom software quality is achieved by all the properties required to satisfy correctly and efficiently the present and future real needs of whoever buys and uses it;
- viewpoint of the **developer**, for whom software quality is achieved by "conformity to functional and performance requirements explicitly stated, to development standards explicitly documented and to implied characteristics supposed for every software developed in a professional way" [18];
- viewpoint of **management**, who are "interested in overall quality rather than in a specific quality characteristic [...] and the need to balance the quality improvement with management criteria" [16].

The relative mix of both the productivity and quality measurement within a single specific corporate model will be determined in each instantiation of the model by the corporate Metric Working Group (**MWG**), which will determine the right proportion between the two components for the project being assessed.

To facilitate ease of understanding and a greater applicability of the examples, *de facto* and *de jure* standards are recommended for the selection of the various measures within specific dimensions, such as:

- **ISO/IEC 9126 –** the ISO list of software product quality characteristics and sub-characteristics **–** used in the Social dimension and the QF calculation;
- **Function Point Analysis**[4] **–** FPA measurement results can be used for both the Economic and Technical dimensions.

---

[4] The reason for the choice of Albrecht's Function Points Analysis [3, 4, 13, 14] can be summarized in the following points:

- **technology-independent**: an analysis based on external vision of product functionalities permits comparison of products written in different programming languages;
- **pre-development measurability**: Function Points Analysis (FPA) can be also used to estimate application size in the planning phase;

BRAN, A., *Quality – The Intersection of Product and Process*, The 6th IEEE International Software Engineering Standards Symposium, ISESS'95, Montréal, Québec, Canada, August 21-25 1995, IEEE Computer Society Press.

[2] ABRAN, A. & BUGLIONE, L., *Implementation of a Three-Dimensional Software Performance Measurement Model*, Technical Report, Université du Québec à Montréal, to be published, 1998.

[3] ALBRECHT, A.J., Measuring Application Development Productivity, *Proceedings of the IBM Applications Development Symposium, GUIDE/SHARE*, Monterey, CA, 14-17/10/79, 83-92.

[4] ALBRECHT, A.J., *IBM CIS & A Guideline 313, AD/M Productivity Measurement and Estimate Validation*, 01/11/84.

[5] BASILI, V.R. & ROMBACH, H.D., *The TAME Project: Towards Improvement-Oriented Software Environment*, IEEE Transactions on Software Engineering, IEEE Computer Society, Vol. 14, No. 6, June 1988, 758-773.

[6] BORSUK K., *Multidimensional Analytic Geometry*, PWN, Warsaw, 1969

[7] BUGLIONE, L., *Graphical User Interface (GUI): vantaggi tecnici, economici e sociali*, Tesi di Laurea, Università "La Sapienza", Roma, 1995.

[8] CAVALLO, A. & BUGLIONE, L., *A 3D Software Productivity Measurement Model*, in "Software Quality Engineering", Eds: C. Tasso, R.A. Adey & M. Pighin, 1st Software Quality Engineering Conference (SQE 97), May 5-7 1997, Udine, Italy, CMP, 1997, 191-200.

[9] DEMING, W.E., *Out of the Crisis*, MIT Press, 1986.

[10] DONALDSON, S.E. & SIEGEL, S.G., *Cultivating Successful Software Development: A Practitioner's View*, Prentice Hall, 1997.

[11] GONZALEZ, R.R., *A Unified Metric of Software Complexity: Measuring Productivity, Quality and Value*, The Journal of Systems and Software, Elsevier Science North-Holland, Vol. 29, No. 1, April 1995, 17-37.

[12] HATFIELD, M.A., *Managing to the Corner Tube: Three-Dimensional Management in a Three-Dimensional World*, IEEE Engineering Management Review, IEEE Computer Society, Vol. 23, No. 3, Winter 1995, 63-68.

[13] IFPUG, *Function Points as Assets: Reporting to Management*, 1992.

[14] IFPUG, *Function Points Counting Practices Manual (release 4.0)* – Chairperson Robin Ragland, Atlanta, Georgia, January 1994.

[15] ISO, *International Standard 8402: Quality - Vocabulary*, 1986.

[16] ISO/IEC, *International Standard 9126: Information Technology - Software product evaluation – Quality characteristics and guidelines for their use*, 1991.

[17] NASA/GSFC, *Software Measurement Guidebook, Revision 1,* Software Engineering Laboratories Series, SEL-94-102, June 1995.

[18] PRESSMAN, R., *Software Engineering: a beginner's guide*, McGraw Hill, 1988.

[19] SERNESI E., *Geometria I*, 1/e, Bollati Boringhieri, Torino, 1989

• ***de facto* standard**: Function Points Analysis is widely accepted in the international MIS community. FPA measurement rules are constantly being reviewed by the **I**nternational **F**unction **P**oint **U**sers **G**roup (IFPUG).