A Model for Performance Management and Estimation

Luigi Buglione and Alain Abran,
École de Technologie Supérieure (ETS) luigi.buglione@computer.org, aabran@ele.etsmtl.ca

***Abstract* —Traditional cost estimation models in software engineering are based on the concept of productivity defined as the ratio of output to input; for instance, detailed software estimation models, such as COCOMO, can take multiple factors into account, but their multipliers lead to a single perspective based on the productivity concept. A less explored relationship in software engineering is the one between productivity and performance. This paper presents some classic concepts on the multidimensionality of performance, and proposes some suggestions to implement multidimensional performance models in software engineering based on certain fundamental concepts from geometry, that is, the QEST/LIME family of models.**

***Index Terms*— Performance Management, Estimation, CMMI, COCOMO, QEST/LIME**

I. Introduction

In the CMMI process improvement model, project estimation is addressed at two distinct levels: first at level 2, based on the productivity concept, and next at level 4, where estimation refers to the performance concept. Even though for many people productivity and performance are used interchangeably, in econometrics and management sciences they are considered as quite distinct concepts. This section presents some key concepts which help distinguish productivity from performance.

A. Productivity-based estimation

The concept of productivity is typically defined as the ratio of the output produced to its corresponding number of inputs. In software engineering, output is typically measured in terms of software size, e.g. LOC or Function Points, and input in terms of project effort in hours. For a project manager starting a project, adequately estimating project costs is quite challenging [1]: he can use a number of techniques ranging from expert opinion taking into account productivity, at least intuitively, to parametric models derived from productivity analysis of historical data.

Well-known Software Process Improvement (SPI) models, such as CMMI [2] or SPICE – ISO 15504 [3], consider the cost estimation practice as a set of steps, the outcome of which is an aggregation of many productivity-related process factors, leading to a single cost figure.

In the Project Planning process of the CMMI, the estimation practice PP SP1.4-1 – *Determine estimates of effort and costs* states[1]: "*Collect the models or historical data that will be used to transform the attributes of the work products and tasks into estimates of the labor hours and cost*," and it specifies that "*Historical data include the cost, effort, and schedule data from previously executed projects, plus appropriate scaling data to account for differing sizes and complexity*."

B. Performance

Most often, software project estimation models refer primarily to effort and costs without explicitly tackling other project attributes such as quality and innovation. It is worth nothing that, in the Project Planning process of the CMMI, the term *performance*[2] is never used in estimation practice PP SP1.4-1.

In the CMMI, project and process performance is explicitly included in two other Process Areas, which are: Organizational Process Performance (OPP) and Quantitative Project Management (QPM), both of which are at level 4 – see Figure 1.

A frequent question about OPP in technical fora and in mailing lists on CMMI is: What does the CMMI[3] consider as a process performance model (PPM)? Which concepts are needed for adequately understanding, managing and estimating performance? Is effort size sufficient to estimate project performance, or should some further concepts be taken into account?

The CMMI provides the following explanation of performance, within the context of performance estimation "*used to estimate or predict the value of a process performance measure from the values of other process and product measurements,*" and lists Complexity [5] and Reliability Growth Models [6].

The CMMI model states in the OPP PA that productivity

[1] Within SG1 (*Establish Estimates*), sub-practice 1

[2] Defined as "*the degree to which a system or a component accomplishes its designated functions within given constraints*" [4].

[3] See for instance: http://groups.yahoo.com/group/cmmi_process_improvement/ (June 26, 2004).

is an attribute of performance[4] (Fig. 1), thereby leaving it to the organizations themselves to figure out the performance model relevant to their context.

C. Relationships between performance and effort-cost estimation

In performance models, a number of other ratios can be used to refine overall project estimates: for the testing phase, for instance, by taking into account a ratio of effort based on a relevant percentage of man/days within the Software Life Cycle (SLC) model selected. A performance estimation model could also take into account some expected defect ratios (defect rate and defect density) from *reliability* models, expected complexity of the project and test assets to be produced and verified.

In Sink [7], performance is fundamentally a multidimensional perspective of concurrent viewpoints, the outcomes of which impact productivity (Fig.2). It includes such dimensions as quality of work life, efficiency, effectiveness and quality, each impacting productivity, which in turn impacts profitability. Innovation is another dimension to introduce change in any of the previously mentioned dimensions with the end purpose of improving both productivity and profitability.

Last but not least, performance management and estimation should be performed considering the various stakeholders' viewpoints simultaneously in a quantitative manner, and the estimates should be revisited throughout the development cycle phases.

II. MULTI-DIMENSIONAL PERFORMANCE MODELS

A. The QEST/LIME models [8]

QEST (**Q**uality factor + **E**conomic, **S**ocial and **T**echnical dimensions) is a software multidimensional performance measurement model [8] developed using geometrical concepts and applied to the measurement of software projects. In the initial version of the QEST model, the measurement of performance (**p**) was defined as 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 (expressed in the model by the component **QF** – *Quality Factor*).

The QEST model is basically a multidimensional *structured shell*, which can handle many dimensions, as defined in each context according to management objectives for any specific project: it is therefore referred to as an *open model*. This topology of performance models makes it possible to handle the multiple and distinct viewpoints[5], all of which can exist concurrently in any software project. The basic purpose of the structured shell of the open model is to express performance as a combination of the specific measures (or sets of measures) selected for each of three dimensions, these values being derived initially from both an instrument-based measurement of productivity and a perception-based measurement of quality. A three-dimensional geometrical representation of a *regular tetrahedron* was selected as the basis for the model, and is illustrated in Fig. 3. A description of the geometrical foundations and calculation rules in the QEST/LIME models is documented in [9, 10]. Furthermore:

- the three dimensions (E, S, T in the initial version of the model) in the space correspond to the corners of the pyramid's base, and the convergence of the edges to 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 pyramidal 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, to make it easy to understand. With this 3D representation, it is possible to determine and represent performance considering distinct geometrical concepts (distance, area and volume). From a geometrical viewpoint, QF is expressed as the distance between the two planes, the one described by the (Qe, Qs, Qt) angles and its transposition with (Qe', Qs', Qt'). RP is expressed – for each dimension – by the distance between a corner and its related Q value (i.e. for the Economic viewpoint, RP is equal to the distance from point *E* to point *Qe*). Therefore, geometrically speaking, a full performance level achieved for the Economic viewpoint would be expressed by the distance EP.

In this 3D representation, the ratio between the volume of the lower part of the truncated tetrahedron and the total volume of the tetrahedron represents the normalized performance level of a project being assessed. The geometrical approach permits representation of the measurement of performance in a simple and visual way for immediate impact and optimal understanding. The original selection of the regular tetrahedron was also

[4] See OPP, SP1.3-1

[5] The three initial viewpoints (Economic, Social, Technical) were chosen looking at the core set of project stakeholders (internal-external) having complementary viewpoints to be integrated in performance evaluation.

suggested by the idea that the vertex of the 3D shape represents, from a conceptual viewpoint, the convergence of different viewpoint evaluations into a final, single one. Another important factor to take into account is the use of normalized values in order to give management greater value readability for taking decisions. Other key features of the QEST model are:

- integrated quantitative and qualitative evaluation from three concurrent organizational viewpoints;
- a three-dimensional geometrical construction to yield a single SLC phase value for each project;
- the recommended use of *de facto* and *de jure* standards (such as the ISO/IEC 9126 standard on Software Product Quality Evaluation).

Currently, the original QEST three-dimensional model has been extended to a generic *n* possible dimensions/perspectives of calculation, using the simplex concept. QEST nD [10] can also, therefore, be used as a generic n-dimensional performance model. Another recent development is its use in consolidating Balanced Scorecard (BSC) measurement outcomes [11].

The LIME (**LI**fe cycle **ME**asurement) [12] model extends the QEST model concepts to a dynamic context, such that the model can be applicable to each step of any topology of SLC selected. For illustrative purposes in Fig. 4, the **LIME** model considers only a generic 6-phase waterfall SLC. The dynamic and sequential nature of SLC requires the use of a notation to describe the processes and their flows. From the various notations proposed in the technical literature, the **ETVX** (Entry-Task-Validation-eXit) [13] notation has been chosen. In this notation system, the output of the $(n-1)^{th}$ phase represents the input for the n^{th} one; processing produces the n^{th} output, which will be the input for the $(n+1)^{th}$ phase, and so on. It must be noted that the measurement results ($I_1, ..., I_6, O_1, ..., O_6$) can be added, since they have been normalized within the QEST model to make them easier to understand and represent in a 3D space.

The key features in the LIME model are:

- flexibility of distinct relative contributions from the three dimensions (E, S, T) in each phase;
- flexibility of distinct relative contributions between quantitative and qualitative evaluations in each phase;
- different sources for QF calculation;
- flexibility in selecting suitable measures and ratios for each SLC phase.

A recent refinement of LIME tailored to project risk management, called R-LIME, is documented in [14].

B. QEST/LIME models and Performance Estimation

The *p* values in QEST/LIME models represent the consolidation of final performance project/SLC phase values, and can be used to estimate future performance. In QEST, performance estimation is performed at the whole-project level. In LIME, the iterative definition, collection and analysis of multidimensional measures consolidated into the single "p" performance values at each life cycle phase offer the feedback required to make adjustments to the project processes in a timely fashion, both for the next phase and for designing future improvements to the process of the preceding phase. Therefore, performance at the i^{th} phase is derived from that particular combination from the list of selected ratios, and represents the dependent variable in this series of relationships. The following formulas can better express that concept:

$$p_i = f(x_{1i}, x_{2i}, ..., x_{ni}) \quad (1)$$

while for estimating the performance of the $(i+1)^{th}$ phase:

$$p_{i+1} = f(p_1, p_2, ..., p_i) \quad (2)$$

Once the estimated p_{i+1} values have been derived, it will be possible to use them for cost estimation, as indicated in the CMMI PP, specific practice SP1.4-1.

III. An Example with the QEST Model

Tables I to III illustrate how to calculate and use the *p* values for estimation purposes for a simulation of data for a set of ten software development projects; Table I includes values from a simulation taking into account the ten projects; the *p* performance values are derived with the calculation rules provided in [8] and are listed in the bottom row.

To make it possible to compare them, a common set of indicators (defined in terms of ratios) is selected from the list of measures available in the corporate measurement program and applied to all ten projects. Table II shows the 11 indicators chosen for Project P001. For each of them, some decisions must be taken:

- assign the indicator to one or more viewpoints (E,S,T in the example)[6]. For instance, the Project Delivery Rate (PDR) is assigned only to the Economic dimension, while the Stability Ratio (SR) to both the Social and Technical dimensions;
- calculate the absolute value and then, after choosing the upper and lower boundaries (Rmin and Rmax), calculate the normalized value (labeled "R Value"), in order to allow for comparison among projects. For instance, the absolute value of SR is equal to 0.0063: it is the ratio C90 (5) over FFP(800), taking into account that Rmin=0.001 and Rmax=0.0095; its normalized value will be equal to 0.6176;
- link that indicator to one or more

[6] The selection is specific to an organizational context and to the parties involved in the selection.

processes/practices in the SPI model used in the organization (e.g. CMMI, SPICE) for continuous process improvement purposes. For instance, the SR has been linked to CMMI REQM GP2.8 and SPICE CUS1.1, CUS3.2 and ENG2.3.

Table III illustrates the last procedure before determining the performance values by dimension and, therefore, the overall *p* value for the project. In Table III, the indicators are weighted according to the relevance to each viewpoint (Economic, Social, Technical) and represent the starting point for the *p* performance calculation. For instance, the indicator FFP/ET, called also Time to Delivery Rate, has been simulated to have a different relevance for the Economic viewpoint (30%, the most relevant for managers – e_4 ratio) and for the Technical viewpoint (20%, t_4 ratio), while it is the t_3 ratio (FFP/DD) that has been considered the most relevant for technicians (with a 40% weight). The QEST model considers both a measurement based and a judgment based assessments as inputs to the calculation of the final performance value – the judgment based assessment in the QEST model is referred to as the Quality Factor - QF: in the example (Table III), a ratio of 75:25 was postulated as the contribution of each type of inputs. Therefore the range for the judgment-based qualitative contribution (labeled as MQL – Mean Quality Level) will be in this case between 0% and 25%. This qualitative part of the evaluation can be obtained with a structured technique or, alternatively, estimated by experience/analogy (e.g. Delphi analysis): in the example, the assessment technique described in [15] was used.

A list of QF values has been simulated next for the 11 projects (Table IV) of 75:25, this explains why the sum of weights for each perspective in Table III is equal to 0.75. The sum of the weighted R values per each perspective provides the input for the final *p* value calculation: in the example p_e=0.2648, p_s=0.4744, and p_t=0.2701. These values must be added to the QF factor (for Project P001, QF=0.0080), providing therefore, the updated triple of p values: p_e=0.2728, p_s=0.4824, and p_t=0.2781.

Finally, applying formula (3)

$$p = 1 - \prod_{i=1}^{n} (1 - p_i) \tag{3}$$

it will be possible to calculate the overall project performance value: in the example, the *p* value for Project P001 will be equal to 72.83% using (4).

$$P = 1 - \prod_{1=1}^{n+1} (1 - p_i)_i = 1 - (1 - 0.2728)(1 - 0.4824)(1 - 0.2781) = 1 - (0.7272)(0.5176)(0.7219) = 0.7283 \tag{4}$$

Figure 5 shows three possible relationships: *size* vs. *effort*, plus two others where performance is the dependent variable (*effort* vs. *performance* and *size* vs. *performance*).

In order to appreciate how much each of those relationships can help in terms of effort predictability, Table VI summarizes the main possibilities, the best highlighted in bold in each regression type chosen.

The strongest relationship in all cases is the one between *size* and *effort (*the higher the order, the higher the R^2 value), followed by the one between *size* and *performance*. The third (*effort* vs. *performance*) has good values only in the polynomial case from the fifth order on. Table VII aggregates the R^2 values obtained by type of relationship used, sorted by order: the usual indication being that the higher the order, the better the correlation between variables, whatever the relationship considered.

The next example (Table V) is a new development project with an estimated size of 2500 FP. Applying the linear regression equation for *size* vs. *performance*, *p** (the estimated *p* value) is equal to 0.8646. Using the *size* vs. *effort* linear regression equation with the same size as input, it provides an estimated effort of 1305 man/days; using this number of man/days as input in the *effort* vs. *performance* relationship, the *p** would be equal to 0.8557. Supposing the other measures to be very similar to those for project P010 (since their size is very similar), as shown in Table V, performing the full QEST calculation, the *p* value will be equal to 0.9220. Considering the two estimated *p* values previously calculated, it is possible to verify that the lower MRE%[7] is the one obtained using the *size* vs. *performance* equation.

Using the same QEST framework as before in terms of list of ratios, related weights and thresholds, it is possible, starting from the *p** value, size and effort estimated, to verify the proper fit of the other measures and balance them in order to optimize the amount of resources used and the final project cost, taking care of those concurrent constraints. In such a way, the use of initial project effort estimation through its size in combination with performance estimation can provide Project Managers with an additional checking tool for controlling project costs.

IV. CONCLUSIONS

Estimating project costs is one of the key challenges for a project manager. The problem is to detect the proper variables for the project that permit minimization of the project's Mean Relative Error (MRE), in terms of both

[7] MRE (Mean Relative Error) is defined as the percentage difference between the predicted effort (Epred) and the actual one (Eact) in absolute values for a certain event/issue. The formula is: $MRE = \frac{|Eact - Epred|}{Eact}$

time and costs. Cost estimation is based mainly on effort; however, performance also plays an important role, even when not formally considered in calculating the effort estimates. It remains important, however, to understand which causal relationships link cost, effort and performance.

Two more elements must be considered: first, estimation performed in the planning phase, which requires the use of early sizing methods to arrive at an approximate software size; second, the integration of multiperspective viewpoints (those provided by the stakeholders in the project).

QEST/LIME represents a family of multidimensional software performance models, which – with a geometrical approach – can help in calculating project performance, providing the needed input for a more comprehensive cost estimation.

The purpose of the performance model, at this point in time, is to be descriptive and to be used for monitoring purposes. Later on, when data about past performance will have been collected, then such historical data can be used as a contribution in building estimation models. Then only comparisons can be made with other estimation models (e.g. COCOMO) but – of course – only for the subset of performance related to the "effort" variable.

REFERENCES

[1] C. Jones, "Software Project Management Practices: Failure Versus Success," *Crosstalk*, Vol.17, no.19, Oct. 2004, pp.5-9. Available: http://www.stsc.hill.af.mil/crosstalk/2004/10/0410Jones.pdf

[2] CMU/SEI, "CMMI[SM] for SE/SW/IPPD/SS Ver. 1.1, Continuous Representation," Technical Report CMU/SEI-2002-TR-01, Carnegie Mellon University – Software Engineering Institute, Pittsburgh, PA: Mar. 2002. Available: http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr001.pdf

[3] ISO/IEC, "TR 15504-5. Software Process Assessment – Part 5: An Assessment Model and indicator guidance," version 3.3, Technical Report, International Organization for Standardization, JTC1/SC7/WG10, 1998

[4] IEEE, Standard Glossary of Software Engineering Terminology, IEEE Std 610.12, 1990 (Rev: 2002)

[5] DACS, *A History of Software Measurement at Rome Laboratory: Complexity Measures*, Technical Report, Data & Analysis Center for Software, 25 Nov. 1996, URL: http://www.dacs.dtic.mil/techs/history/His.RL.2.1.4.html

[6] R. Al-Ekram, *Software Reliability Growth Modeling and Prediction*, Dept. of Electrical Computer Engineering, Univ. of Waterloo, 5 March 2002; URL: http://www.swen.uwaterloo.ca/~rekram/reports/software_reliability_growth_modeling_and_prediction.pdf

[7] S.D. Sink, "Productivity Management: Planning, Measurement and Evaluation, Control and Improvement", John Wiley & Sons, 1985

[8] "QEST/LIME webpage," SEMQ website, 1 June 2005. Available: http://www.geocities.com/lbu_measure/qestlime/qestlime.htm

[9] L. Buglione and A. Abran, "Multidimensional Software Performance Measurement Models: A Tetrahedron-based Design," Software Measurement: Current Trends in Research and Practice, R.Dumke/A.Abran (eds.), Deutscher Universitats Verlag GmbH, ISBN 3-8244-6876-X, 1999, pp.93-107

[10] L. Buglione and A. Abran, "QEST nD: n-dimensional extension and generalisation of a Software Performance Measurement Model." *International Journal of Advances in Engineering Software*, Elsevier Science Publisher, Vol. 33, no. 1, Jan. 2002, pp.1-7

[11] A. Abran and L. Buglione, "A Multidimensional Performance Model for Consolidating Balanced Scorecards." International Journal of Advances in Engineering Software. Elsevier Science Publisher, Vol. 34, no.6, June 2003, pp.339-349

[12] L. Buglione and A. Abran, "LIME: A Three-Dimensional Software Performance Measurement Model for Project Management." Proceedings of the 2nd World Conference on Software Quality (2WCSQ), Sept. 25-29, 2000, Yokohama, Tokyo (Japan)

[13] R.A. Radice and R.W. Phillips, "Software Engineering: An Industrial Approach," Prentice-Hall, ISBN 0-13-823220-2, 1988

[14] A. Abran, L. Buglione and D. Girard, "R-LIME: Improving the Risk dimension in the LIME model." 3rd World Conference on Software Quality (3WCSQ), Munich (Germany), 25-29 Sept. 2005

[15] L.Buglione and A.Abran, "A Quality Factor for Software", 3rd International Multidisciplinary Congress in Quality and Reliability (QUALITA99), ENSAM-RUFEREQ, Paris, France, March 25-26, 1999,ISBN 2-900-781-43-4, pp. 335-344

Dr. **Luigi Buglione**, ***Ph.D.***, is an Associate Professor at the École de Technologie Supérieure (ETS) – Université du Québec, and currently works as a software quality engineer at Atos Origin. Previously, he worked at the European Software Institute in Spain. His main research interests are software measurement, software quality and process improvement and multi-dimensional performance models. Dr. Buglione is a Steering Committee member of the Italian Function Point Users Group – Italian Software Metrics Association (GUFPI-ISMA) and Coordinator of the Software Measurement Committee. A technical reviewer for international journals on software engineering, he developed and has been involved with ESPRIT and Basque Government projects on metrics programs, EFQM models, the Balanced IT Scorecard and QFD for software. He is the author of the book (in Italian) "Misurare il software" (Measuring Software). He is also a member of the IEEE Computer Society. He can be reached at luigi.buglione@computer.org

Dr. **Alain Abran**, ***Ph.D.***, is a Professor and the Director of the Software Engineering Research Laboratory at the École de Technologie Supérieure (ETS) – Université du Québec. He is currently Co-executive Editor of the Guide to the Software Engineering Body of Knowledge project. He is also actively involved in international software engineering standards and is Co-chair of the Common Software Metrics International Consortium (COSMIC). Dr. Abran has more than 20 years of industry experience in information systems development and software engineering. The maintenance measurement program he developed and implemented at Montreal Trust, Canada, received one of the 1993 Best of the Best awards from the Quality Assurance Institute. He can be reached at alain.abran@ele.etsmtl.ca

LIST OF FIGURES

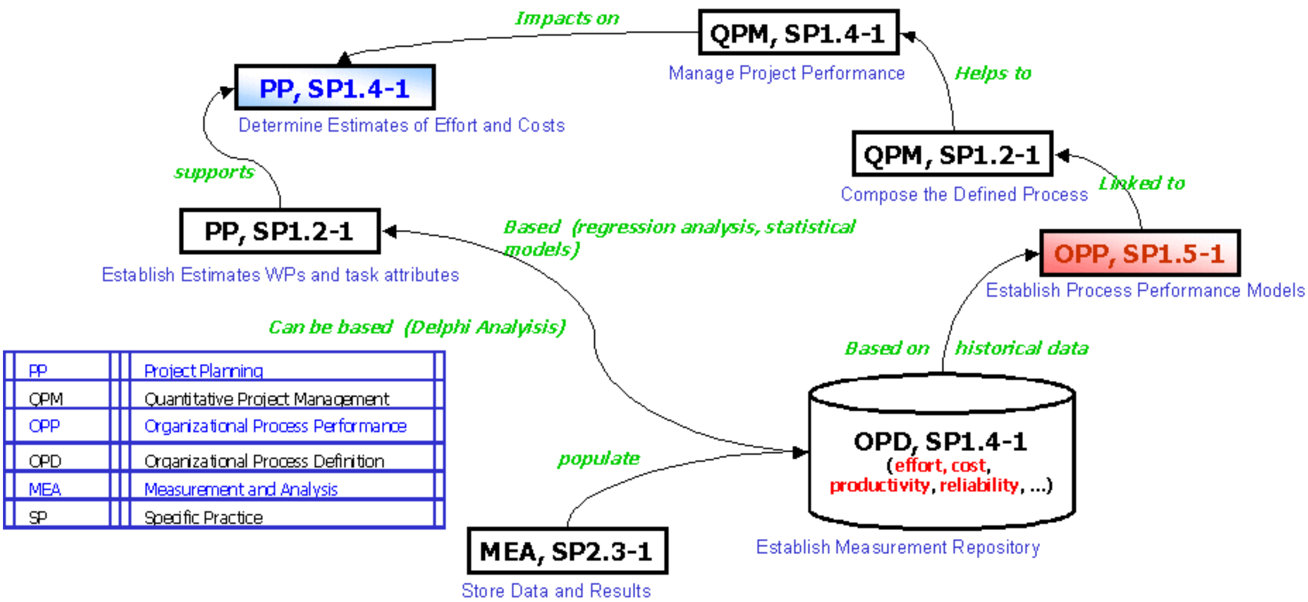

Fig. 1. Estimation and related inputs from a CMMI perspective.

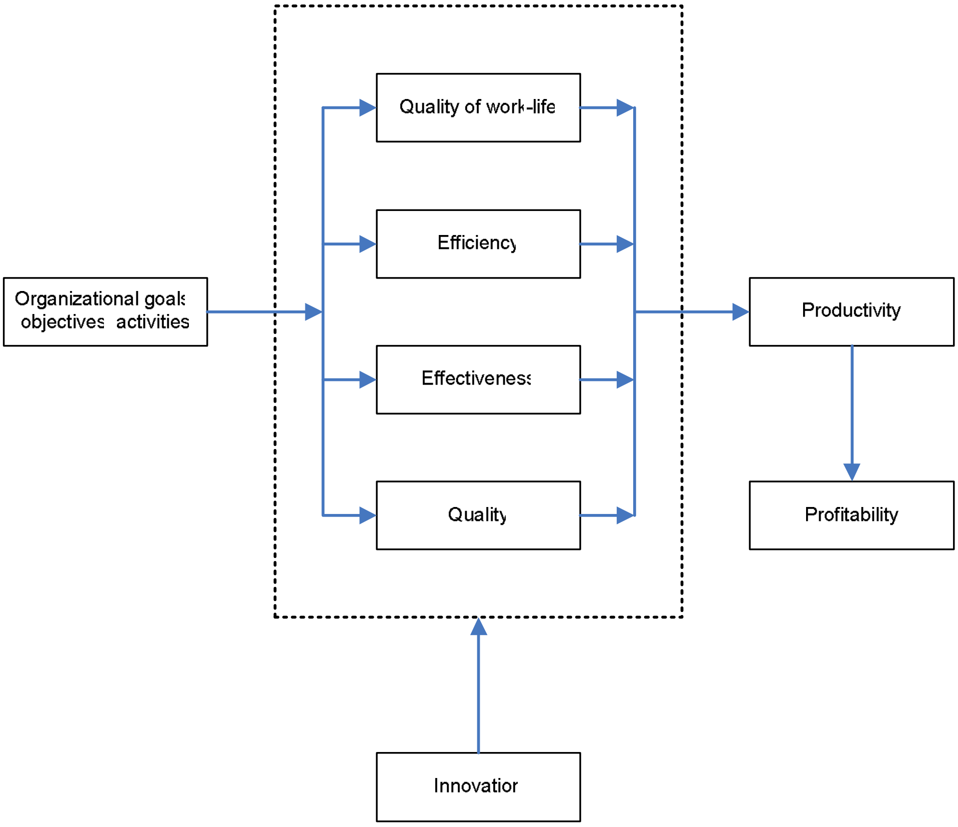

Fig. 2. Relationship between Performance and Productivity [6]

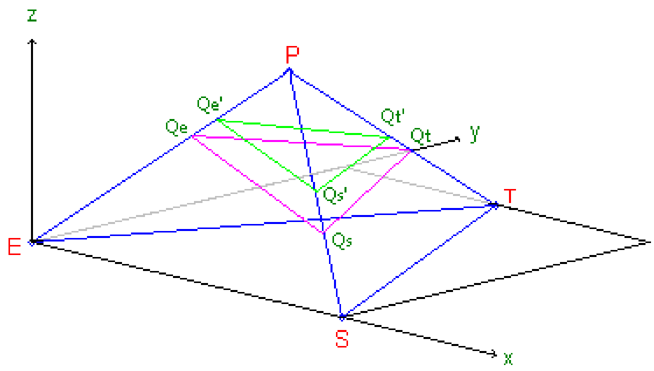

Fig. 3. The QEST model.

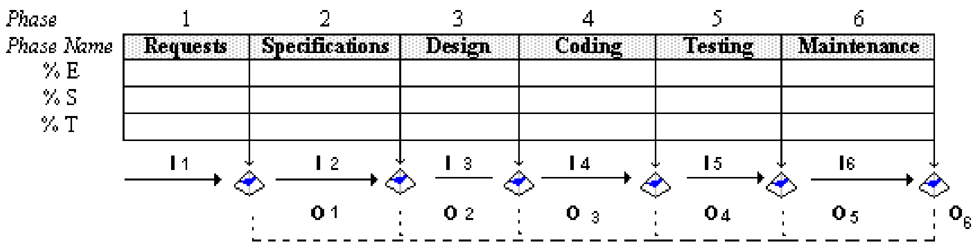

Fig. 4. The LIME model.

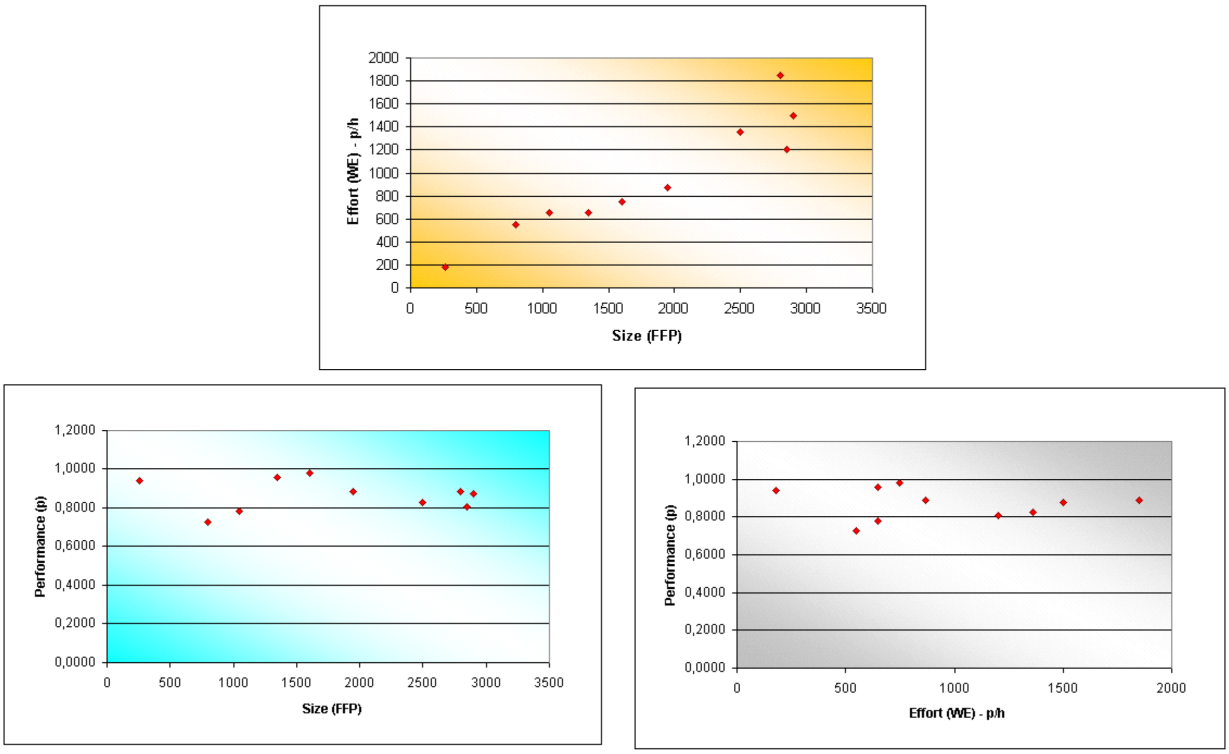

Fig. 5. Scattered point diagram – three possible correlations

LIST OF TABLES

TABLE I

A SIMULATION OF 10 PROJECTS PERFORMANCE

# m	Project Measures	Definition	Prj value P001	P002	P003	P004	P005	P006	P007	P008	P009	P010
m1	**FFP** - Full Function Points	according to COSMIC-FFP MM2.2	800	260	1050	2900	2800	1350	1950	1600	2500	2850
m2	**WE** - Work Effort (total)	in person/hours (p/h)	550	180	650	1500	1850	650	870	750	1360	1200
m3	**QRC** - Quantity of Reuse Code	in FFPs	100	0	250	300	300	450	200	200	240	140
m4	**QTC** - Quantity of Total Code	in LOC (Java)	22400	7280	29400	81200	78400	37800	54600	44800	70000	79800
m5	**WER** - Work effort for Reuse	in person/hours (p/h)	75	0	55	12,5	90	40	65	55	50	180
m6	**TTT** - Total Training Time	in person/hours (p/h)	30	20	40	60	50	0	16	20	40	28
m7	**EDT** - Esteemed Developed Time	in person/months (p/m)	0,6875	0,225	0,8125	1,875	2,3125	0,8125	1,0875	0,9375	1,7	1,5
m8	**ET** - Elapsed Time (actual)	in person/months (p/m)	3,4375	1,125	4,0625	9,375	11,5625	4,0625	5,4375	4,6875	8,5	7,5
m9	**HDC** - calls to Help Desk	# of calls per unit of time	37	15	40	50	50	42	45	35	48	47
m10	**RTT** - Requested Training Time	in person/hours (p/h)	40	30	48	80	50	20	80	80	60	120
m11	**DD** - Defect Density	# of defects per software component	9	5	18	15	12	16	11	10	14	12
m12	**TE** - Test effort	in person/hours (p/h)	30	23	55	200	155	50	160	100	120	235
m13	**IE** - Inspection effort	in person/hours (p/h)	18	4	15	60	32	16	36	28	72	42
m14	**C90** - Changes during first 90 days after implementation	# of changes	5	10	5	15	12	5	13	8	12	15
	p	QEST performance value	**0,7283**	**0,9400**	**0,7811**	**0,8748**	**0,8867**	**0,9577**	**0,8865**	**0,9805**	**0,8256**	**0,8054**

TABLE II

DIMENSIONS, BOUNDARIES, NORMALIZED VALUES AND SPI RELATIONSHIPS[8].

P001	RATIO	NAME	DIM (E, S,T)	Rmin	Rmax	Abs Value	R Value	CMMI v1.1 PAs Involved	SPICE v3.3 PAs Involved
m1/m2	FFP/WE	Project Delivery Rate (PDR)	E	2,0000	6,6670	1,4545	-0,1169	PP SP1.4, PMC SP1.1	MAN.1.4, MAN.1.7, MAN1.10, ORG.5.5
m3/m1	QRC / FFP	% of reused code	E, T	0,0000	0,5000	0,1250	0,2500	PMC SP1.6, TS 2.1	MAN.1.2, ORG.5.8, ORG.5.9
m5/m2	WE / WER	% of WE spent for reuse	E	0,1000	0,5000	0,1364	0,0909	PMC SP1.6	MAN.1.7
m14/m1	C90 / FFP	Stabiility Ratio (SR)	S, T	0,0010	0,0095	0,0063	0,6176	REQM GP2.8	CUS.1.1, CUS.3.2, ENG.2.3
m1/m11	FFP / DD	inverse of Defect Ratio (IDR)	T	1,0000	1000,0000	88,8889	0,0880	VER GP2.8	MAN.2, ORG.1.3
m6/m10	TTT / RTT	Training Time Coverage (TTC)	S	0,0000	1,0000	0,7500	0,7500	OT GP2.8	CUS.5.1, ORG.4.1, ORG.4.3
m1/m8	FFP / ET	Duration Delivery Rate (DDR)	E, T	142,0000	250,0000	232,7273	0,8401	PP SP1.4, PMC SP1.1	MAN.1.10, ORG.5.5
q	Rxy (experience / usability)		S	0,0000	1,0000	0,4000	0,4000	---	CUS.5.4
q	Rxy (education / usability)		S	0,0000	1,0000	0,4300	0,4300	---	CUS.5.4
q	Rxy (age / usability)		S	0,0000	1,0000	0,3000	0,3000	---	CUS.5.4
q	Rxy (ideal / real evaluation)		S	0,0000	1,0000	0,6000	0,6000	---	CUS.5.4

Legend: q=derived from questionnaire

TABLE III

SELECTED RATIOS WEIGHTED BY DIMENSIONS (E, S, T) WITH THRESHOLDS FOR PROJECT P001

Quantity Level: **0,75**

Quality Level (MQL): **0,25**

P001		Weight	R Values	Final Values	Acceptability Threshold	Delta Values
E				**0,2648**	**0,3300**	-0,0652
e1	FFP / WE	0,25	-0,1169	-0,0292	0,1500	-0,1792
e2	QRC / FFP	0,15	0,2500	0,0375	0,0300	0,0075
e3	WE / WER	0,05	0,0909	0,0045	0,0240	-0,0195
e4	FFP/ ET	0,30	0,8401	0,2520	0,1600	0,0920
		0,75				
S				**0,4744**	**0,4200**	0,0544
s1	Rxy (experience / usability)	0,05	0,4000	0,0200	0,0180	0,0020
s2	Rxy (education / usability)	0,02	0,4300	0,0086	0,0085	0,0001
s3	Rxy (age / usability)	0,03	0,3000	0,0090	0,0090	0,0000
s4	Rxy (ideal / real evaluation)	0,25	0,6000	0,1500	0,1750	-0,0250
s5	TTT / RTT	0,30	0,7500	0,2250	0,2300	-0,0050
s6	C90 / FFP	0,10	0,6176	0,0618	0,0400	0,0218
		0,75				
T				**0,2701**	**0,2500**	0,0201
t1	QRC / FFP	0,07	0,2500	0,0175	0,0135	0,0040
t2	C90 / FFP	0,08	0,6176	0,0494	0,0370	0,0124
t3	FFP / DD	0,40	0,0880	0,0352	0,0510	-0,0158
t4	FFP/ ET	0,20	0,8401	0,1680	0,1000	0,0680
		0,75				

[8] The values for the first four « Social » measures of S are derived from a questionnaire filled by representatives from all the viewpoints considered.

TABLE IV

SIMULATED QF AND CORRESPONDING P VALUES FOR THE 11 PROJECTS

					P_e	P_s	P_t				
	e	s	t	QF	(e+QF)	(s+QF)	(t+QF)	1-p_e	1-p_s	1-p_t	*p*
P001	0,264847	0,474365	0,270116	0,0080	0,2728	0,4824	0,2781	0,7272	0,5176	0,7219	**0,7283**
P002	0,205271	0,825524	0,53802	0,0070	0,2123	0,8325	0,5450	0,7877	0,1675	0,4550	**0,9400**
P003	0,372407	0,481133	0,307365	0,0060	0,3784	0,4871	0,3134	0,6216	0,5129	0,6866	**0,7811**
P004	0,48082	0,428287	0,559978	0,0070	0,4878	0,4353	0,5670	0,5122	0,5647	0,4330	**0,8748**
P005	0,277892	0,52918	0,324436	0,1256	0,4035	0,6548	0,4500	0,5965	0,3452	0,5500	**0,8867**
P006	0,627945	0,254408	0,457918	0,1725	0,8004	0,4269	0,6304	0,1996	0,5731	0,3696	**0,9577**
P007	0,642263	0,328942	0,505639	0,0070	0,6493	0,3359	0,5126	0,3507	0,6641	0,4874	**0,8865**
P008	0,595013	0,373934	0,487947	0,2300	0,8250	0,6039	0,7179	0,1750	0,3961	0,2821	**0,9805**
P009	0,434779	0,478331	0,402004	0,0020	0,4368	0,4803	0,4040	0,5632	0,5197	0,5960	**0,8256**
P010	0,463624	0,35768	0,423395	0,0040	0,4676	0,3617	0,4274	0,5324	0,6383	0,5726	**0,8054**
P011	0,470402	0,729584	0,439165	0,0040	0,4744	0,7336	0,4432	0,5256	0,2664	0,5568	**0,9220**

TABLE V

PROJECT P011 MEASURES AND MRE% FOR BOTH RELATIONSHIPS WITH PERFORMANCE

# m	Project Measures	Definition	P011
m1	**FFP** - Full Function Points	according to COSMIC-FFP MM2.2	2500
m2	**WE** - Work Effort (total)	in person/hours (p/h)	1305,3
m3	**QRC** - Quantity of Reuse Code	in FFPs	120
m4	**QTC** - Quantity of Total Code	in LOC (Java)	70000
m5	**WER** - Work effort for Reuse	in person/hours (p/h)	170
m6	**TTT** - Total Training Time	in person/hours (p/h)	36
m7	**EDT** - Esteemed Developed Time	in person/months (p/m)	1,6317
m8	**ET** - Elapsed Time (actual)	in person/months (p/m)	8,1583
m9	**HDC** - calls to Help Desk	# of calls per unit of time	40
m10	**RTT** - Requested Training Time	in person/hours (p/h)	48
m11	**DD** - Defect Density	# of defects per software component	10
m12	**TE** - Test effort	in person/hours (p/h)	120
m13	**IE** - Inspection effort	in person/hours (p/h)	52
m14	**C90** - Changes during first 90 days after implementation	# of changes	10

p	QEST performance value	**0,9220**	
	p estimated	**0,8646**	**0,8557**
	MRE%	6,23%	7,19%

TABLE VI

REGRESSION TYPES BY ORDER AND R^2

Regression type	Order	Relationship	R^2	Rank	Equation
Linear	1	*Size vs Eff*	**0.8733**	☺	**$y=0.5034x+46.833$**
		Size vs Perf	0.0026	☹	$y=-4E-06x+0.8746$
		Eff vs Perf	0.0098	☹	$y=-2E-05x+0.8818$
Logarithmic	1	*Size vs Eff*	**0.7492**	☺	**$y=579.85Ln(x)-3279.9$**
		Size vs Perf	0.0083	☹	$y=-0.0098Ln(x)+0.9379$
		Eff vs Perf	0.0333	😐	$y=-0.0222Ln(x)+1.0156$
Polynomial	2	*Size vs Eff*	**0.8784**	☺	**$y=5E-05x^2+0.3213x+159,87$**
		Size vs Perf	0.0137	☹	$y=-1E-08^2++4E-05x+0.8481$
		Eff vs Perf	0.0582	☹	$y=7E-08x^2-0.0002x+0.9405$
Polynomial	3	*Size vs Eff*	**0.8809**	☺	**$y=-6E-08x^3-0.0002x^2+0.667x+57.305$**
		Size vs Perf	0.2187	😐	$y=-8E-11x^3+4E-07x^2-0.0005x+0.9985$
		Eff vs Perf	0.0583	☹	$y=-5E-12x^3+8E-08x^2-0.0002x+0.9429$
Polynomial	4	*Size vs Eff*	**0.9062**	☺	**$y=-3E-10x^4+2E-06x^3-0.004x^2+3.5562x-513.2$**
		Size vs Perf	0.7921	😐	$y=2E-13x^4-1E-09x^3+3E-06x^2-0.0027x+1.4329$
		Eff vs Perf	0.1769	☹	$y=8E-13x^4-3E-09x^3+4E-06x^2-0.0022x+1.199$
Polynomial	5	*Size vs Eff*	**0.9149**	☺	**$y=-3E-13x^5+2E-09x^4-6E-06x^3+0.0061x^2-2.0197x+386.39$**
		Size vs Perf	0.8683	😐	$y=-2E-16x^5+1E-12x^4-5E-09x^3+8E-06x^2-0.0053x+1.8528$
		Eff vs Perf	0.6056	☹	$y=-5E-15x^5+2E-11x^4-5E-08x^3+4E-05x^2-0.0155x+2.6399$

TABLE VII

RELATIONSHIP TYPES SORTED BY INCREASING ORDER OF R^2

Relationship	Regression type	Order	R^2	Trend	Equation
Size vs Eff	Linear	1	0.8733	---	$y=0.5034x+46.833$
	Logarithmic	1	0.7492	↓	$y=579.85Ln(x)-3279.9$
	Polynomial	2	0.8784	↑	$y=5E-05x^2+0.3213x+159,87$
	Polynomial	3	0.8809	↑	$y=-6E-08x^3-0.0002x^2+0.667x+57.305$
	Polynomial	4	0.9062	↑	$y=-3E-10x^4+2E-06x^3-0.004x^2+3.5562x-513.2$
	Polynomial	5	**0.9149**	↑	$y=-3E-13x^5+2E-09x^4-6E-06x^3+0.0061x^2-2.0197x+386.39$
Size vs Perf	Linear	1	0.0026	---	$y=-4E-06x+0.8746$
	Logarithmic	1	0.0083	↑	$y=-0.0098Ln(x)+0.9379$
	Polynomial	2	0.0137	↑	$y=-1E-08^2++4E-05x+0.8481$
	Polynomial	3	0.2187	↑	$y=-8E-11x^3+4E-07x^2-0.0005x+0.9985$
	Polynomial	4	0.7921	↑	$y=2E-13x^4-1E-09x^3+3E-06x^2-0.0027x+1.4329$
	Polynomial	5	**0.8683**	↑	$y=-2E-16x^5+1E-12x^4-5E-09x^3+8E-06x^2-0.0053x+1.8528$
Eff vs Perf	Linear	1	0.0098	---	$y=-2E-05x+0.8818$
	Logarithmic	1	0.0333	↑	$y=-0.0222Ln(x)+1.0156$
	Polynomial	2	0.0582	↑	$y=7E-08x^2-0.0002x+0.9405$
	Polynomial	3	0.0583	↑	$y=-5E-12x^3+8E-08x^2-0.0002x+0.9429$
	Polynomial	4	0.1769	↑	$y=8E-13x^4-3E-09x^3+4E-06x^2-0.0022x+1.199$
	Polynomial	5	**0.6056**	↑	$y=-5E-15x^5+2E-11x^4-5E-08x^3+4E-05x^2-0.0155x+2.6399$