HOW FUNCTIONAL SIZE MEASUREMENT SUPPORTS THE BALANCED SCORECARD FRAMEWORK FOR ICT

LUIGI BUGLIONE | **ALAIN ABRAN**
Software Engineering Management Research Laboratory
Université du Québec à Montréal
C.P. 8888, Succ. Centre-Ville
Montréal, Québec, Canada
E-mail: luigi.buglione@computer.org
Tel: (39) 338.95.46.917
Fax: (39) 06-233.208.366
E-mail: abran.alain@uqam.ca
Tel: +1 (514) 987-3000 (8900)
Fax: +1 (514) 987-8477

ROBERTO MELI
D.P.O. S.rl.
Data Processing Organization
Via Flaminia, 217
I-00196 Rome, Italy
E-mail: roberto.meli@dpo.it
Tel: (39) 06-322.68.87
Fax: (39) 06-323.36.28

Abstract

The Balanced Scorecard (BSC) represents one of the performance management frameworks adopted with great success in business circles in recent years. One of its most valuable strengths is its linkage of the strategic and operational levels, through a quantitative and qualitative management using a series of indicators from four different perspectives: Financial, Customer, Internal Process, Learning & Growth. The success of this framework in the business world has led to some tailored extensions in the ICT world, with a few examples developed in the second half of the '90s. A key issue that needs to be addressed in the design and implementation of a BSC for ICT companies is measurement of the software itself. To build a BSC, once the overall strategic direction has been identified, Goals, Drivers and Indicators (GDI elements) must be selected for each perspective. Even though significant attention has already been paid to the first two elements (Goals and Drivers), the last (Indicators) has been largely neglected. To address this measurement issue in the ICT field, we propose that Functional Size Measurement (FSM) be used as a key measure to normalise other measurement results across reference values. In summary, this paper illustrates how the use of Functional Size Measurement can strengthen an ICT BSC, from the operational point of view of measurement.

Keywords – ICT Balanced Scorecard, Functional Size Measurement, Function Point Analysis, Metrics, Measurement Plan, Strategy.

Conference Topics – ICT Balanced Scorecard, Functional Size Measurement, Function Point Analysis, Metrics, Measurement Plan, Strategy.

1. Introduction

Performance Measurement occupies a central role in the management practices of organisations throughout the world and has been central to the business mode of operation over recent centuries. In the ICT field, quality and process improvement models such as ISO 9001and Malcolm Baldridge, and EFQM and SPI models such as CMMI and ISO/IEC 15504 (SPICE), have become better known, but are sometimes used as *blind models*, in the sense that they are restricted to a long list of requirements to be met, rather than as part of a strategy to support an organisational vision.

Thus, performance measurement models such as the Balanced Scorecard (BSC) owe their success to a focus on establishing improvement programs within the context of the overall business strategy, while an SPI initiative represents only a part of that strategy.

During the last decade, several attempts have been made to tailor the BSC for the ICT world and the resulting tools have been proposed to the Software Engineering community; however, these tools focused more on the structure of the framework (GD elements – Goals and Drivers – and perspectives used) than on the framework's concrete "content", that is, on the measures and indicators required to fill out the framework.

On the software measures side, a significant development in the last 25 years has been, without doubt, the measurement of software functionality, initially defined 20 years ago by Albrecht, and now being standardised at the ISO level. In this context, it is important to explore whether or not Functional Size Measurement could help operationalise a BSC in ICT organisations.
In section 2, the BSC is introduced, as well as the tailoring performed for the ICT world. In section 3, the highlights of Functional Size Measurement (FSM) are presented, from the initial Albrecht study to current ISO work in progress. The way in which FSM can define specific measures for the operationalisation of an ICT BSC is explained in section 4, followed by some concluding remarks in section 5.

2. The Balanced Scorecard (BSC)

2.1. The original method

Evaluation of the IT function still remains a challenge: well-known financial measures such as the "return on investment" (ROI), "internal rate of return" (IRR), "net present value" (NPV) and the "payback" time (PB) have been demonstrated to be inadequate to explain IT investment decisions or to assess them [BRYN93]. In order to assess IT investments, it is crucial to understand how organisational and strategic goals are achieved, and how IT investments contribute to these goals.
Since the financial dimension proved to be insufficient and hid the relations (e.g. cause-and-effect) among processes, an improvement step was suggested which would introduce additional dimensions (or perspectives) of analysis. In 1993, for example, Robert S. Kaplan of the Harvard School of Business and consultant David Norton developed the Balanced Scorecard (BSC), an evolution of the concepts included in the *Tableau de Bord* which emerged in France at the turn of the 20th century [EPST97]. The aim of the *Tableau* had been to translate each company's unitary vision and mission into a set of objectives, through the identification of Key Success Factors and Key Performance Indicators.
Kaplan and Norton defined the Balanced Scorecard (BSC) [KAPL96][1] as a multi-dimensional framework for describing, implementing and managing strategy at all levels of an enterprise and by linking objectives, initiatives and measures to an organisation's strategy. The scorecard then provides an enterprise view of an organisation's overall performance: it integrates the financial measures with other key performance indicators around customer perspectives and internal business processes, and around organisational growth, learning and innovation. It must be noted that the BSC is not a static list of measures, but rather a framework for implementing and aligning complex programs of change, and, indeed, for managing strategy-focused organisations. In summary, a scorecard is to be used to facilitate the translation of strategy into action.
The BSC provides a framework for studying causal links based on internal performance measurement through a set of goals, drivers and indicators (lag and lead types) grouped into four different perspectives:

- ✓ Financial: typically relates to profitability – measured by ROI, ROCE and EVA, for instance;
- ✓ Customer: includes several core or generic measures of the successful outcomes of company strategies - for instance, customer satisfaction, customer retention, and market and account share in targeted segments;
- ✓ Internal processes: focuses on the internal processes that will have the greater impact on customer satisfaction and on achieving the organisation's financial objectives;
- ✓ Learning and growth: identifies the infrastructure the organisation has to build in order to create long-term growth and improvement through people, systems and organisational procedures.

A good BSC should tell the story of the organisation's strategy[2]. Three criteria help in determining whether or not this objective has been achieved:

- **Cause-and-effect relationship**: *every measure selected should be part of a cause-and-effect relationship (causal relationship chain) which represents the strategy;*
- **Performance Drivers**: *the drivers of performance (lead indicators) tend to be unique since*

1 [KAPL96] is an extended and more comprehensive reference on the BSC. The first two publications on the BSC by Kaplan and Norton are [KAPL92] [KAPL93].
2 http://www.balancedscorecard.com

they reflect what is different about the strategy of a company. They should be properly mixed with lag indicators;

Linked to financial indicators: *while there is a proliferation of new strategic goals such as quality, customer satisfaction and innovation, these goals must also translate into measures which are ultimately linked to financial measures.*

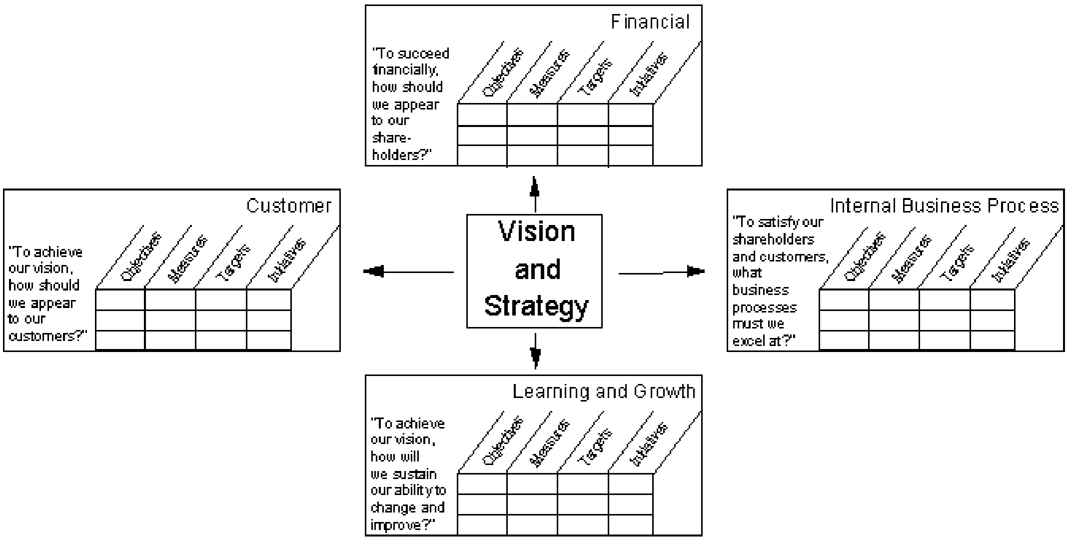

Figure 1 – *Balanced Scorecard: original perspectives*

2.2. ICT-tailored BSC

Some software organisations have recently tried to use the BSC for achieving better results, since Software Process Improvement (SPI) and SPI models are not goals in themselves, but just elements in the overall strategy of the company. Thus, there have been a few attempts to build an ICT scorecard for Software Intensive Organisations (SIOs[3]), such as:

- the Balanced IT Scorecard (BITS) [IBAÑ98] [REO99a] [REO99b], proposed by the European Software Institute (ESI) - it provides a new version of the four original perspectives (financial, customer, internal process, infrastructure and innovation) and adds a fifth, the People perspective[4], and
- the BSC of Advanced Information Services Inc. [FERG99] - it considers the "employee" element as a distinct perspective, thereby expanding the analysis to five perspectives (financial, customer, employee, internal business process, learning and growth).

Particular attention is paid here to the first - BITS. The ESI has adapted and extended the principles of the Balanced Scorecard to provide a well-defined approach to quantitatively manage SPI programmes in SIOs and to validate their effect on organisational business goals. More specifically, BITS supports:

(a) the identification and prioritisation of software process improvement needs derived from the organisation's business goals;
(b) the agreement and communication of the business strategy among the SIO's manager, sponsor and software engineers, thereby strengthening the required commitment from all parties;
(c) the identification of the critical set of factors affecting the achievement of the SIO's business goals;
(d) the selection of the minimum set of indicators to monitor the performance of the software processes.

[3] SIOs are organisations whose main objective is software development and sales, departments or organisations which develop software as an integral part of their end-products, and organisations which develop software for internal use to achieve better business results or whose software department can be described as an independent organisational unit (European Software Institute definition, 1997).

[4] In fact, personnel are the "prime material" of software development. The knowledge and experience of people represents a most important asset and should not be relegated to the infrastructure level. Competence, satisfaction and retention are the three drivers to reaching higher productivity levels [REO00].

Therefore, the five distinct perspectives, derived from the original scorecard, are:

- **Financial Perspective**: How do our software processes and SPIs add value to the company?
- **Customer Perspective**: How do we know that our customers (internal and external) are delighted with our product?
- **Process Perspective**: Are our software development processes performing at sufficiently high levels to meet customer expectations?
- **People Perspective**: Do our people have the necessary skills to perform their jobs and are they happy doing so?
- **Infrastructure & Innovation Perspective**: Are process improvement, technology and organisational infrastructure issues being addressed with a view to implementing a sustainable improvement program?

A generic ICT Balanced Scorecard, without a predefined number of perspectives, will, however, be taken here as the basis of discussion from a measurement perspective.

2.3. ICT BSC: an operationalisation issue

A BSC is composed of basic elements arranged in the so-called GDI (Goal-Driver-Indicator) structure. Up to now, attention had been focused on the first two (Goals & Drivers) considered as structural elements, because of their relevance in the BSC architecture, the linkage among goals and the choice of the right driver for those goals. However, the third element (I – Indicator), defined as a content element, which much be specific to the selected drivers, has often been neglected, thereby making it difficult for organisations to operationalise the BSC in specific environments, such as ICT contexts. The objective of the paper is to demonstrate that the usage of FSMs is not and has not to be limited to the sizing, but can have a wider impact also on organisational issues, not only technical, properly fitting with the multi-dimensional philosophy of the BSC. But, at the same time, as shown in Section 4.3, previous studies on using FPA-based measures have not yet been sufficient to support the robustness required by BSC framework. This paper therefore attempts to identify which kind of contribution those software measures can provide in implementing an organisational framework such as an ICT BSC and which perspectives can be "touched" through the analysis of the main informative sources.

3. Functional Size Measurement (FSM)

3.1. Evolution of FSM

The so-called "*productivity paradox*" cited by Jones [JONE96a] serves as a starting point for the analysis of the usefulness of functional size measures. Albrecht [ALBR79] proposed the first software measure with no technology binds and simply based on what the system user could see from the outside. In this way, the number of implemented functionalities took into account 5 element types, weighted and corrected with 10 adjustment factors. The basic technique has since been refined, first by Albrecht himself [ALBR84], then by various researchers, and, more recently, by an international group of experts in FSM [ABRA99, ABRA01]. An outline of the evolution of FSM appears in Figure 2, and includes the second generation of FSM developed by the Common Software Measurement International Consortium - COSMIC.
In addition, the Software Engineering community is working through the ISO process in ISO JTC1/SC7 (Working Group - WG12) to establish generic guidelines both for FSM and for approving specific methods [ISO00]. The ISO suite of documents on this topic is being referred as the 14143 Series.

ISO 14143 is a 5-part International Standard produced under the direction of JTC1/SC7/WG12 of the International Standards Organisation which was founded in 1994. The aim of this standard is to provide an internationally accepted set of meta-standards to describe the concept and practices of FSM methods. It is important to stress that this first set of standards does not define a specific FSM method, but rather defines the characteristics any given method must have before it can be properly referred to as a functional size measure.

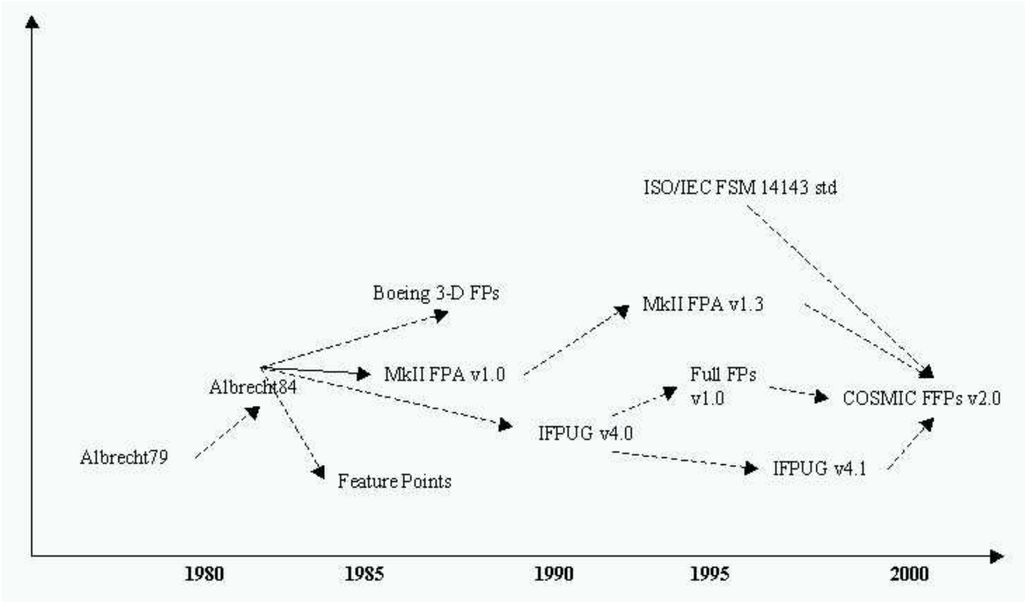

Fig. 2 – *Evolution of FSM techniques*

3.2. FSM and Technical Size Measurement (TSM)

The field of software measurement has frequently been an arena where supporters of Functional Measurement Methods - Function Points, mainly – were opposed to supporters of Technical Measurement Methods – like the ones based on LOC, number of programs, modules, reports, screens, classes, objects, components, boxes, widgets etc.
Measuring a piece of software from a pure functional perspective, however, fulfils different purposes than measuring the same software from a technical perspective. We should increasingly get used, in the future, to the fact that a software application has “many sizes” to be measured and that different but complementary types of measures are needed. The two most important “sizes” are “logical size” and “technical size”. They are not necessarily correlated to each other. We might find an application which is very “large”, functionally speaking, but which is relatively small in terms of technical items by which it is composed of (modules, programs, objects etc.) This can happen, for example, when the software application benefits from a significant level of reuse of technical items.
Even if the terms “logical” and “technical” are commonly perceived as representing different specification types, we should consider that they are not clearly and univocally assignable to software specifications since they depend strictly on the point of view that we choose to describe problems and requirements. If we are the CEO of a company, our problems and goals are relative to the whole organisation and its relations with the environment. A system is considered logically at a very high level. If we are operational personnel, we are interested into smaller problems and detailed features of the systems. Finally, if we are dealing with the problem of managing a TCP/IP network our problems and features are very close to the hardware level. In all of these three cases we may define logical and technical requirements and models. In each of these three contexts, “logical” is the term related to the level of abstraction that a user of a particular piece of software would use in describing his needs and, “technical”, is the term related to any lower level which is instrumental to the so defined user needs. In architectural approaches, this consideration leads to the usage of the “layer” concept, where the needed functionalities of a required system are distributed into homogeneous levels of abstraction (layers) which define what is logical and what is technical for any particular level. It is usually stated that the logical level is linked to the question

"what?" and the technical or design level to the question "how?". In reality, border lines are not so sharp and some "how"s might become "what"s if we only change viewpoint.

When we consider software from the economical point of view - as an asset capable of satisfying buyers' needs - a functional measure seems to be more appropriate to the context of use, since it is linked to a quantification of the different "services" that are supplied by the software application under consideration. Any functional measurement method, indeed, is strongly correlated to the "use value" of the software application itself, so it seems sensible – from a customer viewpoint – to "pay more" for systems that have a higher functional measure.
Focusing contracts on functional measurements, then, has, among the other benefits, a very positive advantage: it motivates the producer to optimise the ratio between functional services (i.e. Function Points) and the technical software items that are needed to implement those functionalities. Since the producer will be paid in proportion to the "services" that the software would release to the users, than he will have an interest in releasing the minimum number of software technical items (i.e. LOC, classes, objects etc.) that will meet the functional requirements needed at the quality level expected. On the contrary, if the contract assigns value directly to the technical software items (i.e. LOC) we should expect – and we often have – an unjustified increase of those measures, the functionalities required being equal. This will generate un-maintainable, incomprehensible, inefficient systems and the paradox that the customer could pay more for suppliers who are less efficient than others (for example, implementing the same functionality using more code).
It is nevertheless true that the effort needed to release a given software product is also dependent on the number and intrinsic nature of the technical items that should be designed, built and tested, as well as on the logical functionalities that those items aim to implement. This is why neither technical measures nor functional measures, alone, are sufficient for a full correlation to effort.
A BSC strategy will probably need "functional" sizing methods (FSM) more than "technical" sizing methods (TSM), whereas project effort estimation in a project management environment will need both of them. There is one case where TSM are equally as important as FSM and should be considered in a BSC approach: a strongly reuse oriented development process where a new software "item" is constructed based on existing and partially adequate (or inadequate) software "items". "Reuse" is a word that could be attributed to two different software viewpoints: the logical organisation and the technical organisation. Both of them are important and describe a different aspect of reuse.

Functional reuse may be defined as the re-use of user recognisable and existing logical data structures and functionalities to build up new logical features. Depending on the particular architectural environment, we might have an extreme situation in which the functional reuse is very high but the technical capability of reusing existing "physical" software items is very low: we must re-build the desired logical item almost from scratch. This is the case, for example, when, in a multi level client-server architecture, we want to deliver a functionality logically similar to an existing one, but in a technical environment completely different from the original one.

Technical reuse may be defined as the re-use of existing physical data structures and software items (modules, objects, programs etc.) in order to build up new technical items to be used in the construction of new logical features. Depending on the particular functional requirements, we might have an extreme situation in which the functional reuse is very low but the technical capability of reusing existing "physical" software items is very high: we can build the desired new logical feature using almost effortlessly some existing technical "items". This is the case, for example, when we want to deliver a set of functionalities to manage (create-update-delete - CRUD) a number of logical files which are similar in structure (i.e. unique id., description, numerical values) but different in contents (i.e. money conversion table, time conversion table, length conversion table etc.).

Functional and Technical reuse may be combined in any possible way. The most significant savings due to a reuse strategy derive from the combination of both reuse types. To improve organisational productivity by the reuse strategy, one should be able to measure the levels of both types of reuse adopted in each project. Since measurement of reuse is not yet a fully mature and standardised we will introduce in this paper only the key concepts of reuse, either Functional Reuse and Technical..

3.3 Multiple uses of FSM in organisations

FSM has been used for a variety of purposes, such as quality assessment, benchmarking, "make vs. buy" decisions, outsourcing contracts [JONE96b] and business process re-engineering analysis [MELI00]. In 1992, IFPUG documented a wider approach to FSM usage, providing a multi-level view on FPA-based measures subdivided into 4 usage areas (Productivity, Quality, Financial, Work Product-Effort) to 5 different audience levels (Corporate, Organization, Responsibility Center, Application, Project) [IFPUG92]. The measures included in the IFPUG document have been listed and sorted by usage type in Appendix A, and by audience level in Appendix B.

Even though the measures identified in the appendix are linked to specific audience levels, there is no clear indication of how they can support corporate strategies, or of how to link them and create the proper "value chain" from the Project level up to the Corporate level (that is, the specific added value of a BSC when compared to a GQM plan [BUGL00].

4. Definition of FSM-based measures for ICT BSC

An analysis of [IFPUG99b] provides an indication that a mapping with a BSC-based measurement framework is possible; it includes in particular:

- Alignment with business objectives (*prioritisation of effort and resources*);
- A large set of measures;
- Integration of measurement into development and support processes (*improved project and process control*).

For the design of a generic ICT BSC with 5 perspectives (the ones cited by both BITS and AIS BSC: Financial, Customer, Process, People, Innovation) a 3-step process was performed to analyse the content of the IFPUG document, map it to the BSC framework and then identify complementary measures to complete the required linkages to the underlying strategic dimension of a BSC:

1. listing of the most relevant FSM-based measures and related audience(s) (as in Appendices A and B);
2. determination, for each measure, of the related candidate ICT BSC perspective (tables below);
3. determination of the ICT BSC most relevant perspectives for the use of a specific FSM-based measure.

In the following tables (Tables 1 to 5), various measures were classified in each of the five ICT BSC perspectives, and, in some instances, related goals/objectives and drivers and questions. In these tables, the IFPUG measures from Appendix A appear in **bold** type**.** The others, appearing in roman type, are additional derived measures usually defined to normalise other measures typically found in business contexts, but quite challenging to operationalise in ICT contexts. For instance, from the Customer perspective, it could be argued that the Market Share of SIO can be increased directly by means of successful development projects, and indirectly by the use of FSM-based measures for effective quantitative project management; in other instances, the ability to implement such measures efficiently can provide strategic advantages (for instance, when they are required by public administrations as a compulsory requisite for submitting a bid).
Some measures can, of course, be listed in more than one perspective. For instance, portfolio size (PS) can be useful both to project managers (Process perspective) and to the Responsibility Center, for the management of multiple application areas from a more business-oriented viewpoint (for instance, to properly distribute the maintenance support workload to several areas in order to achieve the best possible Application Support Rate (ASR).

	GOAL / OBJECTIVE	DRIVER	INDICATOR	COMMENTS / EFFECTS
FINANCIAL (F)				
	Asset Management	Existing asset utilisation	• Total Assets (FSAV) / # employees ($)	
			• **FSAV – FS_{units} Asset Value**	
			• **PS – Portfolio Size**	
	Revenue & Profitability	Revenue	• Revenues / FSAV (%)	

		Growth	• Revenues from new customers / Total Revenues (%)	• *New customers acquired using FSM as a contractual condition for measuring the project – Derived (Improve project governance)*
		Profitability	• Profits / FSAV (%)	
	Financial Management	Organisational Investments	• Investments in IT	
		Project Investments	• **PCFS – Project Cost per FS$_{unit}$**	
			• **ECFS – Enterprise Cost per FS$_{unit}$**	
			• **AMCFS – Application Maintenance Cost per FS$_{unit}$**	

Table 1 – *FSM-based measures: Financial Perspective*

	GOAL / OBJECTIVE	DRIVER	INDICATOR	COMMENTS / EFFECTS
CUSTOMER (C)				
	Customer partnership and involvement	Collaboration	• % projects using integrated teams	
			• **SR – Stability Ratio**	
	Customer satisfaction	SLA	• % SLA met	• *if the agreement uses FSM as a basis for the contract*
	Business Process Support	Innovation usage	• % IT solutions supporting process improvement projects	• *project measurement using FSM*
		Requirements Management	• Requirement Turnover Index [MELI01] RTI = [(Σ_j CRFS$_j$)/ Final FS$_{units}$] * 100 • CRFS = Change Request Function Size units	• *Showing the level of turbulence in requisites during the development phase*
		Problem Management	• **DR – Defect Ratio**	
			• **AR – Application Reliability**	
	Business Growth	Market Share	• % Market share	• *increasing % using FSM as an initial contract condition*

Table 2 – *FSM-based measures : Customer Perspective*

	GOAL / OBJECTIVE	DRIVER	INDICATOR	COMMENTS / EFFECTS
PROCESS (PR)				
	Application Development & Maintenance	Size	• **FS$_{unit}$ – Functional Size unit,**	*According to the FSM method used, it can be expressed for instance by:* • ***FP –** Function Points* • ***C$_{fsu}$** - COSMIC functional size units –*
			• **PS – Portfolio Size**	
		Effort	• **WE – Work Effort**	
		Productivity	• **PDR – Project Delivery Rate**	
			• **EP – Enterprise Productivity**	
		Support	• **ASR – Application Support Rate**	
			• **DDR – Duration Delivery Rate**	
			• **AMPL – Application Maintenance Load per Person**	
		Defectability & Test	• **RCR – Repair Cost Ratio**	
			• **SR – Stability Ratio**	
			• **DR – Defect Ratio**	
			• **TPR – Testing Proficiency Ratio**	
			• **MTTR – Mean Time To Repair ratio**	
			• **AR – Application Reliability**	
			• **DER – Defect Detection Ratio**	
			• # defects / 100 FS$_{unit}$ according to user acceptance	
		Reuse	• **FR – Functional Reuse %**	
			• **TR – Technical Reuse %**	

Table 3 – *FSM-based measures : Process Perspective*

	GOAL / OBJECTIVE	DRIVER	INDICATOR	COMMENTS / EFFECTS
PEOPLE (PE)				
	Core Competencies & Skills	Core Competencies & Skills	• Feedback from FSM-based courses (I&I)	
		Effects of Training	• **DER – Defect Detection Ratio**	

Table 4 – *FSM-based measures : People Perspective*

	GOAL / OBJECTIVE	DRIVER	INDICATOR	COMMENTS / EFFECTS
INNOVATION & INFRASTRUCTURE (I&I)				
	Workforce Improvements	Workforce Competency and development	• IT expended on Training / IT expenses (%)	• *Leverage on the increased forecasting ability of Project Managers (Process perspective) and on their increased satisfaction (People perspective)*
			• % of staff trained in relevant standards or new technologies	• *Training in functional measurement for planning and governance*
			• % employees skilled in advanced application measurement methods	
		Tools & Products	• Investment in new product support and training ($)	• *For FSM-based tools or for courses about FSM-based techniques*
	SPI Improvements	Methodology currency	• % projects measured using recognised methods	
		Support	• **PDR – Project Delivery Rate**	
			• **ASR – Application Support Rate**	
			• **DDR – Duration Delivery Rate**	
			• **AMPL – Application Maintenance Load per Person**	
			• **RCR – Repair Cost Ratio**	

Table 5 – *FSM-based measures: Innovation & Infrastructure Perspective*

Tables 1 to 5 have been drafted for illustrative purposes. Further work is required to verify both their completeness and their usefulness in the context of the BSC. More specifically, further analytical work is required to compare results to date with other public measurement frameworks, such as in PSM and in the framework of the International Software Benchmarking Standards Group – ISBSG [ISBSG00].

Of course, as can be seen from Tables 1 to 5, Process (Pr) is the perspective impacted most in a *direct* way with the use of FSM-based measures, in terms of sizing, estimating and managing the software process. It is still meaningful, however, to stress how the extension of their use to other viewpoints in an ICT BSC will be useful, permitting the creation of indicators with *indirect* reference to them. Furthermore, using FSM-based measures in these other perspectives helps the whole organisation to *indirectly* increase their knowledge – however limited – on the uses of measures of software applications.

5. Concluding remarks

A key issue that needs to be addressed in the design and implementation of a BSC for ICT companies is the identification of measures representative of the software itself, and meaningful for the development of business indicators.

To build a BSC, once the overall strategic direction has been identified, Goals, Drivers and Indicators (GDI elements, e.g. measures) must be chosen for each perspective. Although significant attention has already been paid to the first two elements (Goals and Drivers), probably seen as “structure-related” topics, the third element (Indicators) has been largely neglected. To address this operational measurement issue in the ICT field, we have proposed that FSM be used as the key measurement method for normalising other measurement results across reference values. More specifically, we have discussed how FSM supports the “measure-indicator” issue in an ICT Balanced Scorecard. Through the use of such functional size measures, a significant number of measures can provide a large enough number of both direct and indirect measures for the operationalisation of BSC, and of its business value and relevance, in an ICT company.

Even though the Process (Pr) perspective remains the most impacted of the five, the other four can be improved using FSM-based measures for building the ICT BSC. Of course, the key test of the use of BSC in ICT will be to verify that the "strategy" element still continues to be central to every (business) action.

REFERENCES

[ABRA99] ABRAN A., DESHARNAIS J.M., OLIGNY S., ST-PIERRE D. & SYMONS C., *COSMIC-FFP Measurement Manual, version 2.0*, Ed. S. Oligny, Software Engineering Management Research Lab., Université du Québec à Montréal (Canada), October, 1999.COSMIC, FFP

[ABRA01] ABRAN, A., SYMONS, C., OLIGNY, S., *AN OVERVIEW OF THE COSMIC FIELD TRIALS*, ESCOM 2001, LONDON (UK), APRIL, 2001.

[ALBR79] ALBRECHT, A.J., *Measuring Application Development Productivity,* Proceedings of the Joint SHARE, GUIDE and IBM Application Development Symposium, October 1979, pp. 83-92

[ALBR84] ALBRECHT, A.J., *AD/M Productivity Measurement and Estimate Validation*, IBM Corporate Information Systems and Administration, IBM Corp., Purchase, NY, May 1984

[BRYN93] BRYNJOLFSSON E., *The productivity paradox of information technology*, Communications of the ACM, December 1993, pp. 67-77

[BUGL00] BUGLIONE L. & ABRAN A., *Balanced Scorecards and GQM: what are the differences?,* FESMA-AEMES 2000 Conference Proceedings, Madrid, October 18-20 2000

[EPST97] EPSTEIN M.J. & MANZONI J.F., *The Balanced Scorecard and Tableau de Bord: A Global Perspective on Translating Strategy into Action*, INSEAD Working Paper 97/63/AC/SM, 1997

[FERG99] FERGUSON P., LEMAN G., PERINI P., RENNER S. & SESHAGIRI G., *Software Process Improvement Works!*, SEI Technical Report, CMU/SEI-TR-99-27, November 1999

[GREM97] VAN GREMBERGEN W. & VAN BRUGGEN R., *Measuring and improving corporate information technology through the balanced scorecard*, the Electronic Journal of Information System Evaluation (EJISE), Vol. 1, Issue 1, December 1997, URL: http://is.twi.tudelft.nl/ejise/vol1/issue1/paper3/paper.html

[IBAÑ98] IBÁÑEZ M., *Balanced IT Scorecard Generic Model Version 1.0,* European Software Institute, Technical Report, ESI-1998-TR-009, May 1998

[IFPUG92] IFPUG, *Function Points As Asset. Reporting to Management*, 1992

[IFPUG99a] IFPUG, *Function Point Counting Practices Manual (Release 4.1)*, May 1999

[IFPUG99b] IFPUG, *Guidelines to Software Measurement,* Release 1.1, October 1999

[ISBSG00] ISBSG, *The Benchmark – Release 6*, International Software Benchmarking Standards Group, April 2000, URL: http://www.isbsg.org.au

[ISO00] ISO/IEC/ JTC 1/ SC 7/ WG12 DTR 14143-4, *Software Engineering – Functional Size Measurement - Part 4: Reference Model*, International Organization for Standardization, 20-01-2000

[JONE96a] JONES C., *Applied Software Measurement*, McGraw-Hill, 1996

[JONE96b] JONES C., *Conflict and litigation between software clients and developers*, Software Productivity Research, March 1996

[KAPL92] KAPLAN R.S. & NORTON D.P., *The Balanced Scorecard – Measures that Drive Performance*, Harvard Business Review, Volume 70 No. 1, January-February, 1992, pp. 71-79

[KAPL93] KAPLAN R.S. & NORTON D.P., *Putting the Balanced Scorecard to Work*, Harvard Business Review, Volume 71 No. 5, September-October, 1993, pp. 134-147

[KAPL96] KAPLAN R.S. & NORTON D.P., *The Balanced Scorecard: Translating Strategy into Action*, Harvard Business School Press, 1996

[MELI00] MELI R., *Functional And Technical Software Measurement: Conflict Or Integration ?* FESMA-AEMES 2000 Conference Proceedings, Madrid, October 18-20 2000

[MELI01] MELI R., *Measuring Change Requests to support effective project management practices.* ESCOM 2001 Conference Proceedings, London, April 2-4 2001

[REO99a] REO D., QUINTANO N. & IBÁÑEZ M., *ESI Balanced IT Scorecard Process Perspective V 1.1*, European Software Institute, Technical Report, ESI-1999-TR-016, February 1999

[REO99b] REO D., QUINTANO N. & BUGLIONE L., *ESI Balanced IT Scorecard Infrastructure & Innovation Perspective*, European Software Institute, Technical Report, ESI-1999-TR-043, December 1999

[REO00] REO D., *Applying the Balanced Scorecard for process improvement - A case study by the European software industry,* Presentation, Measuring The Business Value Of IT With The Balanced Scorecard, February 23 & 24, 2000 • Phoenix, AZ

APPENDIX A[5] - *FSM-based measures (sorted by usage area)*

USAGE TYPE	INDEX NAME	PURPOSE	FORMULA	REPORTING FREQ.	AUDIENCE LEVEL
BASE MEASURES					
	FS_{unit} – Functional Size Unit,	**To measure the size of projects and applications It is a primary component of FS_{unit} ratios.** It can be expressed, for instance, by: • FP – Function points • C_{fsu} – COSMIC $_{fsu}$	**[IFPUG99a]** **[ABRA99]**	**According to the counting scope**	• **All audience levels**
	WE – Work Effort	To measure the work-effort expended on application development, major enhancements and software purchase projects, and on maintenance activities	Time in hours	When the work is performed, in terms of time periods (quarterly or annually) and at the end of the project	• All audience levels
PRODUCTIVITY					
	PS – Portfolio Size	To establish and reflect the overall size of the enterprise's portfolio of applications	Sum of **FS_{unit}**	At least annually	• Corporate executive • Organisation • Responsibility center
	PDR – Project Delivery Rate	To measure the delivery rate of new developments and major enhancements or to purchase packaged software projects	**FS_{unit}** /WE	Determined once, at the end of the project	• Responsibility center • Application
	ASR – Application Support Rate	To measure the support rate for existing applications Application support can be divided into categories (repair, enhancement, conversion, user application assistance, preventive maintenance), each of which can be tracked separately	WE/ **FS_{unit}**	Smaller applications annually and larger applications quarterly	• Responsibility center • Application
	DDR – Duration Delivery Rate	To measure how quickly **FS_{unit}** are delivered	**FS_{unit}** / Elapsed Time	At the end of the project and upon request	Calculated by: • Organisation • Responsibility center • Application Reported to: all levels
	EP – Enterprise Productivity	To measure the productivity or rate at which the whole Information System organization provides new or enhanced functions to the end-user.	Tot. **FS_{unit}** / Tot. Information System WE	At least annually In large organisations, quarterly or semi-annual reporting is suggested	• Corporate executive (large companies) • Organisation • Responsibility center

5 Adapted from [IFPUG92]. These measures appear in Tables 1 to 5 in bold characters.

					(smaller companies)
	FR – Functional Reuse %	To measure the level of functional reuse, as specified and discussed in Section 3.2	Every company will define its own standard definition for these indicators. This is a topic with a high level of evolution foreseen for next years.		• Organisation • Project
	TR – Technical Reuse %	To measure the level of technical reuse, as specified and discussed in Section 3.2			• Organisation • Project
	AMLP – Application Maintenance Load per Person	To measure the overall workload of maintenance and support staff	$\mathbf{FS}_{unit}$ / No. of maintenance people	At least annually on all applications and in the month following the end of any major project	• Application • Responsibility center • Organization • Corporate (aggregated values)
QUALIITY					
	RCR – Repair/Cost Ratio	To measure the cost required to research and repair defects (not prevention costs)	[(Tot. Hrs to Repair Defects) * (Hourly Cost)] / $\mathbf{FS}_{unit}$	Every 6 months after project implementation	• Project • Application • Other levels (aggregate)
	SR – Stability Ratio	To measure how well the new application or enhancement met the expectations of the user	(No. of Changes in first 90 days after implementation) / $\mathbf{FS}_{unit}$	90 days after implementation	• Project • Application See also Support Rates
	RTI – Requirements Turnover Index	To measure the impact of Change Requests to the development process. It considers: • Number of CRs • Size of CRs • Amount of wasted results	RTI = [(Σj CRFPj)/ Final FP] * 100 CRFP = Change Request Function Point Defined as in [MELI01]	At each one of the main Phase Review	• All audience levels
	DR – Defect Ratio	To measure the quality of the new development and application enhancements delivered to the user (calculated for all defects or any class of defect)	(No. of defects) / Project $\mathbf{FS}_{unit}$	90 days after project implementation	• Project Lower value for higher levels
	TPR – Testing Proficiency Ratio	To measure the ratio of system defects to the total FP in the project	(No. of defects) / Project $\mathbf{FS}_{unit}$ in application or user test	After each application or user test cycle	• Project Lower value for higher levels
	MTTR – Mean Time To Repair Ratio	To measure the average elapsed time needed to repair defects, showing how quickly defects are repaired.	Elapsed time / (No. of problems)	Monthly	• Application

	AR – Application Reliability	To measure the number of application failures that occurred while in production against the No. of Adjusted **FS**$_{unit}$ in the application	Application Failures / Application **FS**$_{unit}$	Monthly or quarterly	• All audience levels
	DER – Defect Detection Ratio	To record the phase where defects are detected, the phase where the defect is introduced, the severity of defect, the cause of defect	Defects / **FS**$_{unit}$	After each project phase is completed At least annually for determining eventual improvements	• All audience levels Mainly at the Project level
Financial					
	FSAV – FS Asset Value	To place a value on the whole portfolio of applications by calculating the value of a typical application in terms of function size units	(Cost / **FS**$_{unit}$) * Portfolio Size	Annually or semi-annually	• Responsibility center • Organisation • Corporate executive (aggregate)
	PCFS – Project Cost per **FS**$_{unit}$	To measure the relative cost of new development, major enhancement and packaged procurement projects	[(WE in hours * Hourly Cost) + Other Expenses] / **FS**$_{unit}$	Once, at the end of the project, at least annually In large organisations, when needed	• Application • Responsibility center
	ECFP – Enterprise Cost per **FS**$_{unit}$	To measure the total cost to the enterprise, including overhead expenses, to deliver new or enhanced functionality to the end-user	Tot. Information System Costs / Total **FS**$_{unit}$	At least annually In large organisations, when needed	• Organisation • Corporate executive
	AMCFS – Application Maintenance Cost per **FS**$_{unit}$	To measure the overall cost of application maintenance and support (it is a sub-value of PCFS!)	[(WE in hours * Hourly Cost) + Other Expenses] / **FS**$_{unit}$	At least annually In large organisations, when needed	• Application • Responsibility center • Organisation

APPENDIX B[6] - *FSM-based measures (sorted by audience level)*

AUDIENCE LEVEL	RESPONSIBLE FOR…	JOB TITLE(S)	OBJECTIVES (GOALS) & ACTIVITIES	SCENARIO (QUESTIONS)	METRICS
Corporate Executive	• …the Enterprise, including the Board of Directors	• CEO • Chairman of Board • Director	• Provide vision and communicate it effectively to the organisation • Co-ordinate business strategies • Provide the resources needed to support the key business strategies	• Implemented technology must be updated. Q1: What is the current value of company's software? What is the cost to replace it? Q2: How much would be saved by investing in new technology and training?	• APS – Application Portfolio Size • APGT – Application Portfolio Growth Trend • FSAV – Functional Size Asset Value • ECFP – Enterprise Cost per FS_{unit} • RTI – Requirements Turnover Index
Organization	• …managing one or more data processing or IS organisations (indent)Includes all those levels above Responsibility Center and below Corporate Executive. This level may not occur in small organisations	• CFO • CIO • Data processing Manager	• Develop the right projects at the right place, on schedule, with the right operating costs and of the right quality	• He has to face questions like: Q1: Are we being more productive? Are we showing improvement trends? Q2: Are the tools and techniques we use improving our productivity? Q3: What is the technical quality of our applications? Q4: What are our maintenance costs? Are they rising or falling?	• DR – Delivery Rates • CR – Cost Ratios • DT – Defect Trends • ML – Maintenance Loads • DL – Development Loads • RTI – Requirements Turnover Index
Responsibility center	• …managing multiple application areas (indent)This level may not occur in small organisations	• RC Manager • Division Manager • Group Manager	• Maintain the system in operation • Satisfy the request backlog • Improve application quality and performance • Manage portfolio growth • Adapt to new technical environments	• He has to face questions like: Q1: How can we distribute the maintenance support workload to 3 sections to achieve the best possible ASR?	• AS – Application Size • WE – Work Effort • PS – Portfolio Size • ASR – Application Support Rate • RTI – Requirements Turnover Index
Application	• …managing maintenance activities as well as new development or new enhancement projects for one or more applications	• Application Manager • System Manager • Unit Manager	• Maintain the quality and usefulness of the application by adapting it to business and technical changes	• He has to face questions like: Q1: Adding new FS_{unit} to production under your maintenance control, will you need – based on historical data – additional staff to maintain the enhanced application and reduce the backlog? Q2: Are your maintenance personnel sufficiently skilled and trained on more relevant techniques? Do they need to attend courses?	• RCR – Repair Cost Ratio • MTTR – Mean Time To Repair • SR – Stability Ratio • AR – Application Reliability • SR – Support Rate • SAT – Support Activity Trends • AS – Application Size • AMLP – Application Maintenance Load per Person • RTI – Requirements Turnover Index
Project	• …managing individual new developments or major enhancement projects	• Project Leader • Project Manager	• Deliver high quality functionality on schedule and within budget to the customer's complete satisfaction	• He has to face questions like: Q1: Based on current project data, is the project feasible? Can the project be delivered in the forecasted period with the allocated resources?	• WP – Work Project • WE – Work Effort • PDR – Project Delivery Rate • DDR – Duration Delivery Rate • RC – Repair Cost • SR – Stability Ratio

[6] Adapted from [IFPUG92]. Trends measures were not reported (because of their nature) in the list described in Appendix A. These measures appear in Tables 1 to 5 in bold characters.

					• RTI – Requirements Turnover Index • DR – Defect Ratio • TP – Testing Proficiency • DER – Defect Detection Ratio • PCFS – Project Cost per $\mathbf{FS}_{unit}$ • FR – Functional Reuse % • TR – Technical Reuse %