ICEBERG: a different look at Software Project Management

LUIGI BUGLIONE

SchlumbergerSema
Via Riccardo Morandi 32
I-00050 Rome, Italy

Software Engineering Management Research Laboratory
Université du Québec à Montréal
C.P. 8888, Succ. Centre-Ville
Montréal, Québec, Canada

E-mail: lbuglione@rome.sema.slb.com
Tel: (39) 06-8307.4472
Fax: (39) 06-8307.4200

ALAIN ABRAN

École de Technologie Supérieure - ETS
1100 Notre-Dame Ouest,
Montréal, Canada H3C 1K3

E-mail: aabran@ele.etsmtl.ca
Tel: +1 (514) 396-8632
Fax: +1 (514) 396-8684

Abstract

Every project – whatever the application field – should be managed taking into account at least four dimensions: Time, Cost, Quality and Risk. To manage these dimensions, a key tool for a Project Manager is to increase ***project visibility,*** *defined as the amount of information about the project associated with its probability of occurrence. This paper uses the "iceberg" metaphor to introduce the* ***ICEBERG*** *(****I****mprovement after* ***C****ontrol and* ***E****valuation-****B****as****E****d* ***R****ules and* ***G****uidelines) approach that can help Project Managers through the use of standard (de jure and de facto) ICT methods and techniques. This approach focuses not only on the management, and measurement, of resources, process and product, but also of the project and the organization itself. A list of candidate measures related to these 5 entities is suggested for a comprehensive software measurement plan in order to reduce project risk.*

Key words – Software Project Management, Software Measurement, Quality Improvements, ICEBERG, Standards.

1. Introduction

Project management is defined in [PMI96] as "*the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project*". Further: "*Meeting or exceeding stakeholder needs and expectations invariably involves balancing competing demands among the following:*

- *Scope, time, cost and quality*
- *Stakeholders with differing needs and expectations*
- *Identified requirements (needs) and unified requirements (expectations)*

The term ***project management*** *is sometimes used to describe an organizational approach to the management of ongoing operations.*"

This definition includes the dimensions that should be tackled during the management of a project, whatever the application field: Time (T), Cost (C), Quality (Q) and Risk (R)[1] which can be expressed as

$$T = f(C, Q, R)$$

There are several relationships among the three independent variables. For instance, the greater the risk, the greater the possibility of incurring a cost increase as well as longer delivery times and lower product quality delivered. In the software engineering community, researchers have tailored and adapted generic classic risk management techniques to the software world [BOEH91] [VANS92] [CARR93] [ABRA99].

Similarly, significant work has been carried out to tackle the quality issue. This paper presents in section 2 a review of the work carried out to date in the quality domain in section 2, and in section 3, a framework for combining and integrating various contributions from progress achieved to date to ensure greater success in the management of a software project.

---

[1] Risk has been defined in several ways, for instance as:

- *hazard, chance of bad consequences, loss, etc, exposure to mischance* (*The Concise Oxford Dictionary*, 7th ed., 1982)
- *the potential for realization of unwanted negative consequences of an event* (Rowe, W.D. *An Anatomy of Risk*. Malabar, Florida: Robert E. Krieger, 1988., p.24)
- *the measure of the probability and severity of adverse effects* (Lowrance, W.W. *Of Acceptable Risk*. Los Altos, California: William, Kaufmann, 1976, p.94)
- *the possibility of suffering loss, injury, disadvantage, or destruction* (Webster's dictionary, 1981, p.1961)

2. Evolution and contribution of Quality Management approaches

2.1. Evolution of Quality Management frameworks

A Quality Management (QM) system is a risk mitigation strategy. Figure 1 presents the European Software Institute (ESI) view of how quality management approaches have evolved over the past century: quality control, quality assurance and, more recently, total quality including process improvements.

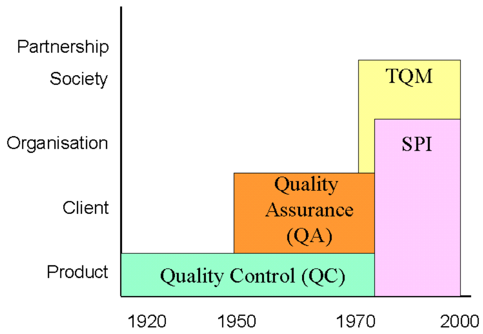

Fig. 1 – *Evolution of Quality usage in the 20th century- ESI*

For instance, Quality Control activities, for example, are typically performed to produce, and document, the quality of the end product (i.e. environmental data). Some classic examples of QC activities include: instrument calibrations, chain-of-custody procedures (they track the samples from collection, through analysis, to disposal) and performance evaluation sampling. Quality Assurance (QA), which came into vogue in the '50s can be seen as a system of management activities involving planning, implementation, assessment and reporting to make sure that the end product (i.e. environmental data) is of the type and quality needed to meet the needs of the user. Some examples of QA processes include: Data Quality Objectives planning system, audits, and Standard Operating Procedures (documents which describe how routine procedures are performed to ensure that they are consistently carried out)[2]. The QA movement remains important in the context of the ISO 9000 certification series.

2.2. Software Process Improvement Models and Quality Models

For a company to be competitive in markets that thrive on innovations, such as in the software field, QA is not sufficient. For example, the control of the conformance of a process adds "repeatability" to the process, but does not address the "*Innovation*" dimension required for continuous improvements of the assets (tangibles and intangibles) owned by the company.

Various software process assessment and improvement (SPAI)[3] models and frameworks have been proposed, such as CMMs, SPICE, Bootstrap. These models are derived from the recognized best practices a company should follow.

There exist, of course, several overlaps, as well as complementarities, between QMs (such as ISO 9001 -- the ISO tailoring for the software organizations) and SPAI models (such as Sw-CMM [PAUL93]). For instance, the CMM tackles some of the areas not addressed in ISO 9001 [ISO94], such as:

- Specific methods (*level 3*)
  - Software architecture analysis
  - Peer reviews
  - Integration testing
  - Software baselines
- Senior management involvement (*level 3-5*)
- Process database (*level 3*)
- Process for development of internal training (*level 3*)
- Standard software process (*level 3*)
- Organization-wide measurement (*level 4*)
- Process improvement (*level 5*)
- Technology change management (*level 5*)

At the same time, there are areas of ISO 9001 that are not addressed in the CMM:

- Records (*clause 4.16*)
- Post-release customer complaints (*clause 4.14*)
- Production (*4.9*), packaging, handling, storage and delivery (*4.15*), servicing (*4.19*)

[2] A list of "popular and powerful methods" for product quality is also cited in [SMIT01].

[3] Those frameworks are commonly referred to as SPI models, but they properly "assess" (even if in an implicit way) and "improve" the organization that adopts them. Thus in our opinion, in order to stress this double role played, it would be more correct to call them SPAI models.

Table 1: *The mapping of SW-CMM v1.1 and ISO 9001 1994 and 2002 versions)*[4]:

TOPIC	SW-CMM v.1.1	ISO 9001:1994	ISO 9001:2000
Corrective actions	L2 KPA SPTO, Goal 2 – Activity 6	4.14.2	8.5.2
Problem prevention	L5 KPA DP	4.14.3	8.5.3
Resources	*Abilities* Common Feature in every KPA	4.1.2.2	6.1 + 6.2.1
Training	*Abilities* Common Feature in every KPA	4.18	6.2.2
Audits	L2 KPA SQA, *Verification* KP in all KPAs	4.17	8.2.2 + 8.2.3
Process and lifecycle definition	L2 KPA SPP, L3 KPA OPD	4.4	7.2 + 7.3.x
		4.9	6.3 + 6.4 + 7.5.1 + 7.5.2
Continuous Improvement	L5	4.14	8.5.2 + 8.5.3
		4.17	8.2.2 + 8.2.3

While, there are no direct connections between QM and SPAI models, there are series of intersections and complementarities – Figure 2. For example, no Key Practices in the CMM are in conflict with the requirements of ISO 9001, and almost all its Key Practices supports some requirements in the ISO standard. Note that ISO 9001 certified companies should have strong foundations at all CMM levels, but without necessarily fulfilling Level 2. Similarly a recognized high maturity level in the CMM model is no guarantee that it will be recognized as compliant with the ISO standard: for instance, it is reported in [PAUL94] that some "mature" organizations have failed to achieve ISO 9001 certification.

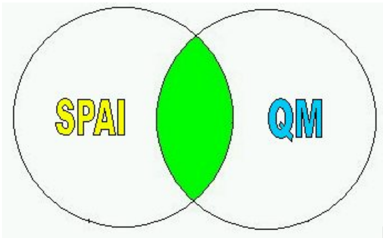

Fig. 2 – *SPAI and QM have both unique characteristics and some overlaps*

It can be also observed that methods and techniques from either model can be grouped according to organizational priorities and strategies for either staged or combined implementation.

2.3. Software Projects Measurement frameworks

Another important set of activities in Project Management involve measurement. This has been recognized in the latest revision of the Sw-CMM model, CMMI (*CMMs Integration*) [SEI02a] [SEI02b], in which a new measurement process area is introduced at Level 2. However, the model does not address the identification and selection of the entities to be measured.

In the software measurement literature, the taxonomy proposed by Fenton & Pfleeger [FENT97] takes into account three types of entities: *resource, process and product.* However, while it is interesting to measure and monitor only the management of a single process or project, not observing and managing the whole picture (the broader context in which a software-related development activity is included) can lead to considerable project risks.

In addition to QM, other approaches, such as the Balanced Scorecard (BSC), have been proposed to tackle the broader organizational perspective by deploying the strategy in a measurable way through the Goal-Driver-Indicator (GDI) elements. A key strength of the BSC is the identification of causal chains (e.g. Porter's Value Chain) which links all the goals across the various perspectives. But BSC and its variants for the ICT field (BITS, AIS BSC, Gartner BSC, etc.) do not have a clear and defined action plan, because of the variability of a strategy which has not been predefined. By contrast, SPAI frameworks are improvement-driven, but miss the strategic element, because there is a predefined and unique strategy foreseen for a generic Software Intensive Organization (SIO[5]). The following table summarizes some of the strengths and weaknesses of the three types of quality approaches (QM, SPAI, BSC):

Table 2: *Quality Models Strengths and Weaknesses*

FAMILY METHOD	STRENGTHS	WEAKNESSES
SPAI	• Path to organizational maturity	• No focus on the business organizational strategy. Pre-defined path from L1 to L5 (staged model)
QM	• Strong focus on Controls and Assurance	• Little attention to improvements (even with the ISO Vision 2000 series)
BSC	• Causal Chain among perspectives	• No clear nor defined action plan after measurement and after drawing the strategy map

[4] For convenience, we refer to the well-known ISO 9001:1994 clauses and then, we present in the following table a cross-reference to the new ISO norm [ISO00], according to the Appendix B.1 mapping table.

[5] SIOs are organizations whose main objective is software development and selling or software, departments of organizations that develop software as integrating part of its final products, or organizations that develop software for internal use to achieve better business results, or whose software department can be qualified as an independent organizational unit (*European Software Institute*, 1997).

3. The ICEBERG approach

To leverage all the approaches surveyed above, we propose to combine and integrate this "path towards quality excellence" under a new acronym: **ICE** (**I**mprovement after **C**ontrol & **E**valuation). Specifically, the "**I**" refers to the Quality Improvement stage in the Quality Evolution schema previously mentioned, the "C" to Quality Control and the "E" to Evaluation in the Quality Assurance context, ICE corresponding to the QC->QA->QI organizational maturity path to Quality.
To manage the risk of a project during its lifecycle, **project visibility**, in particular, must be achieved, monitored and managed. Project visibility has been defined as the amount of available information about the project associated with its probability of occurrence, that it will come true or, using a formula:

$$PV = \sum_{i=1}^{n} (AI * p)_i$$

Where:
PV=Project Visibility
AI=Available Info
P=probability of Occurence of the fact associated to that info
I=current index
N=amount of information that could be managed

This can be illustrated with the **iceberg** analogy, as in Figure 4, where:

- the project is like an *iceberg* – that is, while the project manager knows great deal about his project, a significant portion of it is either not precisely known, or contains large areas of uncertainty and risk, and will evolve over the life of the project;
- the project manager has the captain's role of steering the ship, trying to discover the hidden characteristics of the iceberg (i.e.: submerged dimensions of mass and form, melting situation, velocity, etc.);
- the project visibility level, and a corresponding *waterline below which it cannot be directly observed;*
- the available timeframe from when a problem has been detected until it can be profitably solved, which corresponds to the distance between the ship and the iceberg ("*time for collision*").

In this metaphor, the captain selects navigational course, which would correspond to the organizational strategy for a software project. Like the captain, the project manager is in a situation where a very significant amount of information is not directly available. The question is: how can greater project visibility be derived when only a small portion of it can be directly obtained with a fair degree of certainty? The objective of the proposed ICEBERG approach is to increase the visibility of the project or, using the iceberg metaphor again:

- to determine what is below the "waterline", through specialized studies and the use of the relevant measurement instruments (*more information on the project available through a more indepth measurement activity*);
- to implement mitigation strategies to maintain a sufficient distance between the "ship" and the "iceberg", avoiding crashes (*problems solved on time*);

The project-related view is not enough, however. At least two additional entity types must be taken into account and measured for planning and control – Figure 3:

- the *organization* (since the way it works and is organized significantly impacts *project* performance), and
- the portfolio of *projects* the organization is running and managing; every project can be thought of as a box containing the traditional IPO (or, according to Fenton's labelling, resources-process-product) elements; this is the logic behind the design of the BSC [KAPL92] [KAPL93] [KAPL96], identifying the causal linkages among processes driven by a strategy.

We refer to this set of 5 entity types as **STAR** (**S**oftware entity **TA**xonomy **R**evised) and will use it to monitor an ICT company and its software projects.

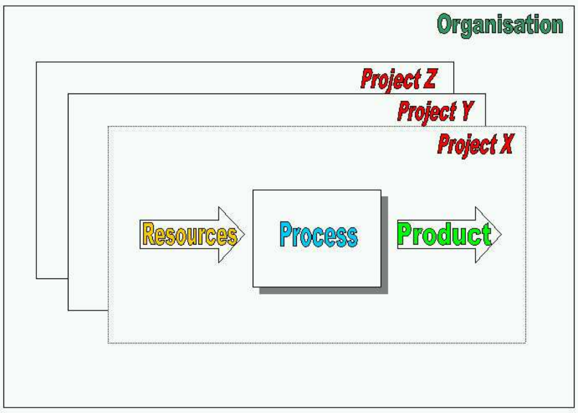

Fig. 3 – *The 5 software entity types measurable in a Software Intensive Organization (SIO) –* ***STAR***

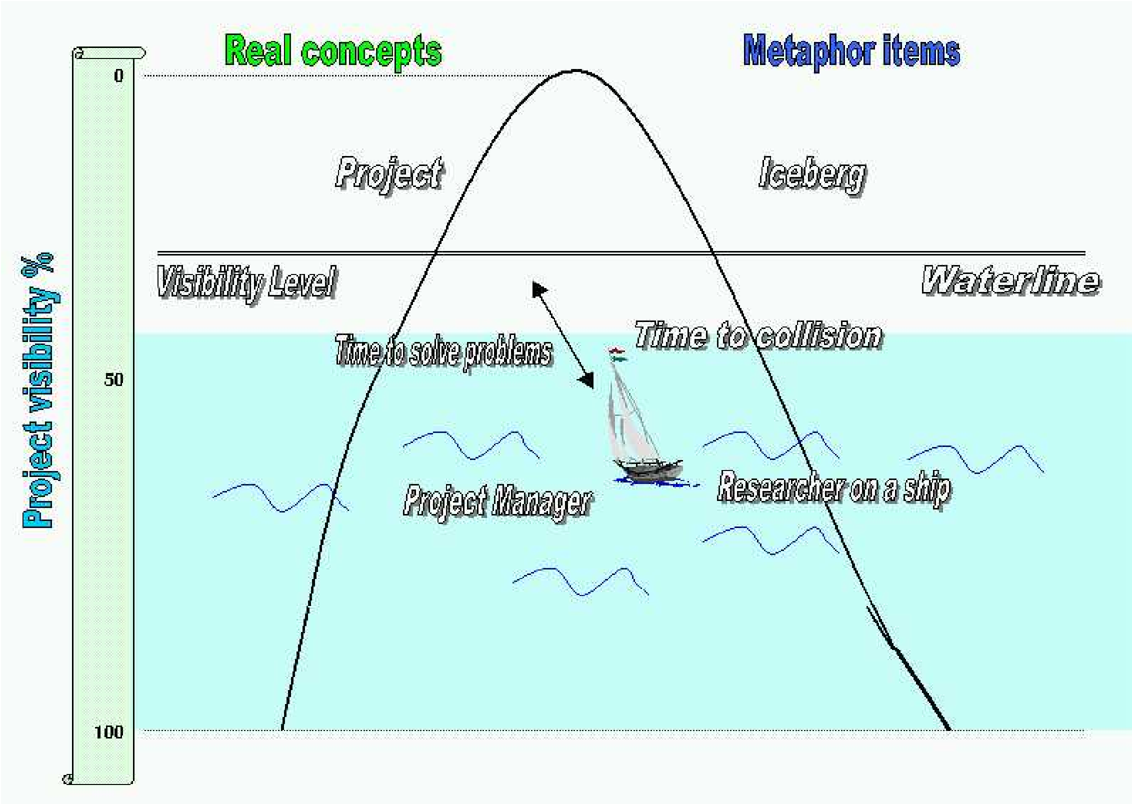

Fig. 4 – *The "Iceberg" metaphor for Project Management*

This combination of SPAI and QM models in Figure 2 can also be combined with the BSC to tackle the strategy viewpoint through the measurement of the five entity types of STAR.. We refer to the combined approach as the ICEBERG (**I**mprovement after **C**ontrol & **E**valuation-Based **R**ules and **G**uidelines) approach.

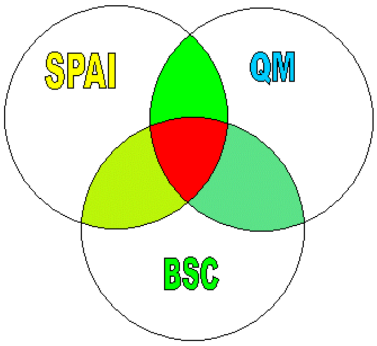

Fig. 5 – *SPAI, QM and BSC models can be combined to improve project visibility with the " **ICEBERG**" approach.*

For a software organization which needs ISO 9000 certification, and also wants to tackle software process improvement (SPI), the set of methods listed in Table 3 would constitute a reasonable strategy.

Table 3: *Families of Models, and respective frameworks available*

FAMILY OF METHODS	FRAMEWORK CHOSEN	WHICH USAGE…
SPAI	• CMMI v.1.1 Continuous Model	• *For the maturity path and the improvement actions (according to BITS)*
QM	• ISO 9001:2000	• *For the Quality Assurance topics*
BSC	• ESI's BITS (5 perspectives) • QEST/LIME [BUGL01] [ABRA02]	• *For the Performance Management issues (general framework)* • *For the measurement of the performances obtained applying BITS*

With the ICEBERG approach, how to address which kind of measures can be used to take into account the five types of entities? While in decentralized environments, managers are left free to choose individual measures (as promoted by the GQM [BASI94] [SOLI99] paradigm), another paradigm involves deploying measures according to a pre-defined

Management framework such as BSC, where the central point is the *strategy* linking all the measurable elements in a coordinated way within an organizational perspective, and from top to bottom[6].

This of course requires implementation of a mapping among the five STAR entities and the BSC approach. Table 4 shows illustrates such a mapping proposal using the 5-perspective classification of the ESI's BITS (Balanced **IT S**corecard) [IBAÑ98] [REO99a] [REO99b] [REO00] for SIOs; this will facilitate the selection and choice of a certain number of measures and indicators.

Table 4 :*Software Entities matching the ICT BSC perspectives*

Software Entity			**Main ICT BSC perspective(s) involved**
1st layer	*2nd layer*	*3rd layer*	
Organization			Financial
	Project		Infrastructure & Innovation
		Resources	People, Customers (requirements), Infrastructure & Innovation
		Process	Process
		Product	Process, Customers (feedback)

In particular:

- *Organizational* entity refers to all the other measurable items required to deal with Infrastructure & Innovation, People, Customers, Financial Perspectives (the other items less than the Process item) in the ICT BSC (i.e. BITS);
- *Project* entity refers to the processes as the logical objects composing a project. Recognized project-related measures are: functional size, Earned Value (EV) and other related time-based measures.

Table 5 lists some possible direct /indirect measures (or indicators) for each entity proposed in the STAR taxonomy.

Table 5 : *STAR entities and related measures/indicators*

Software Entity	**Measures /Indicator**	**Notes**
Organization	- ROI (Return On Investment) - ROS (Return On Sales) - ROCE (Return On Capital Employed) - EVA (Economic Value Added) - Breakeven time - Percent of revenue from products developed in last 4 years - Proposal win % - Cost performance - Net present value of cash outflows for development and commercialization and the inflows from sales - …	*These measures deal with financial issues, as in the Financial Perspective in the ICT BSC*
Project	- Development cycle time trend (normalized to program complexity) - Earned Value (EV) - Schedule performance - Program/project cost performance - Actual staffing (hours or headcount) vs. plan - Personnel turnover rate vs. % of milestone dates met - Schedule performance - Milestone or task completion vs. plan - On-schedule task start rate - Phase cycle time vs. plan - Time-to-market or time-to-volume - …	*These measures are of interest to a Project Manager in deploying his project activity, looking at both Technical and Economical viewpoints.*
Resources	- Percent project personnel receiving team building/team launch training/facilitation - Average training hours per person per year or % of payroll cost for training annually - IPT/PDT turnover rate or average IPT/PDT turnover rate - Percent core team members physically collocated - Staffing ratios (ratio of each discipline's headcount on project to number of design engineers) Personnel ratios - Staffing (hours) vs. plan - Requirements coverage - Technology impact - …	*These measures focus on the management of people, infrastructure and materials, searching for information about the degree of efficiency with which they are managed*
Process	- Product ship date vs. announced ship date or planned ship date - Mean time between failures (MTBF) - Labor hours or labor hours / target labor hours - Mean time to repair (MTTR) - Productivity - Cycle time - Defect containment - Process audit findings - Reference model ratings	*These measures focus on the way a certain process (typical in a given industry) is deployed, in a direct or indirect way*

[6] Some of the key differences between the GQM and the BSC approaches are discussed in [BUGL00].

	- …	
Product	- Product performance or product performance / target product performance or technical performance measures (e.g., power output, mileage, weight, power consumption, mileage, range, payload, sensitivity, noise, CPU frequency, etc.) - Number of parts or number of parts / number of parts for a last-generation product - Defects per million opportunities or per unit - Field failure rates or failure rates per unit of time or hour of operation - Engineering changes after release by time period - Design/build/test iterations - % of requirements analyzed/simulated - …	*These kinds of measures are intended to focus on the final product*

4. Conclusions

Managing a project requires taking into account several project and organizational aspects, in addition to only the technical ones. Many constraints must be considered simultaneously, and one of main challenges is to decrease the probability of project risks.

Several families of methods and techniques have been created and developed somewhat independently over the years; the implementation of one at a time, in any sequence, would not necessarily optimize expected benefits or minimize implementation costs.

In this paper we have proposed a combined implementation by merging Quality Models (such as ISO 9001) with SPAI methods (such as SPICE or CMMI) from an improvement viewpoint and, as also stated in the new ISO 9001:2000 standard. Such a solution is referred to as the **ICE** (Improvement after Control & Evaluation) approach..

Trying to improve processes without a proper cause-effect analysis could however be misguided and represent inefficient use of resources. A more comprehensive framework has been proposed with BSC since ICE alone cannot take into account the organizational strategy. So, the intersection of ICE plus an ICT BSC technique was proposed in a combined approach referred to as **ICEBERG** (Improvement after Control & Evaluation-BasEd Rules and Guidelines).

The application of a BSC framework within ICEBERG means that a greater number of entities will be measured to tackle a larger number of required perspectives. Therefore, the well-known three-entity taxonomy (Resource-Process-Product) was further expanded to handle the two types of entities: *organization* and *project*; this was referred to as the **STAR** (Software entity Taxonomy Revised) taxonomy. A number of candidate measures have been proposed for the five entity types.

Some issues are still open and could be solved through observations in industrial implementations, such as:

- How can ICT BSC be applied in a single technique at the Project level, reporting to the Corporate level). A candidate solution for properly managing the measurement of performance for the different perspectives is illustrated in [ABRA02];
- How to migrate from traditional software project management to an ICT BSC logic using STAR;
- What are the risks inherent in implementing an ICT BSC.

REFERENCES

[ABRA99] ABRAN A., LAFRAMBOISE L. & BOURQUE P., *A Risk Assessment Method and Grid for Software Measurement Programs*, Montréal, Université du Québec à Montréal, Département d'informatique, Technical Report 99-03, 1999

[ABRA02] ABRAN A. & BUGLIONE L., *A Multidimensional Performance Model for Consolidating Balanced Scorecards*, ETS/UQAM, to be published, 2002

[BASI94] BASILI V.B., CALDIERA G. & ROMBACH H.D., *Experience Factory*, Encyclopædia of Software Engineering. Wiley, 1994, pp. 469-476

[BOEH91] BOEHM, BARRY W., *"Software Risk Management: Principles and Practices,"* IEEE Software, January 1991

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

[BUGL01] BUGLIONE L. & ABRAN A., *Multidimensionality in Software Performance Measurement: the QEST/LIME models*, SSGRR 2001 (2nd International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet), L'Aquila, Italy, August 6-10, 2001

[CARR93] CARR M.J., KONDA S.L., MONARCH I., ULRICH F.C. & WALKER C.F., *Taxonomy-Based Risk Identification*, Technical Report, CMU/SEI-93-TR-6, Software Engineering Institute, June 1993

[FENT] FENTON N. & PFLEEGER S.L., *Software Measurement: A Practical Approach*, 2/e, PWS, 1998, ISBN 0534-95429-1

[FRUH00] FRUHAUF K., *A Product is not a Project is not a Process*, Proceedings of the 2nd World Congress on Software Quality (2WCSQ), Yokohama (Japan), September 25-29, 2000

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

[ISO94] ISO, *9001:1994 - Quality systems -- Model for quality assurance in design, development, production, installation and servicing* , International Organization for Standardization, TC176/SC2, 1994

[ISO00] ISO, *9001:2000 - Quality management systems: requirements*, International Organization for Standardization, TC176/SC2, 2000

[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

[PAUL93] PAULK M., WEBER C., GARCIA S., CHRISSIS M.B. & BUSH M., *Key Practices of the Capability Maturity Model Version 1.1,*

Technical Report, CMU/SEI-93-TR-025, February 1993
[PAUL94] PAULK M., *A Comparison of ISO 9001 and the Capability Maturity Model for Software,* Technical Report, CMU/SEI-94-TR-12, July 1994
[PMI96] PROJECT MANAGEMENT INSTITUTE, *A Guide to the Project Management Body of Knowledge (PMBOK)*, PMI Standards Committee, 1996
[POLL] POLLARD D. & BUCKLE R., *Multi_level Balanced Scorecards: Aligning Performance Measurement for Individuals, Team, Business Units and Whole Organizationss*, Ernst & Young Canada, URL: <http://www.pbviews.com/downloads/articles/general002.pdf>
[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
[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
[RISK99] RISKDRIVER, *RISK Management for Projects and Process Improvements*, Eureka EU530 Project, URL: http://www.riskdriver.com
[SEI02a] SOFTWARE ENGINEERING INSTITUTE, CMMI for Systems Engineering / Software Engineering / Integrated Product / Process Development / Supplier Sourcing, Version 1.1, Staged Representation, Technical Report, CMU/SEI-2000-TR-012, March 2002
[SEI02b] SOFTWARE ENGINEERING INSTITUTE, CMMI for Systems Engineering / Software Engineering / Integrated Product / Process Development / Supplier Sourcing, Version 1.1, Continuous Representation, Technical Report, CMU/SEI-2002-TR-011, March 2002
[SEMA02] SOFTWARE ENGINEERING MEASUREMENT AND ANALYSIS TEAM (SEMA), *Process Maturity Profile of the Software Community 2001 Year End Update*, Software Engineering Institute, Software Engineering Institute / Carnegie Mellon University, March 2002
[SMIT01] SMITH L.R., *Six Sigma and the Evolution of Quality in Product Development*, Six Sigma Forum Magazine, Vol. 1 No.1, November 2001, URL: http://www.asq.org/pub/sixsigma/evolution.html
[SOLI99] VAN SOLINGEN R. & BERGHOUT E., *The Goal/Question/Metric Method. A practical guide for Quality Improvement of Software Development*, Mc-Graw Hill, 1999, ISBN 0-07-709553-7
[STSC02] STSC, *CMMI-SE/SW v.1.1 to SW-CMM v1.1 Mapping,* USAF Software Technology Support Center, 2002, URL: http://www.stsc.hill.af.mil/cmmi/
[VANS92] VAN SCOY, R.L., *Software Development Risk: Opportunity, Not Problem*, Software Engineering Institute (CMU/SEI-92-TR-30, ADA 258743), Pittsburgh, Pennsylvania, September 1992.